APPENDIX 1.

TASK DESCRIPTIONS.

NUMBER TASKS.

A1.1.1 N-(1-TO-1).

Purpose: Assess the subject's ability to determine the numerical equivalence relation between three collections each of unspecified cardinal value, using a one-to-one correspondence operation.

Materials: Ten plastic bolts; 10 plastic nuts; 10 plastic washers; cardboard box; 3 opaque plastic jars with screw-on lids.
Two of the bolts were white and were 4 cms in diameter: two were blue and 3.5 cms in diameter; two were green and 3 cms in diameter; two were red and 2.5 cms in diameter; and two were yellow and 2 cms in diameter. The nuts and washers fitted the bolts.

Procedure: The experimenter emptied the bolts, nuts, and washers out of the cardboard box onto the table. He unscrewed the nut and washer off one of the white (largest diameter) bolts, and asked the subject if he knew the names of the three objects. If the subject did not know the names, he was told them. The experimenter then drew the subject's attention to the fact that every bolt also had a nut and a washer, every nut had a bolt and washer, and every washer had a nut and bolt. The experimenter then asked him to unscrew all the nuts; to take all the nuts and all the washers

off the bolts; to place all the nuts in one jar, all the bolts in another jar, and all the washers in a third jar. He was told not to count the nuts, bolts or washers. He was then asked to place the tops on the jars, and to tell the experimenter the names of the parts in each jar. The experimenter then reminded the subject that, for every bolt, there was also a nut, and a washer. The experimenter then went through the following questions:-

(i)"Pick up the jar with the bolts in one hand, and the jar with the nuts in the other. Are there more nuts in there than bolts in there, the same number of nuts there as bolts there, or less nuts than bolts? You can shake the jars, but you can't look inside. How did you work that out? Did you count them?"

(ii) "Pick up the jar with the bolts in one hand and the jar with the washers in the other. Are there the same number of bolts there as washers there, more bolts than washers, or more washers than bolts? You can shake them, but don't look inside. How did you know that? Did you count them."

(iii) "Pick up the jar with the nuts in one hand and the jar with the washers in the other. Are there less nuts than washers, more nuts than washers, or the same number of nuts as washers? You can shake them, but don't look inside. How did you know that?" Did you count them?"

(iv) "O.K., make a guess - how many nuts are there in here? Did you count them?"

If the subject gave correct answers to questions (i), (ii), and (iii), he was told he was was correct and congratulated, provided that he indicated that he hadn't counted. If the subject gave the incorrect answer, or indicated that he counted, the experimenter moved onto the next question, without comment.

Scoring: To pass the subject had to answer questions (i) to (iii) correctly; to give appropriate verbal justifications; and to indicate by answers to question (iv) that he had not counted. Appropriate verbal justifications mentioned some aspect of one-to-one matching, for example: "there was one each when I unscrewed them"; and "every bolt had one."

A1.1.2 N-ORD.

Purpose: Assess the subject's capacity to form an ordered series of collections of objects.

Materials: Four semi-transparent plastic jars, each of which was 10 cms in height and 7 cms in diameter; four lids, one each of blue, red, green, yellow; three blue, five red, seven green and nine yellow plastic blocks, each 2x2x2 cms. The blocks were placed in the jars with the correspondingly coloured lids. The jars were known as the blues, reds, greens and yellows.

Procedure: The experimenter handed the blues, reds and yellows to the subject, and said: "See if you can place these in order. Go from here to here. Put the one with the most here, the one with least here, and the other one in the middle. You cannot count them, but if you pick them up and shake them, you will be able to guess which one has most." (A number of children didn't understand the word "least" but "smallest number" sufficed as a synonym.) When the subject completed that task, he was handed the greens, and asked to find out where they went in the line so that "they all stay(ed) in order." After the subject made the insertion, the

experimenter picked up the greens and moved them to the right of the yellows. He placed his hands on the greens and yellows, and said: "How can you move the blues and reds down this end, so that they will still be in order?" If the subject did not transpose the blues and reds when moving them to the other end of the line, he was asked to think about it again, and was given another opportunity to order the jars. The following illustrates the moves involved:-

(i) Subject sets up series:-

3	5	9
blues	reds	yellows

(ii) Subject inserts greens:-

3	5	7	9
blues	reds	greens	yellows

(iii) Experimenter shifts greens:-

3	5	9	7
blues	reds	yellows	greens

(iv) Subject moves blues and reds:-

9	7	5	3
yellows	greens	reds	blues

Scoring: To pass the task, the subject had to construct the first three term series, insert the greens in the correct place, and invert the series.

A1.1.3 N-CNT.

Purpose: Assess the subject's ability to count an array of n objects and to demonstrate co-ordination of ordinal position and cardinal value.

Materials: Collection of nine coloured plastic building blocks, each measuring 2x2x2 cms: two each of white, red, green, and blue, and one yellow.

Procedure: The collection of nine blocks was handed to the subject, and the subject asked to find out how many blocks there were. After the subject had counted the blocks, he was asked to count them again but, this time, as he counted each block he was asked to push it across the table to the experimenter. As the subject was pushing across the fourth block the experimenter stopped him, and asked if he knew how many the experimenter already had in his pile (three blocks). This was repeated on the eighth block.

Scoring: If a subject gave an incorrect answer he was given another opportunity, but was not told that his answer was wrong. The usual comment was: "let's try that again." A subject had to give correct answers to all questions to pass the task.

A1.1.4 N-TI-EQ.

Purpose: Assess the subject's ability to make a transitive inference of equality with respect to discrete quantity.

Materials: Three semi-transparent plastic jars, each of which was 10 cms in height and 7 cms
in diameter; three lids, one blue, one red, and one green, for the jars; and seven blue blocks, seven red blocks, and seven green blocks, each 2x2x2 cms. The blue blocks were in the jar with the blue lid, the red

blocks in the jar with the red lid, and the green blocks in the jar with the green lid. The jars were known as the "blues," the "reds", and the "greens." It was possible to see the blocks inside the jars, but not sufficiently clearly to enable the subject to count them.

Procedure: The experimenter handed the subject the blues and the reds, and said: "Hold the blues in your left hand and the reds in your right hand. Shake the jars and see whether you can guess whether they each have the same number of blocks, or has one got more than the other. You will not be able to count them, so don't try that." If the subject said "the same," the experimenter agreed, and moved on to the red and green pairing. Otherwise, the experimenter prompted him until he agreed that the jars contained the same number of blocks. If the subject mentioned a specific number of blocks he was told that that was wrong. He was then told not to guess how many were in the jars, just whether the jars had the same number of blocks or not. The same procedure was followed for the red and green jars. After both comparison had been presented, the experimenter placed the reds behind his back, held the blues in his left hand, and the greens in his right hand, so that only the lids were visible to the subject. He then said: "O.K., now you said the blues had the same as the reds, and the reds had the same as the greens. Is that right? Well, what about the blues and the greens, can you work out whether they have the same number of blocks, or has one got more? How did you know?"

Scoring: To pass the task, the subject had to give a correct verbal statement of transitive reasoning.

A1.1.5 N-TI-NE.

Purpose: Assess the subject's ability to make transitive inferences of inequality with respect to discrete quantity.

Materials: Six semi-transparent plastic jars, each of which was 10 cms in height and 7 cms in diameter; six lids, one each of blue, red, green, yellow, white, and black; five blue, seven red, nine green, nine yellow, seven white and five black blocks. The blocks were placed in the jars with the correspondingly coloured lids.

Procedure: As for N-TI-EQ, except that the first presentation tested the relation of less-than, and the second presentation the relation of greater-than. The first presentation was:-

(i) five blues (less than) seven reds, and
(ii) seven reds (less than) nine greens, so
(iii) five blues....? nine greens.

The second presentation was:-

(i) nine yellows (greater than) seven whites, and
(ii) seven whites (greater than) five blacks, so
(iii) nine yellows ....? five blacks.

Scoring: To pass the task, the subject had to give a correct verbal statement of transitive reasoning.

A1.1.6 N-CONS:

Purpose: Assess the subject's ability to conserve number.

Materials: 16 green plastic building blocks, each 2x2x2 cms.

Procedure: The experimenter handed a collection of eight blocks to the subject, and said: " I'm going to put my blocks down in a line across here, so that they are about that far apart (5 cms). I would like you to put yours down in a line opposite mine, so that one of yours is opposite one of mine. Don't count them as you do it, or we will have to start all over again." When the subject and experimenter completed building the lines of blocks, the experimenter said: "We have the same number of blocks in each line, haven't we? Don't count them, just see if you can tell by looking." When the subject confirmed that each line contained the same number of blocks, the experimenter said: "Now I'm going to spread mine out like that, but don't move yours. O.K., now do you think that my line has more blocks than your line, or do you think that they both still have the same number of blocks? Why? How did you figure that out?" The experimenter then reset his line, so that it again stood, block for block, opposite the subject's. He then shortened his line, and asked the subject: "Now, do you still think that my line has the same number of blocks as your line, or do you think your line has more? Why? How did you know that?"

Scoring: The subject was required to pass both versions and to give appropriate verbal justifications, to pass the task. Appropriate verbal justifications were those in the category of either inversion, compensation, or quantitative identity.

A1.1.7 N-ADD-V.

Purpose: Assess the subject's ability to predict the results of addition operations, when the two collections to be added together are visible.

Materials: Twelve red plastic building blocks, each measuring 2x2x2 cms.

Procedure: The experimenter placed the collection of 12 blocks on his side of the table, and told the subject to take two blocks from that collection to his side of the table. The subject was then asked how many blocks he had. When the subject answered "two", he was asked if he knew how many he would have, if he took one more from the experimenter's pile. If the subject attempted to take the block before answering, he was told not to. If he attempted to count using his fingers, or some other object, as a counter, he was also told not to. When the subject answered "three", he was told "that was good, take the block over to your pile." The experimenter then said: "O.K., you've got three blocks now, how many would you have if you took another three blocks from my pile and put them with your's?" This pattern was followed for the following operations:- 2+1; 3+3; 6+2; 8+4.

Scoring: If the subject gave an incorrect answer the experimenter said: "how about doing that again". To pass the task the subject had to give correct answers to all questions.

A1.1.8 N-SUB-V.

Purpose: Assess the subject's ability to predict the outcome of subtraction operations when the two collections are visible.

Materials: Twelve blue plastic building blocks, each measuring 2x2x2 cms.

Procedure: The procedure was the same as for N-ADD-V, except that the subject was asked if he knew how many the experimenter would have left in his pile, if the subject were to take n blocks away. The subtraction operations tested were: 12-3; 9-4; 5-2; 3-1.

Scoring: As for N-ADD-V.

A1.1.9 N-SOL-V.

Purpose: Assess the subject's ability to find the numerical difference between two collections (a and b), and, by addition or subtraction, make the collections equal - this is usually called solving for a difference.

Materials: 16 green plastic building blocks, each measuring 2x2x2 cms.

Procedure: The experimenter handed the collection of 16 blocks to the subject, asked him to give five to the experimenter, four to himself, and to place those left over in the middle of the table. The experimenter then asked who had more blocks. After the subject said that the experimenter did, the experimenter told him that the objective was to make the experimenter's and the subject's collections have the same number of blocks. The subject was told the rules of the game. They were:

(i) Blocks could only be moved from or to the middle, and from or to the experimenter's or subject's piles. They could not be moved from the experimenter's pile to the subject's pile, or vice-versa.
(ii) Only one move was allowed, but, if the subject wanted to move more than one block he had to move them at the same time.
(These rules were designed to force the subject to work out the answer "in his head", and, thereby avoid trial and error solutions). If the subject moved one block from the middle to his pile so that both piles contained five blocks, the experimenter congratulated him and then said: "That's one way of doing it. Let's set it back the way it was, four for you and five for me. Now, can you see another way of doing it?" If the subject took one away from the experimenter's pile so that both piles contained four blocks, he was congratulated. If the subject could not work out the answer, he was prompted by the experimenter saying: "It's all right to take some of mine, or give more to me, if that would help." If he still couldn't solve the problem, the experimenter demonstrated the solution. After this preamble, which was not scored, the test phase was entered. Using the same format, the subject was required to solve the following problems: 7-4=? and 4+?=7; 6-4=? and 4+?=6; and, 6-2=? and 2+?=6.

Scoring: All test questions had to be answered correctly for the subject to pass the task. If the subject made an incorrect move, the experimenter said: "let's put them back and you think about it again." If the subject's second move was also incorrect, he failed that question.

A1.1.10 N-BAL-V.

Purpose: Assess the subject's ability to solve problems of the form: if a > b then [a-(a-b/2)] = [b+(a-b/2)], when all objects are visible (that is, balance the two collections by sharing the difference between them.)

Materials: 16 yellow plastic building blocks, each measuring 2x2x2 cms.

Procedure: The experimenter handed the collection of 16 blocks to the subject, asked him to give six blocks to the experimenter, four to himself, and to put the remainder back in the box and out of sight. The experimenter then asked who had more blocks. After the subject answered correctly, he was told that the objective was to make the piles the same, in that each had the same number of blocks. The subject was told that the rules of the game were:

(i) blocks could only be moved one pile to the other, they could not be moved to one side; and
(ii) only one move was allowed, but as many blocks as the subject wished could be moved at the same time.

(These rules were designed to avoid trial and error solutions.) If the subject answered the practice item correctly the experimenter moved onto the test phase. If not, he was prompted by the experimenter saying: "it's all right to take some of mine across to your side. Do you think that would help?" If he still could not solve the problem, the solution was demonstrated by the experimenter, and the test phase then entered.

The preamble was not scored. In the test phase, he was required to solve the following problems: a=7 and b=3, hence a´=5 and b´=5; a=9 and b=3, hence a´=6 and b´=6. The questioning and prompting format used in the preamble was employed in the test phase.

Scoring: Both test questions had to be answered correctly for the subject to pass the task. If the subject made an incorrect move, the experimenter said: "let´s put them back and you think about it again." If the subject´s second move was also incorrect, he failed that question.

A1.1.11 N-ADD-NV.

Purpose: Assess the subject´s ability to determine the outcome of adding n objects to a collection of similar objects but where that collection of objects is not visible.

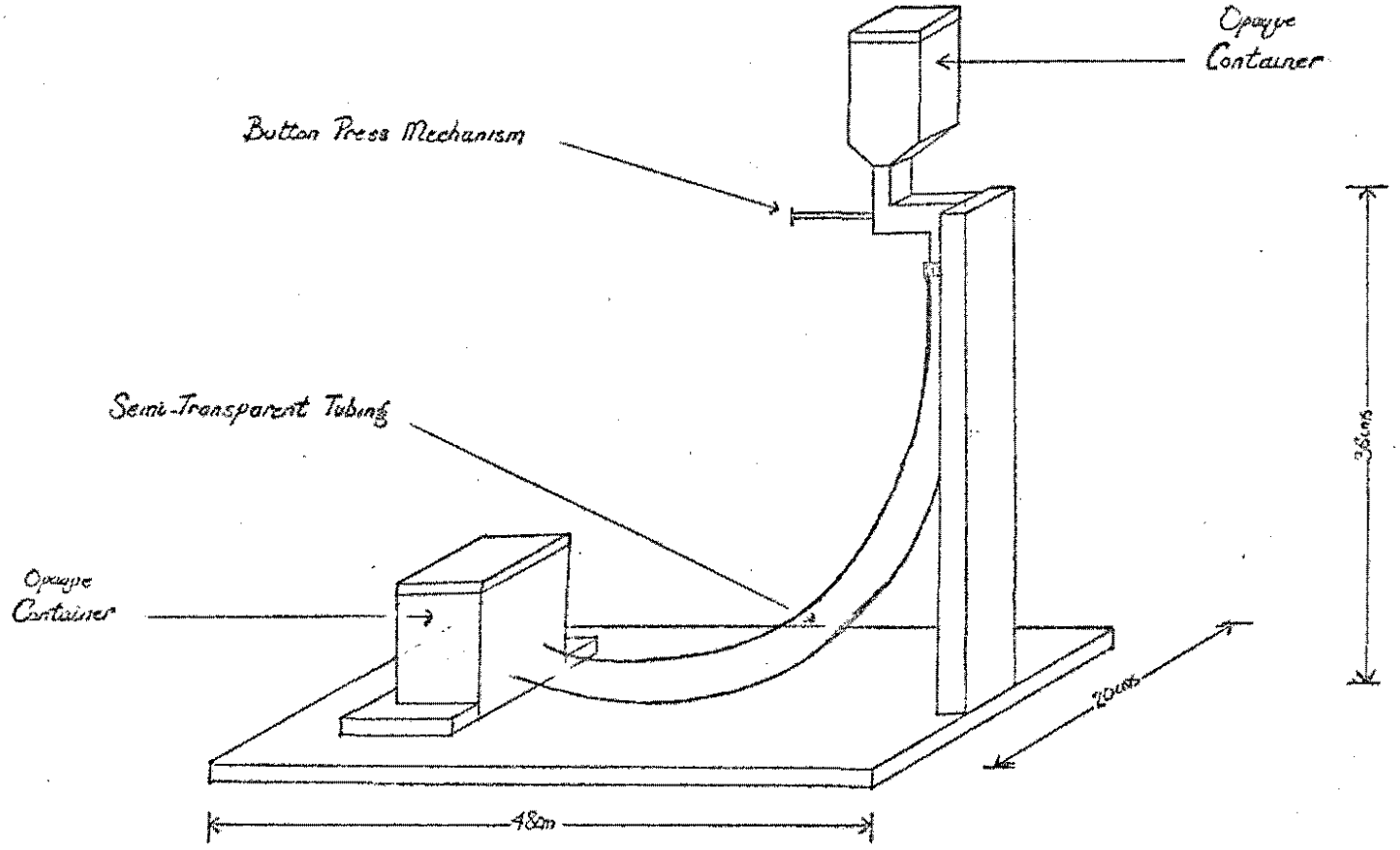
Materials: A counting tube apparatus consisting of one vertical wooden post attached to a rectangular wooden base; an opaque plastic jar attached to the top of that post; another opaque plastic jar attached to the wooden base; a length of transparent plastic tubing connecting the bottom of the top jar with the side of the bottom jar; a machined metal mechanism containing a spring loaded button attached to the bottom of the top jar and the connecting tube; and 12 silver coloured metal ball bearings of 1 cm diameter (see Figure A1.1 for dimensions of apparatus.) The apparatus was designed so that the ball bearings could be placed in the top jar and then sent down the tube to the bottom jar, one by one, by pressing the button. Each button press released one ball bearing. The ball bearings were only visible whilst they were rolling down the tube.

Procedure: The experimenter asked the subject to take the lid off the top jar and check that there were no metal balls inside, and to do the same with the bottom jar. The experimenter then handed the 12 metal balls to the subject, and asked him count them into the top jar. Then, he was asked how many metal balls were in the jar. The subject was then told how the apparatus worked. The experimenter demonstrated it by pressing the button twelve times, and counting the balls as they rolled down the tube into the bottom jar. The subject was then asked how many balls were in the the bottom jar, and how many were left in the top jar. This preamble was repeated, if necessary, until the subject clearly understood what was happening. The subject's responses during the preamble were not scored as part of the task. Following the preamble, the experimenter again asked the subject to verify that there were no balls in either jar or in the tube. The experimenter then handed the 12 balls to the subject, and asked him to count them into the top jar. The subject was then asked to send two balls down to the bottom jar. The experimenter then said: "right, there were none in the bottom jar and you sent two down the tube, so how many balls are there in the bottom jar now?" If the subject gave the wrong answer, the question was repeated. If the answer was again wrong the the experimenter asked the subject to take the top off the bottom jar, look inside and count how many were there. Next, the experimenter said: "O.K., there are two balls in the bottom jar, send down four more." The experimenter then asked the subject how many were in the bottom jar, reminding him that there were two there before, and that he had sent four more down the tube. This format was followed until all of the following addition operations had been performed: 0+2; 2+4; 6+3; 9+1.

Scoring: The subject had to give correct answers to all questions to pass (a correct answer on the second attempt was considered a pass).

N-ADD-NV: SCHEMATIC OF COUNTING TUBE APPARATUS.

FIGURE A1-1



A1.1.12 N-SUB-NV.

Purpose: Assess the subject's ability to determine the outcome of subtracting n objects from a collection of similar objects where that collection but where that collection is not visible to the subject.

Materials: As for N-ADD-NV.

Procedure: The preamble and questioning format used in N-ADD-NV was followed in this task, except that the question related to how many balls were left in the top jar. The subject was reminded of how many were in the top jar before sending the last batch down the tube. The operations tested were: 12-2; 10-4; 6-3; 3-1.

Scoring: As for N-ADD-NV.

A1.1.13 N-CYC-NV.

Purpose: Assess the subject's ability to concurrently work on two collections in a situation where adding to collection (a) meant subtracting from collection (b), and where the objects in those collections were not visible to the subject: that is, the objects cycled from one collection to the other.

Materials: As for N-ADD-NV.

Procedure: The preamble and questioning format used in N-ADD-NV was followed in this task, except that the question related to how many balls were left in the top jar, and how many were in the bottom jar. Also, in posing the question, the subject was reminded that there were 12 balls in total, and that before sending n down there were m left in the top jar. The operations tested were those used in N-ADD-NV and N-SUB-NV.

Scoring: As for N-ADD-NV.

A1.2 LENGTH TASKS.

A1.2.1 LR-B-IN-A.

Purpose: Assess the subject's understanding that if length (a) is greater than length (b), then (a) may be considered as (b) concatenated with some other length (that is, a sense in which (b) is included in (a)).

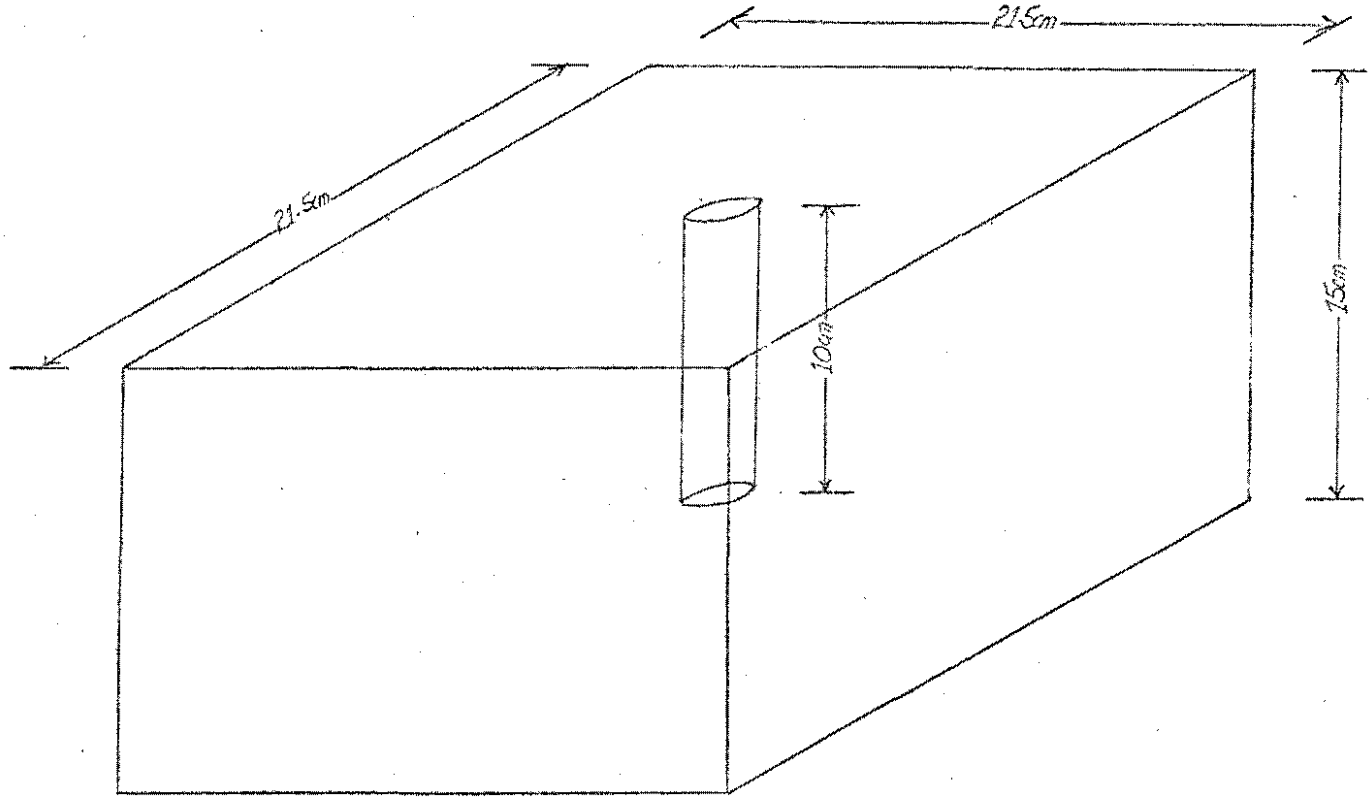
Materials: A perspex box having the following dimensions was used: 21.5 cms in length and width and 15 cms in height, and having walls and a top surface made of perspex 6 cms in thickness. Set vertically into the box was a hollow plastic tube of the following dimensions: 10 cms in length, outer diameter of 1.4 cms, inner diameter of .9 cms. The top of the tube was flush with the top surface of the box. Three lengths of wooden dowel, each of .8 cms in diameter, and of lengths 7 cms, 10 cms, and 13 cms, were also used. See Figure A1.2 for details.

Procedure: The experimenter placed the box on the table, and drew the subject's attention to the hollow tube. The experimenter then placed the three pieces of dowel on the table so that they were in order of length, and drew his attention to that order. The experimenter then asked him to pick up the middle stick, and to push it into the tube so that it went all the way to the bottom. The experimenter then drew his attention to the fact that the top of the stick was flush with the top of the box, and reminded him that that stick was the middle one of the three on the table. The subject was then asked whether he thought that the longer stick of the two left on the table would project above the top surface of the box if it were inserted in the tube, or would it come just to the top, or would it not reach the top. The same question was asked in relation to the shorter stick. The subjects were asked to justify their answers.

LR– BinA.

PERSPEX BOX WITH TUBE.

FIGURE A1.2.



Scoring: Both questions had to be answered correctly, and appropriate verbal justifications given, to pass.

A1.2.2 L-P/W.

Purpose: Assess the subject's knowledge that any object may be considered as a concatenation of arbitrarily selected sub-lengths (that is, an understanding of part/whole relations of length).

Materials: Eight red and eight blue plastic building blocks, each measuring 2x2x2 cms. The red blocks were pushed together to make a red rod, and the blue blocks to make a blue rod.

Procedure: The experimenter placed the red and blue rods side by side, end points aligned, on the table, and sought the subject's agreement that they were the same length. The experimenter then dismantled the red rod, forming a pile of red blocks, and said: "If I put all these together again, would my red rod be as long as that blue one? How do you know?"

Scoring: As well as the correct evaluative response, the subject was required to give an appropriate verbal justification, to pass the task.

A1.2.3 L-INVAR-ADD.

Purpose: Assess the subject's understanding that the length of an object is invariant unless something is added to or subtracted from it (setting aside expansion and contraction processes).

Materials: Ten red plastic building blocks pushed together to make a red rod 20 cms in length; two red blocks; and a perspex box 15 cms high.

Procedure: The experimenter placed the red rod in front of the subject and asked him to indicate with his hands the length of the rod. The experimenter then rotated the rod 90 degrees, and asked the subject whether it was still the same length, and how he knew the answer. Then the experimenter stood the rod on the table and repeated the question. Then the experimenter placed the rod on top of the box and again repeated the question. Then the experimenter placed the two extra red blocks on the table, and asked the subject how he could make the rod shorter; and then, how he could make it longer. In each case, the subject was asked if that was the only way the rod could be made shorter or longer.

Scoring: All questions had to be answered correctly, and the appropriate verbal justifications given. If the subject gave an incorrect answer, he was asked to show the experimenter where the length of the rod started and where it finished, and then to think again about the answer. If the subject again gave the wrong answer, he was asked to explain how he worked it out. Then the experimenter moved onto the next question. If the experimenter gave the correct answer on the second attempt, and an adequate verbal justification, he passed that question.

A1.2.4 LR-INVAR-ADD.

Purpose: Assess the subject's understanding that the length relation between two objects is invariant unless something is added to or subtracted from one of the objects (setting aside expansion and contraction processes).

Materials: Ten blue plastic building blocks pushed together to make one blue rod, and ten green blocks making one green rod; two blue and two green blocks: one perspex box 15 cms in height.

Procedure: The two rods were placed side by side, end points aligned, on the table. The experimenter sought the subject's agreement that they were the same length. The blue rod was then moved through the positions described in the similar one-length task L-INVAR-ADD. The questioning format used in that task was also used in this task, except that, in this case, the questions concerned the length relations between the blue and green rods.

Scoring: As for L-INVAR-ADD.

A1.2.5 LR-INVAR-SP.

Purpose: Assess the subject's understanding that the length relation between two objects is invariant under transformations involving only change of position.

Materials: Two pieces of wooden dowel, each 20 cms in length.

Procedure: The experimenter placed the two pieces of dowel flat on table so that they formed a "V", with the apex pointing toward the subject. The experimenter then asked the subject to find out which stick was the longer, or if they were the same length, telling the subject that he could move the sticks if he needed to. When the subject agreed that they were the same length, the experimenter placed one stick flat on the table with

its long axis parallel to the subject's side of the table. The other stick was placed half way along the length of the first, but vertically on the surface of the table. The experimenter then asked the subject if he still thought that the sticks were the same length, and how he knew that. The positions of the sticks were then reversed, and the questions repeated. The sticks were then placed at opposite ends of the table. The one on the subject's left was parallel with the long side of the table. The one on the subject's right was parallel with the short side of the table. The question was then repeated.

Scoring: All questions had to be answered correctly, and appropriate verbal justifications given, for the subject to pass the task. If an incorrect response was given, the subject was asked to think again. If the subject gave the correct answer on the second attempt, he passed the task.

A1.2.6 LR-ORD.

Purpose: Assess the subject's ability to order objects according to their lengths.

Materials: Five pieces of wooden dowel - one each of the following lengths: 16.5, 17.0, 17.5, 18.0, 18.5 cms.

Procedure: The experimenter handed the three shorter pieces and the longest piece of dowel to the subject, and asked him to arrange them in order of increasing length. After the subject had completed the ordering, the experimenter handed the subject the fifth piece (18 cms) and asked him to insert it in the correct place in the series. If the subject indicated

that he had completed the series, but the sticks were not ordered, the experimenter asked if he was sure that he had finished. If the subject was adamant that he had finished, and that the sticks were in order, the experimenter went through the series asking: "is this longer than than that? No, well maybe you havn't finished yet. I'll pick them up, shuffle them again, and you have another go at it." This was repeated up to three times. If, on the third attempt, the subject still hadn't built a length-ordered series, he failed the task. If he succeeded on or before the third attempt, he passed that phase. Subjects were given only one opportunity on the insertion phase, and were asked to explain why they had placed the stick in the position chosen.

Scoring: To pass, subjects had to construct a length-ordered, four-term series, insert the fifth stick in the correct position, and give an appropriate verbal justification.

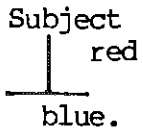
A1.2.7 LR-TI-EQ.

Purpose: Assess the subject's ability to make transitive inferences of equivalence with respect to length.

Materials: Three pieces of dowel, each 25 cms in length, one red, one green, and one blue.

Procedure: The experimenter handed the red and green sticks to the subject, and asked him to find out which was longer, or whether they were the same length. The experimenter then took back the red stick and handed the

subject the blue stick, and repeated the question. The experimenter then took back both sticks, and placed the red and blue sticks in front of the subject in the form of a "T", e.g.



The experimenter then asked: "remember the red was the same as the green, and the green the same as the blue, but what about the red and the blue? Without moving them, can you tell whether they are the same length, or is one longer? How did you know that?"

Scoring: The correct evaluative response and an appropriate verbal justification was required to pass.

A1.2.8 LR-TI-NE.

Purpose: Assess the subject's ability to make transitive inferences of greater than, and less than, with respect to length.

Materials: Six pieces of dowel of the following lengths and colours: red - 15 cms; green - 15.5 cms; blue - 16 cms; yellow - 16.5 cms; white - 17 cms; and grey - 17.5 cms.

Procedure: The procedure used in LR-TI-EQ was also followed in this task. The presentations were:-

Longer Than:

(i) blue. longer. green;

(ii) green. longer. red;

(iii) blue. ? . red, where the blue and red were arranged in the form of a "T", with the shorter one (red) forming the vertical arm pointing toward the subject.

Shorter Than:

(i) yellow. shorter. white;

(ii) white. shorter. grey;

(iii) yellow. ? grey, where the yellow and grey were arranged in the form of a "T", with the shorter one (yellow) forming the vertical arm pointing toward the subject.

Scoring: Correct evaluative responses and appropriate verbal justifications were required on both phases of the task to pass.

A1.2.9 LR-CARD.

Purpose: Assess the subject's understanding that the ordinal length relation between two objects is the same as the cardinal numerical relation between the collections of parts comprising those objects (provided that the lengths of the respective parts are the same).

Materials: Eleven red and 12 green building blocks, each 2x2x2 cms; two semi-transparent plastic jars, 10 cms x 7 cms; two screw-top lids, one red and one green.

Procedure: The experimenter placed the blocks on the table, and asked the subject to find out whether:

(a) all the red blocks were the same size;

(b) all the green blocks were the same size;

(c) the red blocks and the green blocks were the same size.

The experimenter then showed the subject

that the blocks could be pushed together to form rods. The experimenter

then placed seven red blocks in the jar with the red top, and five green blocks in the jar with the green top, without allowing the subject to count how many went in each jar. The experimenter then handed the jars to the subject, and asked him to guess which jar had more blocks, but not to take the tops off and count them. When the subject gave the correct answer, he was reminded that all blocks were the same length, and then asked: "if you took all the red ones out and pushed them together, you would have a red rod. If you did that with the green ones, you would have a green rod. Which rod would be longer, the red one or the green one? How do you know?" This procedure was repeated using eight blocks in each jar, after which all the blocks were placed back on the table. The experimenter then pushed the 12 green blocks across to the subject, and asked him to put them together to make a green rod while he (the experimenter) made a red rod out of the 11 red blocks. When that was done, the experimenter asked the subject to find out which one was longer. After the subject responded, he was asked to disassemble his rod and to place all the green blocks in the jar with the green lid, while the experimenter did the same with the red rod. When that was done, the experimenter then took both jars to his side of the table, covered the jars with his hands, and asked the subject if he knew which jar, red or green, had the greater number of blocks, and how he knew the answer. This procedure was repeated using 10 blocks in each jar. At all times the subject was observed for signs of counting, and was requested not to do so.

Scoring: All four phases had to be answered correctly, with appropriate verbal justifications, to pass.

A1.2.10 LR-TI-CARD.

Purpose: Assess the subject's ability to deduce length relations between objects by applying transitive reasoning to the cardinal number relations between the collections of unit parts.

Materials: Seven red, nine green, and eleven blue plastic building blocks, each 2x2x2 cms; three semi-transparent plastic jars; three screw-top lids, being one each of red, green and blue.

Procedure: The experimenter placed seven red, seven green and seven blue blocks on the table, and asked the subject to check that all blocks were the same length. He then demonstrated that they could be pushed together to make a rod. The experimenter then took back the blocks and placed seven reds in the jar with the red lid, nine greens in the jar with the green lid, and eleven blues in the jar with the blue lid, making sure that the subject could not count the number of blocks in each. The experimenter then handed the blue and green jars to the subject, and said: " hold the blues in your left hand, the greens in your right, shake the jars and see whether you can guess whether they each have the same number of blocks? Or has one got more than the other? You will not be able to count them, so don't try that." After the subject made the correct response, the experimenter took the blue jar back, handed the subject the red jar, and repeated the previous instructions. The experimenter then placed the green jar behind his back, held the red jar in his left hand and the blue jar in his right, so that only the lids were visible to the subject, and said: "O.K., you said that there were more greens than reds, and more blues than greens, what about the reds and the blues? If you put all the reds together you would have a red

rod. If you put all the blues together you would have a blue rod. Which rod would be longer? How did you work it out?"

Scoring: To pass the subject had to give the correct evaluative response, and an appropriate verbal justification.

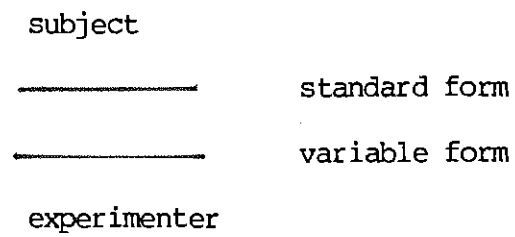
A1.2.11 L-CONS.

Purpose: Assess the subject's capacity to conserve length.

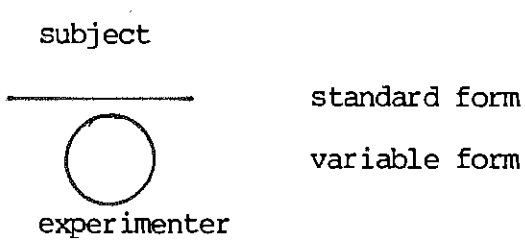
Materials: Two pieces of string, each 20 cms in length.

Procedure: The experimenter showed the subject the two pieces of string. He asked the subject to open one piece out so that it formed a straight line, and to put a finger on each end. The experimenter did the same with the other piece of string, placing it side by side with the subject's, so that the end points were aligned. The experimenter sought the subject's agreement that the two pieces of string were the same length. He then asked the subject to leave his string as it was, while the experimenter bent his into the form of a circle. The experimenter then asked the subject if he thought that the pieces of string were still the same length or not, and to explain his answer. The experimenter then restored his piece of string to its former position side by side with the subject's, and again sought the subject's agreement that they were the same length. He then said that he wanted to form a "T" with the two pieces of string, by leaving his in its present position, and by the subject turning his 90 degrees so that it formed the vertical arm. The same questions were then put to the subject. The two variants of the task are illustrated below:-

(i) (a) first presentation -

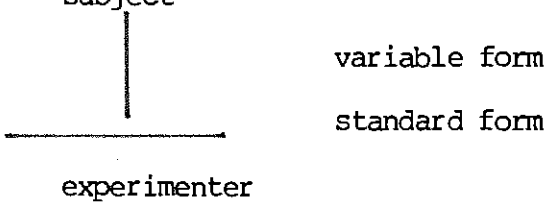


(i) (b) test presentation -



(ii) (a) first presentation - as for (i) (a) above

(ii) (b) test presentation -



Scoring: Both sets of questions had to be answered correctly, and appropriate verbal justifications given, to pass.

A1.2.12 L-UNIT

Purpose: Assess the subject's ability to iterate a unit part along the length of an object.

Materials: Strip of white cardboard 24 by 3 cms, and a second strip 3 by 3 cms; pencil; rubber.

Procedure: The experimenter placed the materials on the table and said: "can you work out how many of these small pieces you would need to make a piece as long as this one? You can move the pieces and use the pencil to put marks on them, if you want to?"

Scoring: Provided the subject attempted to iterate by marking off equal units, and correctly counted them, he passed. (A subject who lacked sufficient motor skill to mark off unit lengths exactly ended up with a portion of the unit measure projecting beyond the end of the strip. Nevertheless, he passed if he attempted to mark off adjacent and adjoining unit lengths.)

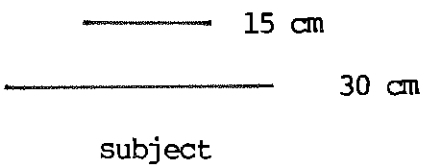
A1.2.13 L-EST.

Purpose: Assess the subject's ability to estimate length in terms of a number of unit lengths.

Materials: Four pieces of dowel, of the following lengths - 1 x 10 cms, 1 x 15 cms, 1 x 22 cms, 1 x 30 cms, and 1 red plastic building block 2x2x2 cms.

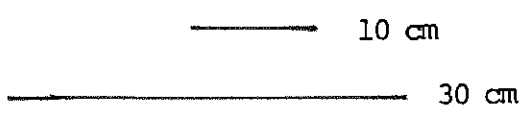
Procedure: The experimenter placed the 15 cm and the 30 cm long stick in front of the subject in the following pattern:-

(i)



He then asked the subject: "how many of these (15cm) would you need to put together to make one as long as that (30cm). How did you work it out without moving them?" The same approach was used for the following two presentations:-

(ii)



subject

(iii)

subject

The subject was not allowed to move the materials, but could point with his finger to the longer object, as he mentally iterated the shorter object.

Scoring: Acceptable answers were:-

(i) two and three;

(ii) two and four;

(iii) eight to fourteen.

Acceptable answers were required to all three questions for the subject to pass. Verbal justifications were not scored.

A1.2.14 L-UNIT-CH.

Purpose: Assess the subject's ability to predict the direction in which the number given by unit iteration would change, if the length of the unit part were to change.

Materials: As for L-UNIT, plus two extra strips of white cardboard, one 4x4 cms, and the other 2x2 cms.

Procedure: This task was presented as an extension of L-UNIT. After the subject had worked out that it took eight of the 3 cm pieces to make one 24 cm strip - or, alternatively, had that demonstrated by the experimenter - the experimenter showed the subject the 4 cm long strip and demonstrated that it was longer than the 3 cm strip. He then asked: "would it take more, less, or the same number of these (4 cm) as these (3 cm) to to make one long one? How did you work it out?" The same procedure was followed with the 2 cm piece.

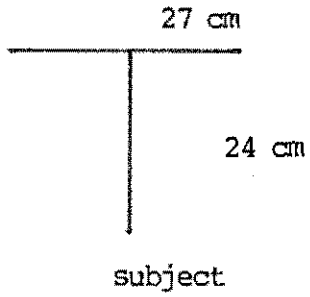
Scoring: Correct evaluative responses to both questions, and appropriate verbal justifications, were required to pass.

A1.2.15 LR-M-CARD.

Purpose: Assess the subject's ability to determine the length relation between two objects on the basis of a measurement operation involving unit iteration and comparison of cardinal numbers.

Materials: Three strips of white cardboard, one 3 x 24 cms, the other 3 x 27 cms, and the third 3 x 3 cms; pencil, rubber.

Procedure: The experimenter placed the two longer strips of cardboard on the table in the form of a "T", with the vertical arm (24 cms) pointing towards the subject, e.g:-



He asked the subject to guess which one was longer, but not to move the strips. The experimenter then handed the 3 cm strip and the pencil to the subject, and said: "O.K., now let's see how good you are at guessing. Without moving these two strips, can you use this little one, and the pencil, to work out whether that one really is longer than this one? You can put marks on them with the pencil if you like."

Scoring: To pass, the subject had to iterate the unit along both longer strips, compare the cardinal numerons, and give the correct answer.

A1.2.16 L-M-ADD.

Purpose: Assess the subject's understanding that numbers representing lengths of objects may be added together, and that the resultant number represents the length of the two objects joined together.

Materials: As for LR-M-CARD.

Procedure: This task was presented as an extension of LR-M-CARD. After the subject had determined that the longer strip required nine units, and the shorter eight units - or, alternatively, had that demonstrated by the experimenter - the experimenter asked: "O.K.,so it takes nine to make that one and eight to make that one. How many would it take to make a strip as long as these two put together? How did you work it out?" When working out the answer the subject was not permitted to reposition the strips, but was allowed to count.

Scoring: The subject had to demonstrate that he knew that the answer was given b adding nine and eight. Subjects demonstrating that knowledge, but having difficulty in carrying out correctly the addition operation, were encouraged to count to get the answer.

A1.2.17 L-ADD.

Purpose: Assess the subject's ability to add lengths in the following (semi-algebraic) fashion: given an ordered series, - (a), (b), (c), (d) - where the increment in length is constant, what is the relation between the combined lengths (a+c) and (b+d)?

Materials: Four pieces of dowel - green, 16.5 cms; blue, 17.0 cms; yellow, 17.5 cms; red, 18.0 cms. Four pieces of dowel of the same colours and lengths and arranged in order of increasing lengths, 2 cms apart, and mounted vertically on a piece of pine-board. One small piece of dowel, .5 cms in length.

Procedure: The experimenter showed the subject the ordered series (the standard), and asked him to construct the same series with the four sticks lying flat on the table. The experimenter called the series a stair-case, and demonstrated with the .5 cm piece that the length of each "step" was the same (i.e. b-a = c-b = d-c = .5 cms.) The experimenter asked the subject to give the first one (16.5 cm.) to the experimenter, the second (17.0 cms) to the subject, the third (17.5 cms) to the experimenter, and the fourth (18.0 cms) to the subject. The experimenter then said: "if I put my two together like this, and you put your two together, who would have the longer stick? How did you work it out?" Notice that the standard was present throughout the presentation, and that the experimenter held his two sticks in such a manner as to prevent the subject from making a visual comparison.

Scoring: The subject passed the task if he gave the correct evaluative response, and appropriate verbal justifications.

A1.3 DISTANCE TASKS.

A1.3.1 D-CONS.

Purpose: Assess the subject's ability to conserve distance.

Materials: One red and one blue plastic building block, each measuring 2x2x2 cms; a path pattern drawn on cardboard (see Figure A1.5); a strip of white cardboard 8 x 13 cms to use as a wall, with a 'door' drawn on both sides.

FIGURE A1.5.

D--COMS.
PATH
PATTERNS.

Procedure: There were three parts to the task.

(i) The experimenter placed the two blocks on the table about 50 cms. apart, and said: "I want you to pretend that that's a red house, and that's a blue house, and that there is a path between the two houses. I want you to pretend that on Monday you walk from the red house to the blue house along the path (experimenter demonstrated with his fingers). On Tuesday you walk back from the blue house to the red house along the path, but when you get to here (20 cms from the red house), you find that somebody has put a wall across the path. But the wall has got a door in it, so you open the door, go through, and walk onto the red house. O.K., now let's see if you remember that. You show me where you walk on Monday. Good. Now, where do you walk on Tuesday? When is the wall on the path? Tuesday. Right. Would you walk further on Monday, or on Tuesday, or the same both days? Why?"

(ii) The experimenter placed the path pattern on the table and asked the subject if it would be further to walk from A to B, or further to walk from A' to B', or would it be the same distance along the respective paths. The experimenter then asked the same question but in connection with the reverse direction - ie. comparing (B -> A) with (B' -> A').

(iii) The experimenter then placed the red block at A, and asked the subject to place the blue block at A'. The experimenter then asked the subject to pretend that the blocks were cars, and asked the subject to move his car along his path the same distance as the experimenter moved his car along his path. The experimenter then moved the red block along the path in increments of about 4 - 5 cms, and noted the subject's movement of his block.

Scoring: All three questions had to be answered correctly and appropriate verbal justifications given to pass.

A1.3.2 D-EST.

Purpose: Assess the subject's ability to estimate distance between two points in terms of a number of unit distances.

Materials: Three sheets of white cardboard on two of which were drawn four squares and on one of which were drawn three squares, 2x2 cms in size, as shown on Figure A1.6; four plastic building blocks, red, blue, green and yellow, each 2x2x2 cms.

Procedure: There were three questions:

(i) The first pattern was placed on the table, and blocks placed in the squares. The subject was then asked to pretend that the blocks were houses, and that there were paths between the houses. The experimenter then asked the subject to guess how many times the shorter path would go into the longer path. If 'times' was not understood, he was asked how many of the shorter paths would he need to put together to make one longer path.

(ii) The second pattern was placed on the table and the same questions asked.

(iii) The third pattern was placed on the table, a red and blue block placed over the left and right squares, and a green block placed on the middle square. The subject was asked to pretend that the red and blue blocks were houses, that there was a path between the houses, and that the path was made of green blocks. The subject's task was to estimate the number of blocks needed to build the path, without moving the blocks.

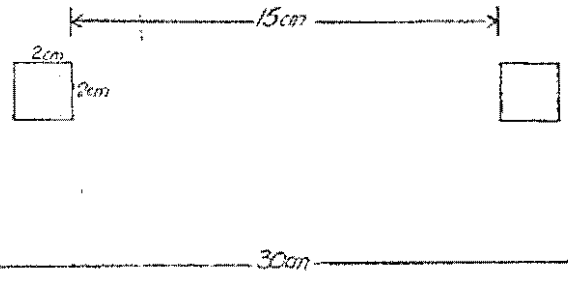
Scoring: As for L-EST.

D_EST.

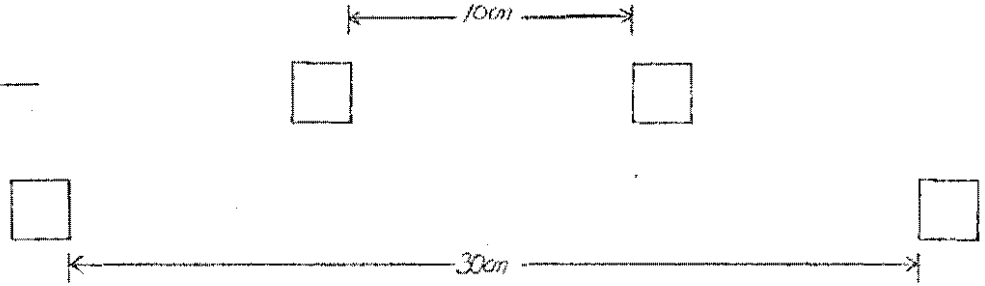
PATH PATTERNS.

FIGURE A1.6.

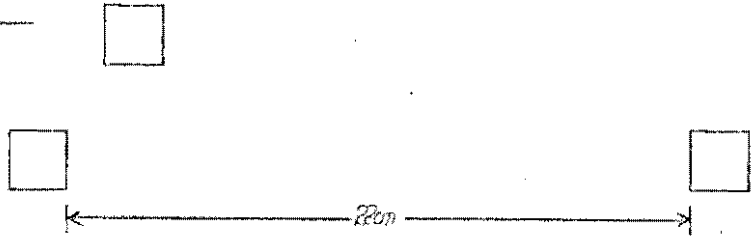
PATTERN 1.



PATTERN 2.



PATTERN 3.



A1.3.3 DR-M.

Purpose: Assess the subject's ability to compare two distances indirectly on the basis of measurement operations, but not necessarily using unit iteration.

Materials: The horizontal distance plate described in D-M above, together with a vertical distance plate of the same construction, except that the pieces of dowel were separated vertically, one being directly above the other; one strip of white cardboard 3 x 35 cms; pencil; rubber. Figure A1.4 illustrates the vertical distance plate.

Procedure: The experimenter placed the two plates on the table, and said: "I want you to look at the distance between here and here (24 cms between arms on the horizontal plate) and the distance between here and here (27 cms between arms on the vertical plate). Without moving them, can you work out whether it is bigger from here to here, than from here to here? You can use this long strip if you want to. You can write on it with this pencil, if you need to."

Scoring: The subject had to use the long cardboard strip in an appropriate manner, and give the correct answer, to pass.

A1.3.4 D-M

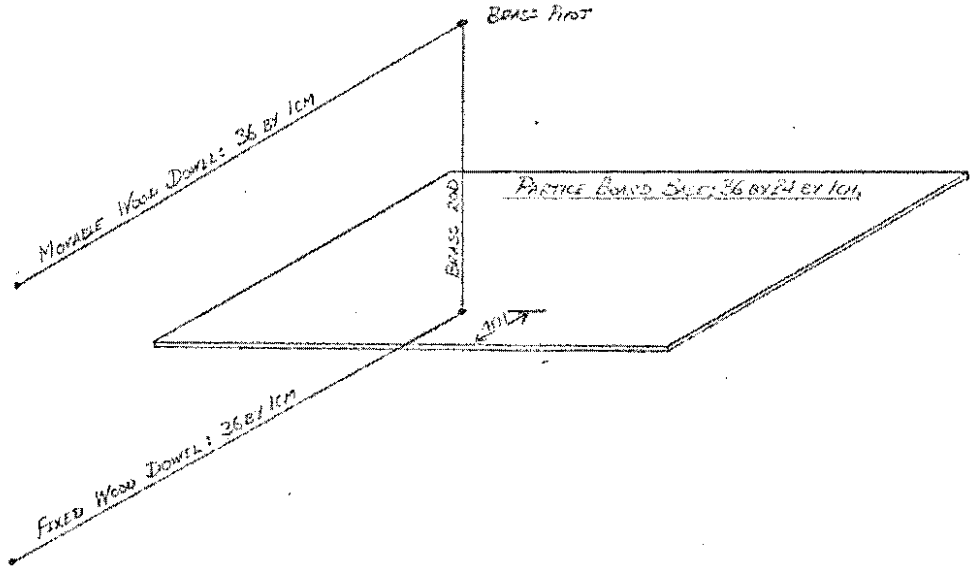
Purpose: Assess the subject's ability to measure the distance between two objects using unit iteration.

Materials: A horizontal distance plate consisting of a wooden base to which were attached two pieces of dowel, each cm in length, projecting

DR_M.

VERTICAL MEASURING PLATE.

FIGURE A1.4.



out from the edge of the base cm; two strips of white cardboard, one 3 x 35 cms, and the other 3 x 3 cms; pencil; rubber. The pieces of dowel were cms apart and could be moved in the horizontal plane so as to adjust the distance between their end points. Figure A1.3 illustrates the apparatus.

Procedure: The experimenter set the arms of the horizontal plate so that the separation between their end points was 24 cms. He placed it on the subject's side of the table, so that the edge of the plate was co-incident with the edge of the table. The experimenter then said: "I want you to pretend that you are going to build a bridge from here (a) to here (b). You want it to go exactly from here to here, so that it doesn't stick out at all. Pretend that you are going to build the bridge out of these small strips of cardboard (3cms). Your job is to find out how many of these small strips you'll need to build the bridge. If you want to, you can use this long piece (35 cms) to help you work it out. You can use this pencil to put marks on them, if you need to."

Scoring: The subject had to use the longer strip and the shorter strip to measure the distance between the arms. The quality of the subject's attempt to iterate the unit was taken into account. Hence, a subject didn't have to be exact in marking off units to pass the task. Provided that the subject attempted to be accurate and precise in marking off units, and counting units, he passed.

FIGURE A1.3

D.M.

HORIZONTAL MEASURING PLATE.

APPENDIX 2.

RAW DATA SCORES.

APPENDIX 2: TABLE A2.1: RAW DATA MATRIX — SUBJECTS BY TASKS — 1 INDICATES PASS AND 0 INDICATES FAIL

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	SUBJECT NUMBER
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	N-ORT
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	N-TI-EQ
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	LR-TI-ER
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	LR-CARD
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	LR-BUA
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	L-B/W
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	N-1784
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	LR-INVAR-ADD
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	N-ADD-V
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	LR-ORD
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	M-SUB-V
1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	N-ORD
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	L-INVAR-ADD
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	N-CONS
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	LR-INVAR-SP
1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	L-CONS
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	N-SOL-V
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	L-BAL-V
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	N-ADD-N
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	L-EST
1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	L-UNIT
1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NR-M
1	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	L-UNIT-CH
1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	LR-TI-CARD
0	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	D-CONS
1	0	1	0	0	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	N-TI-NV
1	1	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	D-EST
0	0	1	0	1	0	0	0	1	1	1	0	1	0	0	1	1	1	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	LR-TI-NV
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	D-M
0	0	1	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	N-SUB-NV
0	0	1	0	1	0	1	0	0	1	1	0	1	0	0	1	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	N-CAL-NV
1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	LR-M-CARD
1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	L-M-ADD
1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	1	0	0	1	1	1	1	1	1	1	1	L-ADD
26	22	27	27	27	27	27	27	25	23	24	26	30	30	30	30	30	30	31	31	31	31	32	33	33	34									TOTALS

TABLE A2.1 CONTD.

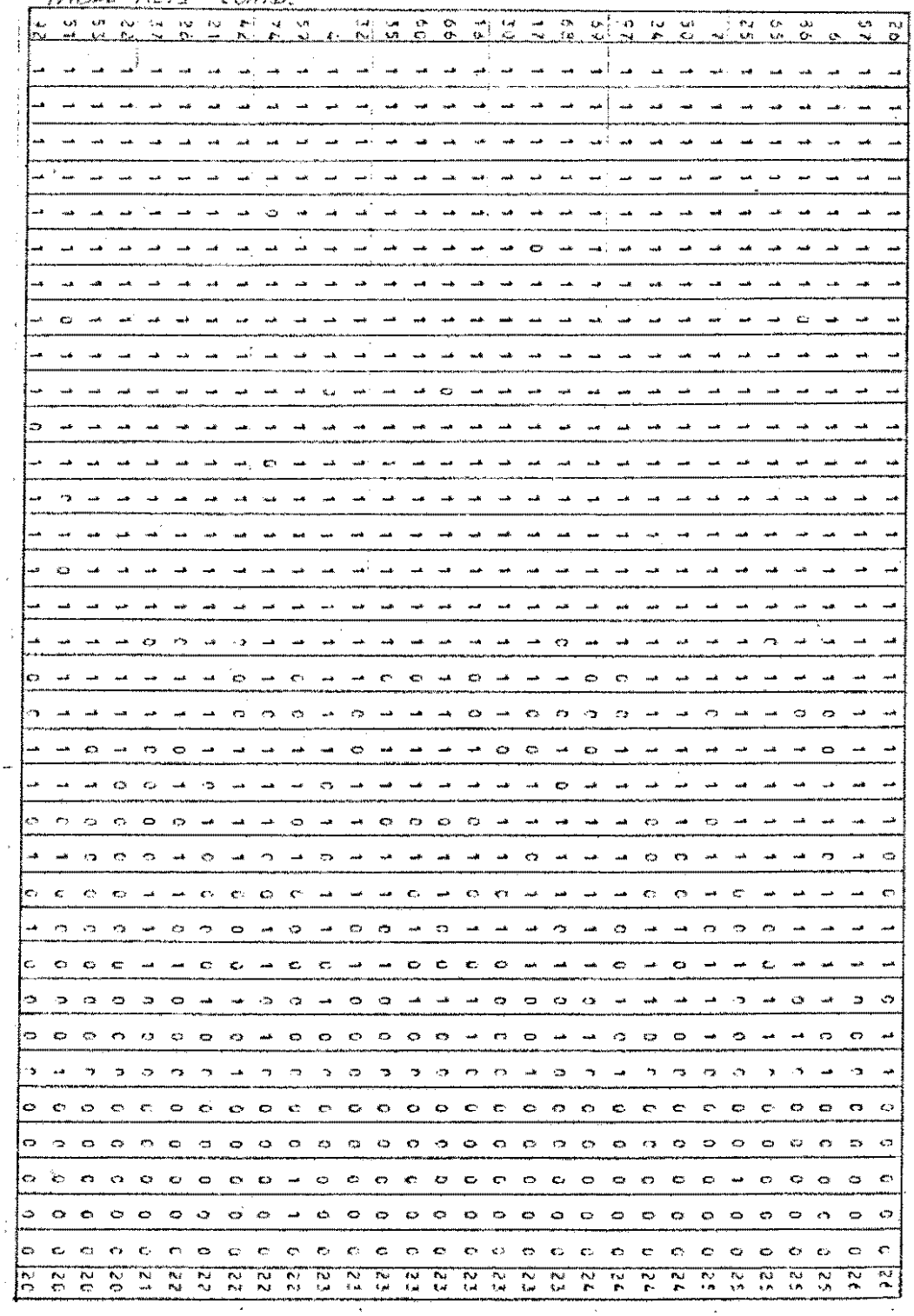


TABLE A2.1 CONTD.

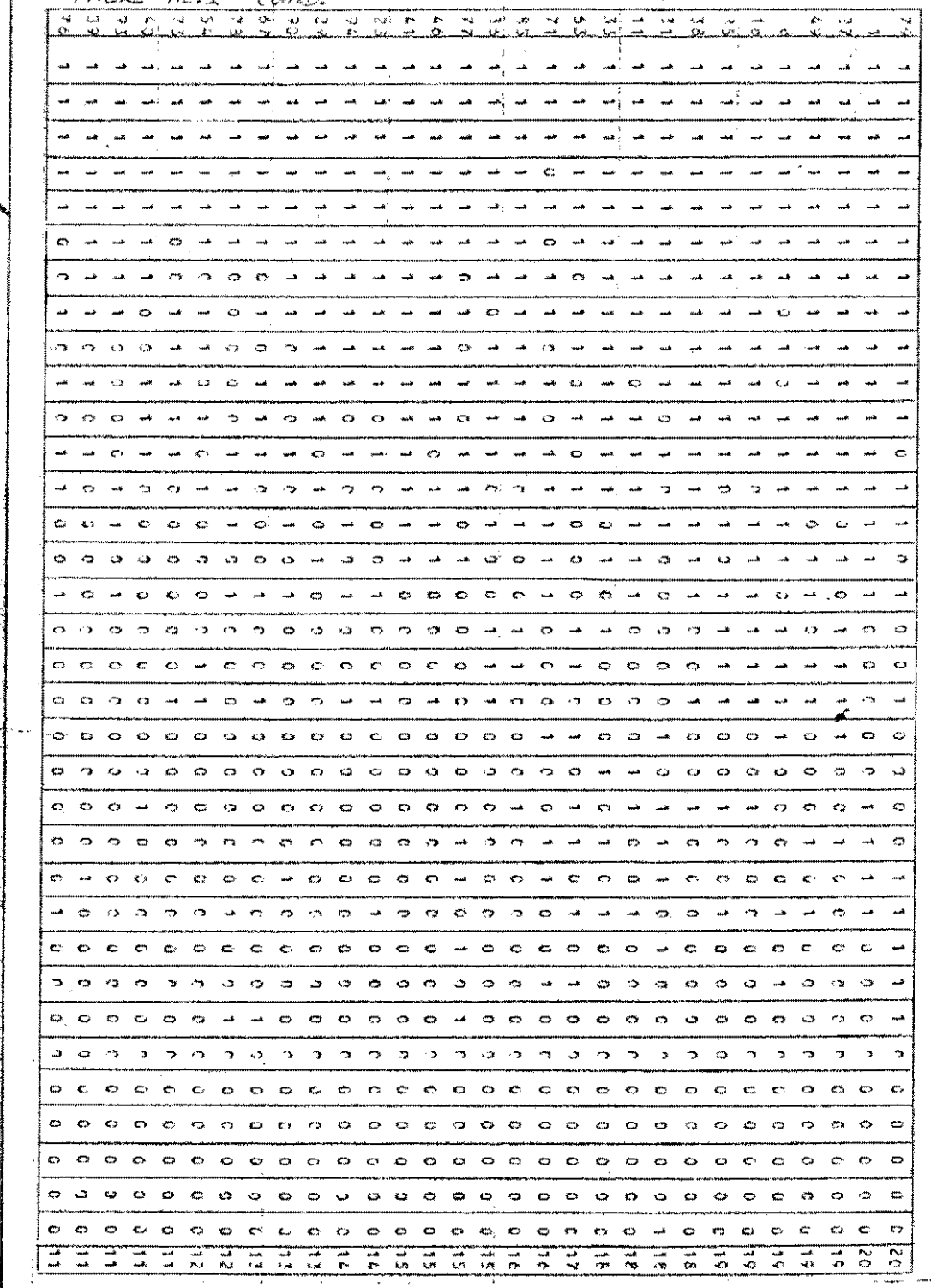


TABLE A2.1 CONTD.

SUBJECT NUMBER	N-CNT	N-TI-EQ	LR-TI-EQ	LR-CARD	LR-BinA	L-P/W	N-1TS1	LR-INVAR-ADD	N-ADD-V	LR-ORD	N-SUB-V	N-ORD	L-INVAR-ADD	N-CONS	LR-INVAR-SP	L-CONS	N-SEL-V	N-BAL-V	N-ADD-NV	L-EST	L-UNIT	DR-M	L-UNIT-CH	LR-TI-CARD	D-CONS	N-TI-NE	D-EST	LR-TI-NE	D-M	N-SUB-NV	N-CYC-NV	LR-M-CARD	L-M-ADD	L-ADD	TOTALS
3	1	1	1	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
32	1	1	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
43	1	1	1	1	1	1	0	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
45	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
43	1	1	1	1	1	1	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
39	1	1	1	1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	10
44	1	1	1	1	1	1	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9
37	1	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
72	1	1	1	1	0	0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
47	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
5	1	1	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	7
70	1	1	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
Σ	100	100	100	9[illegible]	95	94	8[illegible]	85	84	82	81	80	80	73	74	74	61	58	58	56	53	53	49	46	48	41	34	29	24	21	16	14	13	102	079

TASKS ARE LISTED LEFT TO RIGHT — IN ORDER OF INCREASING DIFFICULTY

SUBJECTS ARE LISTED TOP TO BOTTOM IN ORDER OF DECREASING TOTAL SCORE

APPENDIX 3.

CHI-SQUARED MATRIX - ALL TASKS.

APPENDIX 3. ALL TASKS.

(The names of the tasks identified by number in the matrix are given below.)

CHI SQUARED MATRIX

TASK	TASK 1	10	36	21	16	20	9	17	2	18	3	12	15	13	16	24	4
1	.00**********		.50	3.20	4.17	12.07	13.07	14.06	16.06	17.05	18.05	18.05	20.05	24.04	24.04	37.03	
10	.00	.00********	.50	3.20	4.17	12.07	13.07	14.06	16.06	17.05	18.05	18.05	20.05	24.04	24.04	37.03	
36	.00	.00	.00	.50	3.20	4.17	12.07	13.07	14.06	16.06	17.05	18.05	18.05	20.05	24.04	24.04	37.03
21	.00	.00	.00	.00	.57	1.50	7.56	8.47	12.07	12.50	15.06	14.45	13.14	15.04	18.89	20.35	35.03
14	.00	.00	.00	.00	.00	.00	4.27	6.75	5.88	8.47	8.45	13.07	9.33	12.19	16.00	17.39	28.66
20	.00	.00	.00	.00	.00	.00	3.06	3.76	5.06	6.05	7.58	7.68	7.68	10.23	15.04	13.88	29.26
9	.00	.00	.00	.00	.00	.00	.00	.00	.06	.64	.84	1.56	1.14	4.90	6.05	5.04	19.86
17	.00	.00	.00	.00	.00	.00	.00	.00	.00	.27	.45	.96	.84	1.71	4.76	4.76	15.56
2	.00	.00	.00	.00	.00	.00	.00	.00	.00	.06	.44	.45	.41	1.39	4.05	3.12	21.04
18	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.08	.03	.41	2.04	2.04	13.79
3	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.17	1.71	1.33	16.41
12	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.04	.06	1.04	1.25	12.00
15	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.05	2.50	.96	9.82
13	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.45	.50	10.24
16	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.04	5.76
24	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	4.65
4	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
5	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
6	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
26	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
28	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
33	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
29	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
23	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
25	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

MATRIX CONTINUED

TASK	5	6	26	28	33	29	23	25	11	27	22	32	7	8	30	31	19
1	40.02	40.02	42.02	45.02	45.02	49.02	50.02	50.02	57.02	64.02	69.01	72.01	77.01	82.01	84.01	85.01	88.01
10	40.02	40.02	42.02	45.02	45.02	49.02	50.02	50.02	57.02	64.02	69.01	72.01	77.01	82.01	84.01	85.01	88.01
34	40.02	40.02	42.02	45.02	45.02	49.02	50.02	50.02	57.02	64.02	69.01	72.01	77.01	82.01	84.01	85.01	88.01
21	38.02	38.02	38.20	43.02	43.02	45.18	46.17	48.02	55.02	60.14	67.01	70.01	75.01	80.01	82.01	83.01	86.01
14	33.23	35.03	35.22	38.20	38.20	44.02	45.02	41.49	50.16	59.02	62.13	67.01	72.01	77.01	79.01	80.01	83.01
20	32.24	32.24	34.22	37.21	37.21	41.19	40.50	40.50	49.16	56.15	63.02	64.13	71.01	76.01	78.01	79.01	82.01
9	22.78	21.44	24.74	31.03	29.26	31.61	34.22	28.52	41.19	48.17	49.78	58.02	63.02	68.01	70.01	71.01	74.01
17	19.31	20.48	21.19	26.69	26.69	30.62	33.23	27.57	40.20	47.17	50.42	55.15	62.02	67.01	69.01	70.01	73.01
2	24.04	22.32	24.30	29.03	27.27	29.64	27.84	29.17	39.20	46.17	47.80	56.02	61.02	66.01	68.01	69.01	72.01
18	15.56	16.53	17.36	23.76	22.40	27.68	28.66	22.69	39.02	39.45	49.16	54.02	59.02	64.02	66.01	67.01	70.01
3	21.04	17.93	18.58	24.30	24.30	24.02	23.81	23.81	34.57	43.18	46.45	53.02	58.02	63.02	65.01	66.01	67.12
12	14.70	14.70	16.53	21.81	20.48	25.71	26.69	22.88	33.58	38.94	45.45	50.16	55.15	60.14	64.02	65.01	68.01
15	13.78	11.02	16.53	21.81	17.33	25.71	24.02	25.29	33.58	42.19	47.17	48.43	57.02	62.02	62.13	63.13	64.34
13	12.89	11.28	15.75	21.33	19.86	20.10	24.74	19.11	33.23	40.20	43.47	50.02	55.02	60.02	62.02	63.02	66.01
16	8.04	6.25	10.32	16.00	12.12	18.58	17.36	14.88	26.26	33.07	37.96	42.48	51.02	56.02	56.15	57.14	58.37
24	7.03	7.03	9.03	14.81	12.12	15.57	17.36	15.63	24.98	31.69	37.96	44.18	49.16	54.15	56.15	57.14	58.37
4	.24	.14	.84	2.72	1.88	3.78	4.11	4.36	11.28	18.27	21.84	29.64	38.02	43.02	43.18	44.18	47.17
5	.00	.05	.04	.59	.70	1.83	2.53	2.25	8.83	13.92	19.12	25.29	35.03	40.02	38.52	39.51	42.48
6	.00	.00	.03	.46	.43	1.49	2.02	1.93	6.56	13.22	16.68	22.88	35.03	40.02	38.52	39.51	42.48
26	.00	.00	.00	.19	.15	1.57	1.63	1.29	6.32	16.96	18.27	24.74	33.03	38.02	40.02	41.02	44.02
28	.00	.00	.00	.00	.04	.41	.55	.46	5.04	9.82	13.92	25.04	26.69	33.23	37.03	38.02	39.20
33	.00	.00	.00	.00	.00	.26	.59	.46	4.32	9.26	14.69	23.31	28.26	33.23	35.22	36.21	41.02
29	.00	.00	.00	.00	.00	.00	.00	.00	1.53	5.60	10.03	13.08	20.25	26.26	28.20	29.17	33.58
23	.00	.00	.00	.00	.00	.00	.00	.03	2.40	5.28	11.17	13.78	23.31	28.26	25.93	28.20	36.03
25	.00	.00	.00	.00	.00	.00	.00	.00	1.09	4.02	9.82	10.50	18.27	24.02	22.69	24.60	32.60
11	.00	.00	.00	.00	.00	.00	.00	.00	.00	.92	5.04	7.26	12.89	18.58	20.48	21.44	29.03
27	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.46	1.75	4.65	9.63	12.03	13.79	15.56
22	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.12	2.04	5.76	5.94	7.03	12.96
32	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.84	3.68	7.56	9.60	11.25
7	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	3.20	1.89	2.72	5.26
8	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.06	.24	1.25
30	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.75
31	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.36

APPENDIX 3 cont.

A list of all tasks in order of increasing difficulty, as shown in the chi-squared matrix, is given below.

TASK NO.	TASK NAME.
1	N-CNT
10	N-TI-EQ
34	LR-TI-EQ
21	LR-CARD
14	LR-BinA
20	L-P/W
9	N-(ltol)
17	LR-INVAR-ADD
2	N-ADD-V
18	LR-ORD
3	N-SUB-V
12	N-ORD
15	L-INVAR-ADD
13	N-CONS
16	LR-INVAR-SP
24	L-CONS
4	N-SOL-V
5	N-BAL-V
6	N-ADD-NV
26	L-EST
28	L-UNIT
33	DR-M
29	L-UNIT-CH
23	LR-TI-CARD
25	D-CONS
11	N-TI-NE
27	D-EST
22	LR-TI-NE
32	D-M
7	N-SUB-NV
8	N-CYC-NV
30	LR-M-CARD
31	L-M-ADD
19	L-ADD.

APPENDIX 4.

DESCRIPTIONS OF PRODUCTION-SYSTEM MODELS.

A4.1 INTRODUCTION.

Listings and execution traces for all models are attached at Addenda 1 to 6. All models are written in PSS (Ohlsson,1980), a variant of PSG. PSS is a Stanford LISP preprocessor.

The models were processed on a DEC-10 system at the Coombs Computer Centre, Australian National University.

In these models, goal manipulation procedures are not grouped together in one or two common-servicing productions, as is usual with production-systems. Instead, they are located separately in productions which trigger particular goal activation, re-activation, suspension and deletion operations. This has been done to assist the reader, who is not familiar with LISP-type languages, to gain an appreciation of the operation of the models. This reduced the programming elegance of the models. However, it did not result in any greater demands being placed on STM, and did not increase the total number of productions fired during execution.

Firstly, the segments common to each model are described briefly. Secondly, the segments specific to each model are commented upon.

A4.2 COMMON SEGMENTS.

The productions which simulate the first phase of N-ADD-NV, N-SUB-NV, and N-CYC-NV are common to all models. In the first phase, the subject is asked to send `n' balls to the bottom jar by pressing the button. When `n' balls have been sent to the bottom jar, the subject is asked how many were there before (and/or in the top jar, depending upon the particular task), and how many more had just been sent down.

An understanding of the simulation of the first phase may be gained by reading the listing for ADD6.PSS(Addendum 1). This is a performance model of N-ADD-NV that uses a table-look-up procedure.

The productions P000, P00 and P0 control the entry of task information into STM, and initiate the run. When that information is entered, the model responds by simulating the subject's button pressing and counting behaviour. The productions responsible are labelled P1 to P8.

P1 inserts an active goal of sending `n' balls down the tube. In the service of that goal, P2 inserts the subordinate goal of pushing the button. P3 notices the ball going down the tube and inserts the subitization goal. P4 simulates subitization of the ball(s) noticed (by P3) going down the tube. P5 carries out a counting operation by accessing and `saying' the next name on a number-name-list. P6 marks the name `said' by P5, and inserts the next number name on the list into STM. P6A to P6C simulate similar marking and moving operations. P7 simulates a checking operation. If the last name `said' by P5 is

the same as the number-name given in the instruction to send `n' balls down the tube (represented in STM by (SEND C)), then the model `knows' that it has finished that part of the task. In that event, it re-activates the goal `to attend' and the next production to fire is be P0. If not, the goal manipulation in P8 ensures that P2 will be the next production to fire, and that a new cycle of button pressing, , subitizing, and counting will be entered. This procedure can be followed by reading the listing for each model concurrently with the trace of the model's execution.

When control is passed back to P0, the user then asks: how many balls were in the bottom jar before? This is represented by the STM element (MANY BOTTOM BEFORE). P9 and P10 simulate the answering of this question, after which control is passed back to P0. The user then asks: how many were sent down the tube? This is represented in STM by the element (MANY TUBE). This question is answered by P11 and P12, after which control is again passed back to P0. The user then asks: how many balls are now in the bottom jar? This is represented in STM by (MANY BOTTOM NOW). That marks the end of the first phase.

In the subtraction models (ADD5.PSS and ADD4.PSS listed in Addenda 4 and 5, respectively), P9 and P10 answer the question: how many were in the top jar? This is represented in STM by the element (MANY TOP BEFORE)). In the addition and subtraction model (ADD7.PSS listed in Addendum 6), P9 and P10 answer the (MANY BOTTOM BEFORE) question, and PA9 and PA10 answer the (MANY TOP BEFORE) question.

The segments specific to each model are described briefly in the following sections.

A4.3 TABLE-LOOK-UP PROCEDURES.

The addition model (ADD6.PSS) and the subtraction model (ADD5.PSS) use table-look-up procedures. These procedures are also used in the addition and subtraction model (ADD7.PSS).

A4.3.1 ADDITION.

An understanding of the addition procedure may be gained by reading the listing for ADD6.PSS, in conjunction with the trace of its execution (Addendum 1).

By entering the question (MANY BOTTOM NOW) into STM the user causes control, on the next cycle, to be given to P13, which activates the addition goal (GOAL * ADD). P14 activates the goal of retrieving from LTM the list containing the results of adding `n' balls to the `m' already in the bottom jar. P15 and P15C simulate the entry of those lists to STM. P16 `reads off' the answer. For example, suppose the bottom jar contained two balls and the subject sent four more down the tube. Then P14 would activate the goal of retrieving from LTM the relevant list (LIST 2), and P15A would insert the entries (1->3),(2->4),(3->5),(4->6),etc. P16 would select the entry (4->6), and extract from it the answer, 6.

The remaining productions, P24A to P27 are responsible for housekeeping functions needed to prepare the model to receive more input.

A4.3.2 SUBTRACTION.

An understanding of the subtraction procedure may be gained by reading the listing for ADD5.PSS in conjunction with the trace of its execution (Addendum 4).

Productions P14 to P16 are responsible for the subtraction table-look-up procedure. They are similar in form to their counterparts in the addition model. Hence, an example should be sufficient to explain their function.

Suppose the top jar had six balls in it, and then the subject sent three balls to the bottom jar. P14 would activate the goal of retrieving (LIST 6). P15A would insert the entries from that list, such as (2->4),(3->3), etc. P16 would select the entry (3->3), and `read-off' the answer, 3.

A4.4 DIRECT COUNTING PROCEDURE.

The addition model, ADD8.PSS, and the subtraction model, ADD4.PSS, use direct counting procedures. They are used instead of table-look-up methods for simulating the answers to the (MANY BOTTOM NOW) and (MANY TOP NOW) questions.

A4.4.1 ADDITION.

The direct counting procedure involves the co-ordination of step-by-step movement through two sequences, each of which is the number-name-list. Items from the first sequence are represented in STM by elements having the form (Y <P> Z), and those from the second sequence by (W <P> V). The letters Y,Z W and V imbedded in these elements are of technical significance only. They constitute a method of marking locations in a list. The symbol <P> assumes numerical values.

P13 activates the addition goal (GOAL * ADD). It sets the location in the first sequence at the point corresponding to the number of balls in the bottom jar before the last series of button presses. P13 also sets the location in the second sequence, just before the first element in the number-name-list. P14 checks to see if the location in the second sequence is the same as the value in the STM element representing the instruction to send `n' balls down the tub. (Essentially, is the value of <P> in the element (W <P> V) the same as the value of <P> in the element (SEND <P>)?) If so, P14 fires, and the answer is extracted from the (Y <P> Z) element currently in STM. If not, either P15 or P16 fires. They control the movement through the two sequences. P17 to P17J carry out the moves from place to place in the first sequence. P18 to P18C perform the same function for the second sequence.

An example may clarify the operation of P13 to P18C. Suppose that the bottom jar had two balls in it, and the subject sent down four more. STM would contain the elements (BOTTOM 2) and (SEND 4). P13 would insert the ele-

ments (Y 2 Z) and (W 0 V). On the first cycle after the firing of P13, P14 would not fire because <P> would be set to 0 in (W 0 V). P15 would initiate an entry to the P17 to P17J group of productions. Specifically, P17B would fire and insert (Y 3 Z) into STM, and set a goal causing P16 to fire on the next cycle. P16 would then initiate an entry to the P18 to P18C group of productions. Specifically, P18 would fire and insert (W 1 V) into STM, and set a goal causing P14's conditions to be examined on the next cycle. Again, P14 would not fire, because <P> would be set to 1 in (W 1 V). Hence, P15 would fire again, and the P15 to P18C procedure would be re-entered. This pattern would continue until on one cycle P14 found <P> set to 4 in (W 4 V). At that time, the (Y <P> Z) element would contain (Y 6 Z). P14 would then extract the answer (6) from that element.

The remaining productions P24A to P27 perform housekeeping functions needed to prepare the model to receive further input.

A4.4.2 SUBTRACTION.

The productions P13 to P18C in the subtraction model (ADD4.PSS) carryout analogous functions. The differences are that:-

- P13 sets <P> in (Y <P> Z) to equal the number of balls in the top jar, initially.
- P17 to P17J moves down the number-name-list, not up it, as in the case of the addition model (ADD8.PSS).

A4.5 MODIFIED-COUNTING PROCEDURE.

The addition model ADD3.PSS uses a modified-counting procedure. In this model, the larger of the two numbers to be added (represented in STM by the elements (BOTTOM B) and (SEND C) is found, and used to set the initial location in the first sequence. Groen and Parkman (1972) argued that this procedure is used by young children, and some adults.

In this model, productions P13 to P16 control the finding of the smaller and larger of the two addends. Productions P17 to P17K simulate movement through the first sequence. Productions P230 to P230C simulate movement through the second sequence. Productions P180 to P210 extract the smaller and larger addends and set the initial locations in the two sequences. P220 fires when movement through the two sequences has resulted in the answer being held in the (Y <P> Z) STM element. The remaining productions simulate either special cases or housekeeping functions. The two special cases are:-

- equal addends - P130A
- one addend of zero - P130B

APPENDIX 5.

COMPARATIVE ANALYSIS OF RESULTS USING ASSESSMENT CRITERIA VARYING IN DEGREE OF STRICTNESS.

A5.1 CORRELATION BETWEEN NUMBER OF QUESTIONS ASKED AND ORDER OF TASK DIFFICULTY.

The analysis in Chapter 10 was based on a strict assessment criterion. In order to pass a task the child had to answer every question correctly and, in many instances, give a correct verbal statement of the reasoning employed. However, every child was also given a number of opportunities to reconsider an incorrect answer. The child was prodded with comments like: "Are you sure? Think about it again. Slowly this time, and see if you get the same answer." Thus, a clinical style of questioning was used to assess the child's knowledge.

An important aspect of this approach is that there is substantial variation between tasks in the number of questions asked. This is because it was necessary to relate the number and type of question asked to the nature of the task. Some tasks (eg. DR-M) required the child to demonstrate by physical action certain kinds of knowledge. Other tasks required only an evaluative answer (eg. "longer" or "shorter"), supported by verbal statements of the reasoning employed. Still other tasks required knowledge of number facts within a certain range.

In these circumstances, it may be thought that the observed orders of difficulty of the collections of number and length tasks reported in Chapter 10 could reflect simply the variation between tasks in the number of questions asked. A correlation analysis was carried out to investigate this possibility.

A5.1.1 NUMBER TASKS

In the case of the number tasks, the number of questions asked ranged from 2 to 8. The degree of association between the number of questions asked and the rank order of difficulty was assessed by computing the Spearman rank correlation statistic Rs (corrected for ties), which is .37. This is not significant at the .05 level (sample t=1.32, criterion t=1.8 at alpha = .05 and 11 d.f. for a one tailed test).

A5.1.2 LENGTH TASKS

In the collection of length tasks, the number of questions asked ranged from 1 to 10. The relevant Spearman rank correlation coefficient is -.42. This is significant at the .05 level (sample t=1.8, criterion t=1.75, at alpha=.05 and 15 d.f. for a one tailed test) but the direction of association between number of questions and task difficulty is the opposite to that which might have been expected. In other words, this analysis reveals that the fewer the number of questions the greater the difficulty of the task.

It may be concluded, therefore, that the orders of difficulty for the number and length task collections do not reflect simply variations in the number of questions asked.

A5.2 EFFECT OF VARYING STRINGENCY OF SCORING CRITERION ON ORDER OF TASK DIFFICULTY.

This conclusion notwithstanding, it may still be objected that the use of a strict assessment criterion in developmental research is inappropriate because, as Brainerd (1978) argues, it increases the risk of false negatives. By the same token, the use of a lax criterion increases the risk of false positives. As Langford (1981) has noted, the former is a fairer test of hypotheses concerning synchronous development, while the latter is more effective in detecting sequential development. In order to gauge the effect of the use of a strict criterion in the present research, the data were also scored using "moderate" and "weak" criteria.

A5.2.1 MODERATE CRITERION.

Under the moderate criterion a child was assessed as having passed the task if he answered the first evaluative question correctly, and gave the correct verbal explanation. For those tasks where verbal explanations were not requested the child was required to answer the first two evaluative questions correctly.

A5.2.2 WEAK CRITERION.

Under the weak criterion a child was assessed as having passed the task if he gave the correct evaluative response to the first question asked.

A5.2.3 DISTRIBUTION OF TOTAL SCORES.

The distribution of total scores for all subjects, and of the number of subjects passing each task, for both moderate and weak scoring procedures,

are depicted in Figures A5.1 to A5.4. When these distributions are compared with the corresponding distributions obtained under the strict scoring regime (given at Figures 10.1 and 10.2, pages 126a and b of the main text), it is apparent that the minor floor effect evident under the strict criterion assessment is exaggerated. For example, under the weak criterion 64% of the subjects scored 24 or more (out of 34); under the moderate criterion, 48% scored 24 or more; and under the strict criterion, 36% scored 24 or more. This outcome is, of course, predictable, since increasing the probability of false positives increases the probability of a floor effect.

A5.2.4 ORDER OF TASK DIFFICULTY

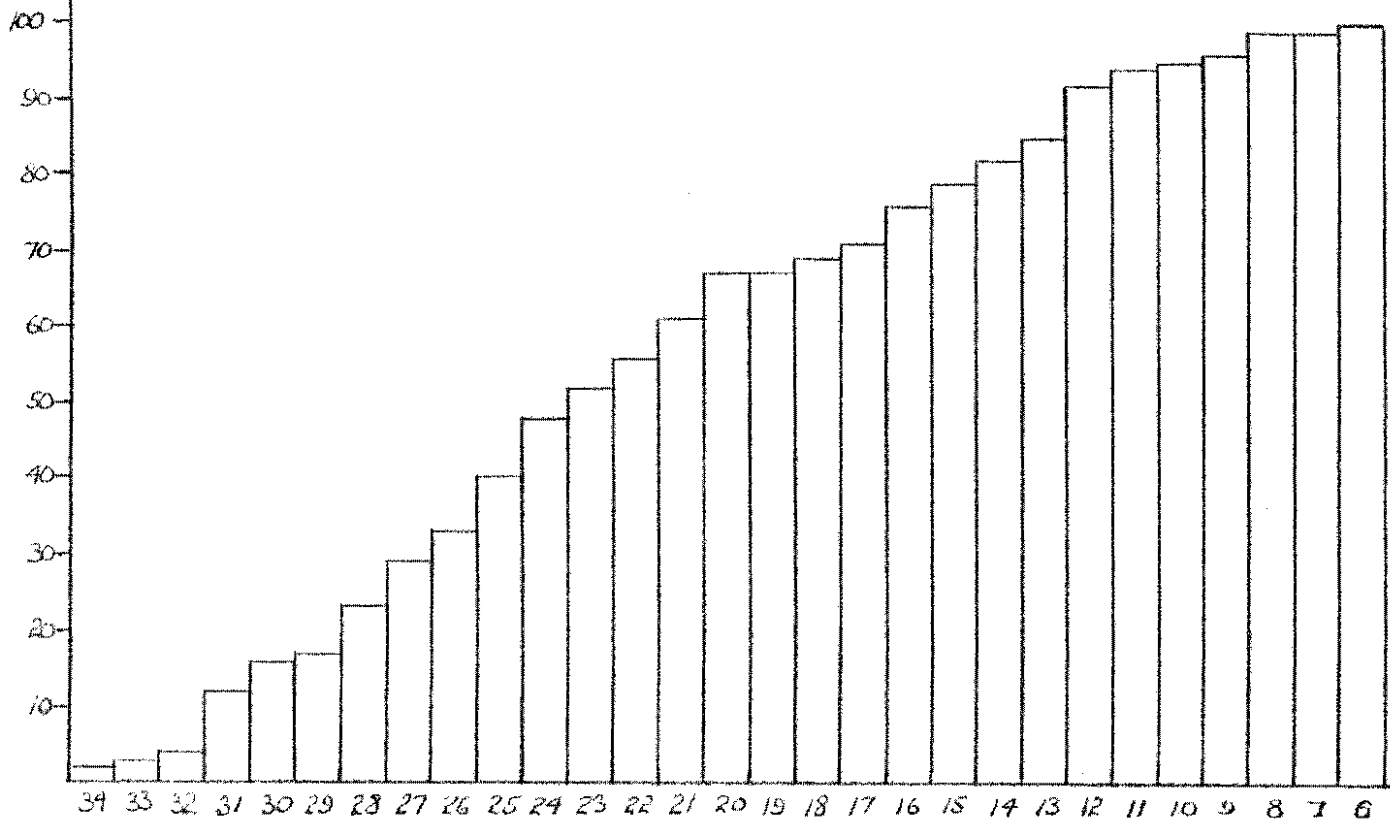
The orders of difficulty obtained under each scoring procedure are given in Table A5.1 for all tasks, in A5.2 for the number tasks, in A5.3 for the length tasks, and in A5.4 for the distance tasks. The general impression obtained from inspection of these tables is that a similar ranking is obtained under all three procedures. This impression is confirmed by the Spearman rank correlation statistics given in Table A5.5. These values indicate that for all tasks taken as one collection, for the collection of number tasks, and for the collection of length tasks, the rank orderings obtained under the strict, moderate and weak criteria are not significantly different. The orderings obtained under the three scoring procedures are significantly (alpha=.05) different, however, for the collection of distance tasks.

The differences for the distance tasks are due to the small number of tasks involved. Since the collection contains only four tasks, a reversal of order between two adjacently ranked tasks, which may differ only slightly in the number of subjects passing each, has the effect of changing 50% of the ranks. This effect is obvious in the case of DR-M and D-CONS.

FIGURE AS.1

DISTRIBUTION OF TOTAL SCORES

MODERATE CRITERION



Each bar indicates no. of subjects achieving a total score of at least N

FIGURE A5-2

DISTRIBUTION OF TASK DIFFICULTY

MODERATE CRITERION

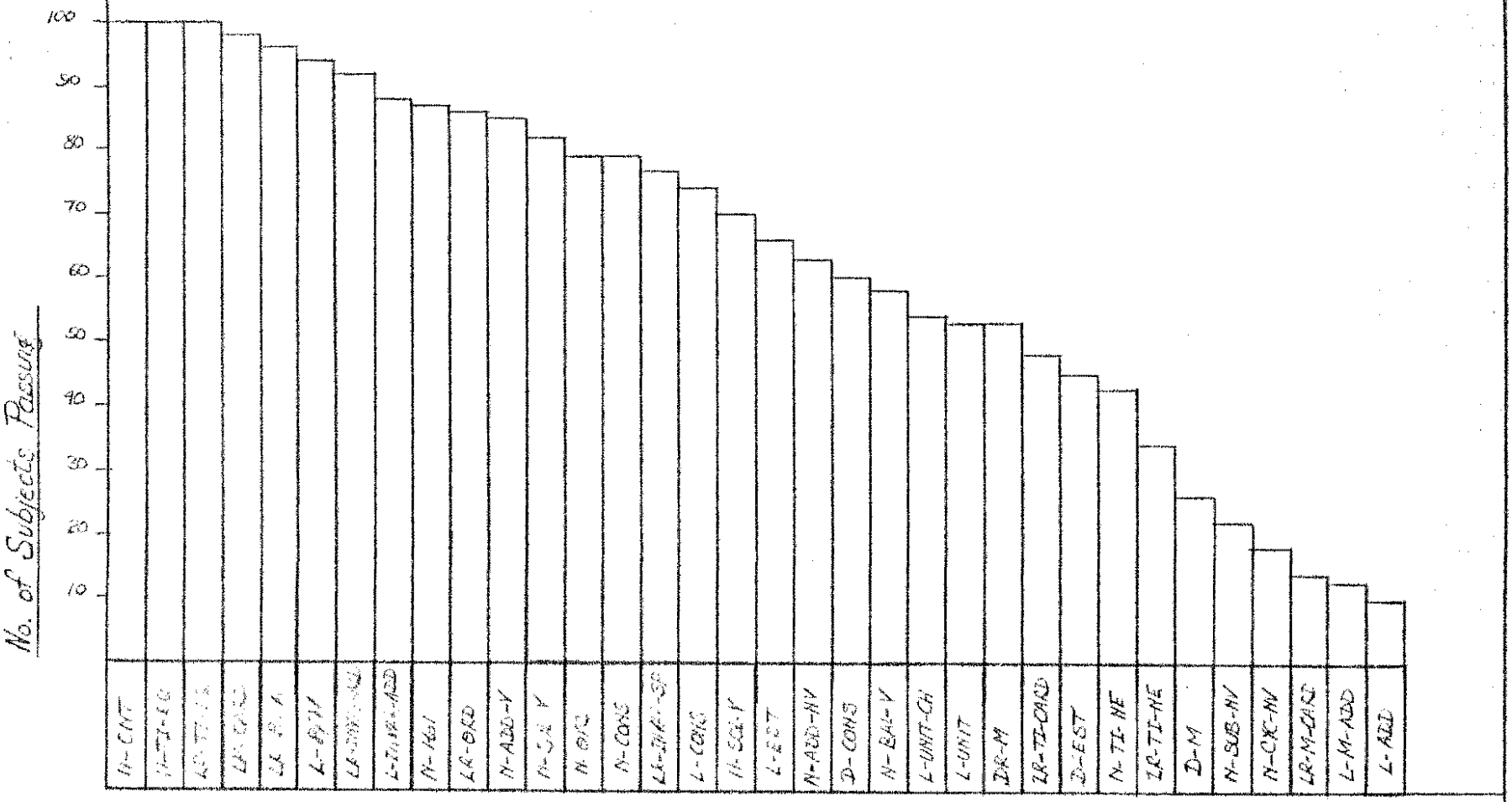
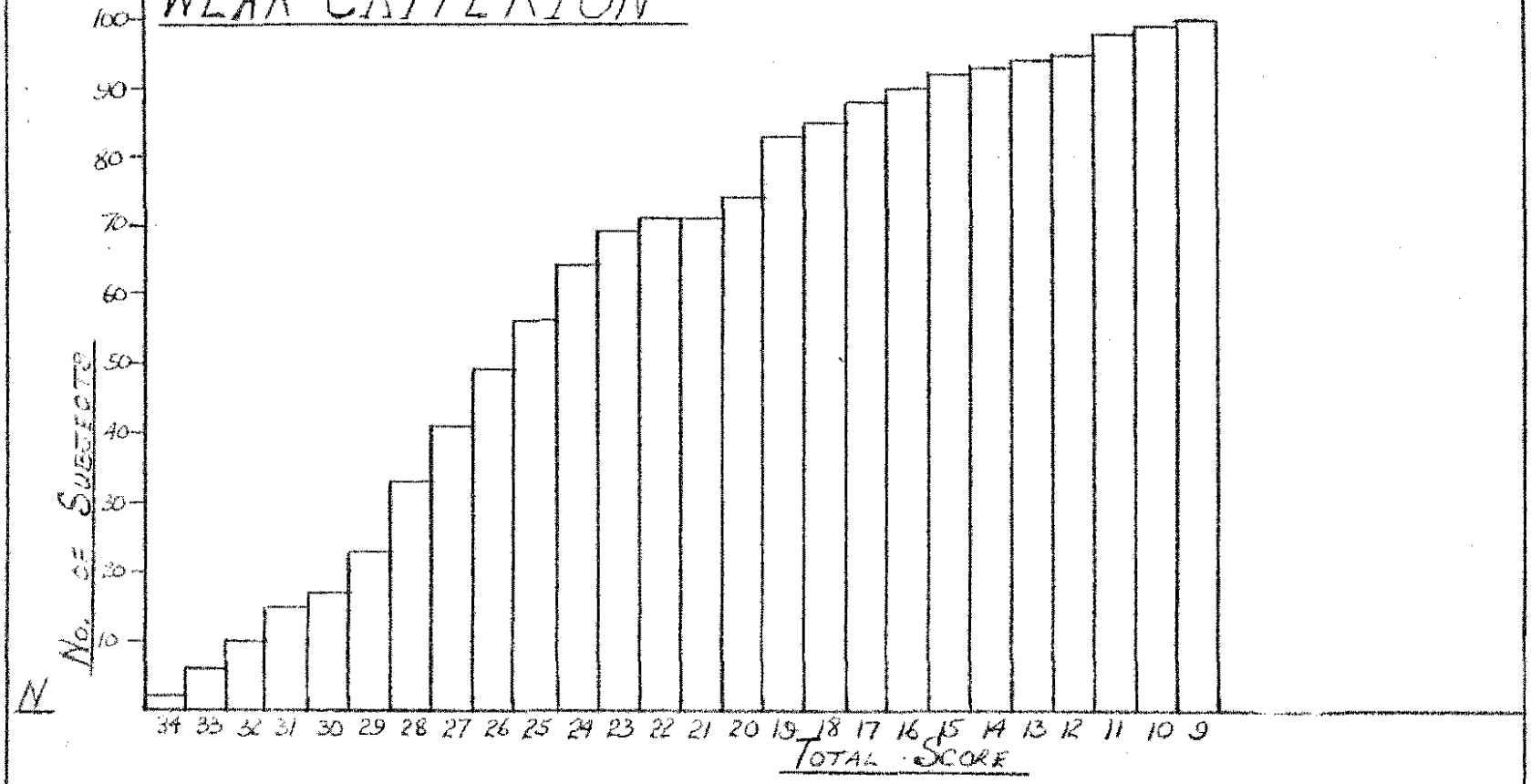


FIGURE A5.5

DISTRIBUTION OF TOTAL SCORES.

WEAK CRITERION

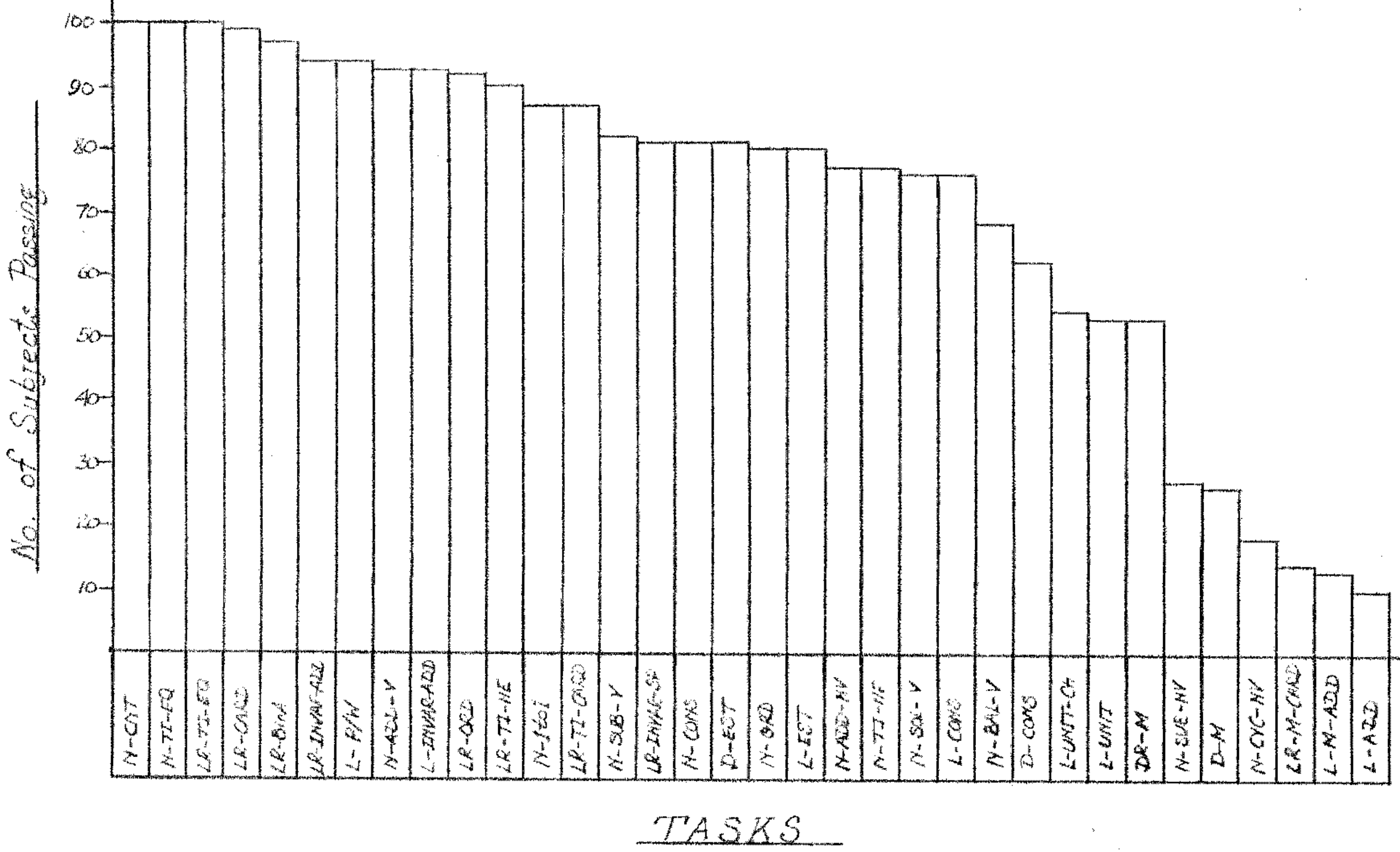


Each bar indicates no. of subjects achieving a total score of at least N.

FIGURE NO.

DISTRIBUTION OF TASK DIFFICULTY

WEAK CRITERION



Under the strict scoring criterion 53 subjects passed the former while 48 passed the latter, yielding ranks of 1 and 2, respectively. Under the moderate scoring criterion, 53 subjects passed DR-M and 65 passed D-CONS, yielding a reversal of order in ranking.

TABLE A5.1: ALL TASKS: ORDER OF TASK DIFFICULTY OBTAINED UNDER THE STRICT, MODERATE AND WEAK SCORING CRITERIA.

TASK	STRICT CRITERION		MODERATE CRITERION		WEAK CRITERION	
	NO.OF Ss. PASSING*	OBSERVED RANK	NO.OF Ss. PASSING	OBSERVED RANK	NO OF Ss. PASSING	OBSERVED RANK
N-CNT	100	1	100	1	100	1
N-TI-EQ	100	1	100	1	100	1
LR-TI-EQ	100	1	100	1	100	1
LR-CARD	98	2	98	2	99	2
LR-BinA	95	3	96	3	97	3
L-P/W	94	4	94	4	94	4
N-1tol	86	5	87	7	87	8
LR-INVAR-ADD	85	6	92	5	94	4
N-ADD-V	84	7	85	9	93	5
LR-ORD	82	8	86	8	92	6
N-SUB-V	81	9	82	10	84	9
N-ORD	80	10	79	11	80	12
L-INVAR-ADD	80	10	88	6	93	5

N-CONS	78	11	79	11	81	11
LR-INVAR-SP	74	12	77	12	82	10
L-CONS	74	12	74	13	76	14
N-SOL-V	61	13	70	14	76	14
N-BAL-V	58	14	58	18	68	15
N-ADD-NV	58	14	63	16	77	13
L-EST	56	15	66	15	80	12
L-UNIT	53	16	53	20	53	18
DR-M	53	16	53	20	53	18
L-UNIT-CH	49	17	54	19	54	17
LR-TI-CARD	48	18	48	21	87	8
D-CONS	48	18	60	17	62	16
N-TI-NE	41	19	43	23	77	13
D-EST	34	20	45	22	81	11
LR-TI-NE	29	21	34	24	90	7
D-M	26	22	26	25	26	20
N-SUB-NV	21	23	22	26	27	19
N-CYC-NV	16	24	18	27	18	21
LR-M-CARD	14	25	14	28	14	22
L-M-ADD	13	26	13	29	13	23
L-ADD	10	27	10	30	10	24

NOTES: *Maximum of 100

TABLE A5.2: NUMBER TASKS: ORDER OF TASK DIFFICULTY OBTAINED UNDER THE STRICT, MODERATE AND WEAK SCORING CRITERIA.

TASK	STRICT CRITERION		MODERATE CRITERION		WEAK CRITERION	
	NO. OF Ss. PASSING*	RANK	NO. OF Ss. PASSING	RANK	NO. OF Ss. PASSING	RANK
N-CNT	100	1	100	1	100	1
N-TI-EQ	100	1	100	1	100	1
N-1tol	86	2	87	2	87	3
N-ADD-V	84	3	85	3	93	2
N-SUB-V	81	4	82	4	84	4
N-ORD	80	5	79	5	80	6
N-CONS	78	6	79	5	81	5
N-SOL-V	61	7	70	6	76	8
N-BAL-V	58	8	58	8	68	9
N-ADD-NV	58	8	63	7	77	7
N-TI-NE	41	9	43	9	77	7
N-SUB-NV	21	10	22	10	27	10
N-CYC-NV	16	11	18	11	18	11

NOTES: * Maximum of 100.

TABLE A5.3: LENGTH TASKS: ORDER OF TASK DIFFICULTY OBTAINED UNDER THE STRICT, MODERATE AND WEAK SCORING CRITERIA.

TASK	STRICT CRITERION		MODERATE CRITERION		WEAK CRITERION	
	NO. OF Ss. PASSING *	RANK	NO. OF Ss. PASSING	RANK	NO. OF Ss. PASSING	RANK
LR-TI-EQ	100	1	100	1	100	1
LR-CARD	98	2	98	2	99	2
LR-BinA	95	3	96	3	97	3
L-P/W	94	4	94	4	94	4
LR-INVAR-ADD	85	5	92	5	94	4
LR-ORD	82	6	86	7	92	6
L-INVAR-ADD	80	7	88	6	93	5
LR-INVAR-SP	74	8	77	8	82	9
L-CONS	74	8	74	9	76	11
L-EST	56	9	66	10	80	10
L-UNIT	53	10	53	12	53	13
L-UNIT-CH	49	11	54	11	54	12
LR-TI-CARD	48	12	48	13	87	8
LR-TI-NE	29	13	34	14	90	7
LR-M-CARD	14	14	14	15	14	14
L-M-ADD	13	15	13	16	13	15
L-ADD	10	16	10	17	10	16

NOTES: *Maximum of 100.

TABLE A5.4: DISTANCE TASKS: ORDER OF TASK DIFFICULTY OBTAINED UNDER THE STRICT, MODERATE AND WEAK SCORING CRITERIA.

TASK	STRICT CRITERION		MODERATE CRITERION		WEAK CRITERION	
	NO. OF Ss. PASSING *	RANK	NO. OF Ss. PASSING	RANK	NO. OF Ss. PASSING	RANK
DR-M	53	1	53	2	53	3
D-CONS	48	2	60	1	62	2
D-EST	34	3	45	3	81	1
D-M	26	4	26	4	26	4

NOTES: * Maximum of 100

TABLE A5.5: COMPARISON OF RANK ORDERINGS OBTAINED UNDER THE STRICT, MODERATE AND WEAK SCORING CRITERIA.

TASK COLLECTION	RANK ORDERINGS COMPARED								
	STRICT WITH MODERATE			STRICT WITH WEAK			MODERATE WITH WEAK		
	Rs	t	P	Rs	t	P	Rs	t	P
ALL TASKS	.989	53.79	<.001	.87	13.38	<.001	.88	14.30	<.001
NUMBER TASKS	.997	60.34	<.001	.97	18.57	<.001	.98	21.87	<.001
LENGTH TASKS	.99	49.67	<.001	.90	11.02	<.001	.91	11.67	<.001
DISTANCE TASKS	.80	2.53	>.05	.20	.32	>.05	.40	.73	>.05

A5.3 EFFECT OF VARYING STRINGENCY OF SCORING CRITERION ON SCALABILITY OF NUMBER AND LENGTH TASK COLLECTIONS.

Since it is evident that adopting a less stringent criterion has the effect of distorting the distribution of total scores, though without altering significantly overall task difficulty ranking, the possibility was investigated that the number and length task collections may not form scaled sets under the moderate and weak scoring criteria. The results are summarised in Table A5.6 which sets out the relevant Guttman and Loevinger indices. If more weight is accorded the Loevinger index of homogeneity than the Guttman co-efficient of scalability, the effect of adopting a less stringent criterion is to reduce marginally overall test homogeneity, and to increase the incidence of chance level responding. However, as is shown in Table A5.6, the number and length task collections form scaled sets, whether assessment is based upon a strict, moderate or weak criterion.

TABLE A5.6: SCALING INDICES OBTAINED FROM DATA DERIVED FROM STRICT, MODERATE AND WEAK SCORING CRITERIA.

TASK COLLECTION	CRITERION	GUTTMAN CO-EFFICIENT OF SCALABILITY	LOEVINGER INDEX OF HOMOGENEITY	LOEVINGER INDEX OF HOMOGENEITY OF AN ITEM WITH A TEST	LOEVINGER INDEX OF HOMOGENEITY OF AN ITEM WITH AN ITEM
NUMBER	STRICT	.589	.570	ALL 13 EXCEED .7 (a)	70 OUT OF 78 EXCEED .5 (b)
	MODERATE	.568	.685	ALL 13 EXCEED .7	69 OUT OF 78 EXCEED .5
	WEAK	.401	.540	12 OUT OF 13 .7 (c)	56 OUT OF 78 EXCEED .5
LENGTH	STRICT	.48	.58	14 OUT OF 17 EXCEED .7	101 OUT OF 136 EXCEED .5
	MODERATE	.455	.563	13 OUT OF 17 EXCEED .7	93 OUT OF 136 EXCEED .5
	WEAK	.431	.539	16 OUT OF 17 EXCEED .7 (d)	98 OUT OF 136 EXCEED .5.

NOTES: (a) A perfectly homogeneous item would have a H(it) value of 1 but values of .7 and higher are regarded as acceptable.

(b) Chance level responding is indicated by a h(ii) value of .5.

(c) The H(it) for N-TI-NE = .66.

(d) The H(it) for LR-TI-NE = .33.

A5.4 EFFECT OF VARYING STRINGENCY OF SCORING CRITERION ON CONSERVATION AND TRANSITIVITY TASK DIFFICULTY RANKINGS.

Much of the controversy surrounding the use of a strict or weak scoring criterion has centred on the classical Piagetian tests for transitivity and conservation. Therefore, the effect of varying scoring criterion on those tasks was examined separately from the issue of overall task difficulty ranking. Table A5.7 sets out the number of subjects passing the various conservation and transitivity tests under each scoring procedure. When the results obtained under the strict criterion are compared with those for the moderate criterion, the same orderings of task difficulty and the same differences in task difficulty - with the exception of D-CONS- emerge. In the case of D-CONS, an analysis of subject protocols revealed that the majority of those failing that task had most difficulty with the second and third variant of the task, rather than in giving verbal accounts of the reasoning involved.

If the orderings of task difficulty obtained under the weak criterion are compared with those gained under the strict and moderate procedures, large differences emerge for the transitivity but not the conservation tasks. This is most evident in the case of LR-TI-NE. Assessed under the strict criterion only 29 subjects passed the task, under the moderate criterion 34 passed, but under the weak criterion 90 passed. This indicates the difficulty subjects experienced in giving a verbal account of the reasoning they employed in reaching their answers.

TABLE A5.7: NUMBER OF SUBJECTS PASSING CONSERVATION AND TRANSITIVITY TASKS ACCORDING TO SCORING CRITERION USED.

TASKS	NO. OF SUBJECTS PASSING		
	STRICT CRITERION	MODERATE CRITERION	WEAK CRITERION
N-CONS	78	79	81
N-TI-NE	41	43	77
L-CONS	74	74	76
LR-TI-CARD	48	48	87
LR-TI-NE	29	34	90
D-CONS	48	60	62

A5.5 SUMMARY OF EFFECTS OF VARYING STRINGENCY OF SCORING CRITERION ON TASK DIFFICULTY RANKINGS.

The main findings of the analysis based upon the strict scoring criterion are summarised at Section 10.11 (pages 152 to 154) of the main report, under the headings "Components of Linear Measurement", "Order of Development of Linear Measurement", and "Expected Pattern of Development." It will be apparent from the immediately preceding discussion of the effects of varying scoring criterion, that the conclusions reported in Chapter 10 do not rest solely, or even in large measure, upon the particular scoring approach adopted.

A5.6 EFFECT OF VARYING STRINGENCY OF SCORING CRITERION ON DEVELOPMENTAL DELAYS AND DISCONTINUITIES.

The conclusions reached in Chapter 10 were further examined in Chapter 11. This examination focussed, in the main, on the delay between acquisition of the assumed components of linear measurement and the emergence of an understanding of linear measurement. That examination also highlighted the discontinuities in the growth of the number and length concept domains. The extent to which these findings depend upon the scoring criterion used should also be examined.

A5.6.1 DELAY IN ATTAINMENT OF LINEAR MEASUREMENT.

Regarding the components of linear measurement, an inspection of Table A5.1 indicates that the components listed at Section 11.7 (page 180) of the main text (derived from the strict scoring criterion) are the same as those which would be yielded by an analysis based on moderate criterion. If a weak criterion were to be used then "knowing how to make transitive inferences of non-equivalence, with respect to length" would have to be added to that list.

Additionally, as noted at page 163 of the main text, under the strict scoring criterion it was found that 13 subjects passed all high-order component tasks but failed the two linear measurement benchmark tasks, LR-M-CARD and L-M-ADD. This was interpreted as indicating a delay between acquiring the underlying components and being able to demonstrate a mature understanding of linear measurement. The same interpretation could be made under the moderate and weak scoring procedures. Using the former, 22 subjects passed all high-order components but failed LR-M-CARD and L-M-ADD, whilst, using the latter, 29 passed the component tasks but failed the two benchmark tasks.

A5.6.2 DISCONTINUITIES IN GROWTH PATTERNS.

Regarding the discontinuities in the growth of the number and length concept domains, Tables A5.8 to A5.11 set out the orders of emergence of the assumed components of the number and length concepts, assessed under moderate and weak criteria. These tables should be compared with Tables 11.1 (page 167) and 11.3 (page 174) in the main text.

Considering the number concept first, under the strict scoring criterion a stepped performance gradient having five levels was detected. Under the moderate criterion, as is indicated by the chi-squared values in Table A5.8, a stepped performance gradient is also apparent but the five levels are reduced to four. Under the weak criterion (Table A5.9), the stepped performance gradient is reduced to three levels.

In the case of the length concept, under the strict criterion four levels of a stepped performance gradient were detected. Under both moderate (Table A5.10) and weak (Table A5.11) scoring criterion the stepped performance gradient is reduced to three levels.

TABLE A5.8 NUMBER TASKS - MODERATE CRITERION: CHI-SQUARED VALUES FOR ADJACENTLY RANKED ITEM PAIRS.

TASK	NO. OF SUBJECTS PASSING	McNEMAR CHI-SQUARED VALUES	P.
N-CNT	100		
		→ 0.00	NS
N-TI-EQ	100		
		→ 13.90	<.001
N-1tol	87		
		→ 0.17	NS
N-ADD-V	85		
		→ 0.33	NS
N-SUB-V	82		
		→ 0.29	NS
N-ORD	79		
		→ 0.00	NS
N-CONS	79		
		→ 2.13	NS
N-SOL-V	70		
		→ 1.10	NS
N-ADD-NV	63		
		→ 0.52	NS
N-BAL-V	58		
		→ 4.50	<.05
N-TI-NE	43		
		→ 10.05	<.005
N-SUB-NV	22		
		→ 0.50	NS
N-CYC-NV	18		

TABLE A5.9 NUMBER TASKS -WEAK CRITERION: CHI-SQUARED VALUES FOR ADJACENTLY RANKED ITEM PAIRS.

TASKS	NO. OF SUBJECTS PASSING	McNEMAR CHI-SQUARED VALUES	P
N-CNT	100		
		→ 0.0	NS
N-TI-EQ	100		
		→ 7.25	<.01
N-ADD-V	93		
		→ 2.00	NS
N-1to1	87		
		→ 0.36	NS
N-SUB-V	84		
		→ 0.31	NS
N-CONS	81		
		→ 0.03	NS
N-ORD	80		
		→ 0.27	NS
N-ADD-NV	77		
		→ 0.00	NS
N-TI-NE	77		
		→ 0.03	NS
N-SOL-V	76		
		→ 1.59	NS
N-BAL-V	68		
		→ 33.70	<.001
N-SUB-NV	27		
		→ 2.32	NS
N-CYC-NV	18		

TABLE A5.10 LENGTH TASKS - MODERATE CRITERION: CHI-SQUARED VALUES FOR ADJACENTLY RANKED ITEM PAIRS.

TASK	NO. OF SUBJECTS PASSING	McNEMAR CHI-SQUARED VALUES	P
LR-TI-EQ	100		
		→ 2.02	NS
LR-CARD	98		
		→ 0.69	NS
LR-BinA	95		
		→ 0.42	NS
L-P/W	94		
		→ 0.31	NS
LR-INVAR-ADD	92		
		→ 0.89	NS
L-INVAR-ADD	88		
		→ 0.18	NS
LR-ORD	86		
		→ 2.69	NS
LR-INVAR-SP	77		
		→ 0.24	NS
L-CONS	74		
		→ 1.52	NS
L-EST	66		
		→ 3.00	NS
L-UNIT-CH	54		
		→ 0.02	NS
L-UNIT	53		
		→ 0.50	NS
LR-TI-CARD	48		
		→ 4.05	<.05
LR-TI-NE	34		
		→ 10.96	<.005
LR-M-CARD	14		
		→ 0.04	NS
L-M-ADD	13		
		→ 0.44	NS
L-ADD	10		

TABLE A5.11 LENGTH TASKS - WEAK CRITERION: CHI-SQUARED VALUES FOR ADJACENTLY RANKED ITEM PAIRS.

TASK	NO. OF SUBJECTS PASSING	McNEMAR CHI-SQUARED VALUES	P
LR-TI-EQ	100		
		1.01	NS
LR-CARD	99		
		1.02	NS
LR-BinA	97		
		1.05	NS
L-P/W	94		
		0.00	NS
LR-INVAR-ADD	94		
		0.08	NS
L-INVAR-ADD	93		
		0.07	NS
LR-ORD	92		
		0.24	NS
LR-TI-NE	90		
		0.44	NS
LR-TI-CARD	87		
		0.95	NS
LR-INVAR-SP	82		
		0.13	NS
L-EST	80		
		0.47	NS
L-CONS	76		
		10.64	<.005
L-UNIT-CH	54		
		0.02	NS
L-UNIT	53		
		34.14	<.001
LR-M-CARD	14		
		0.04	NS
L-M-ADD	13		
		0.44	NS
L-ADD	10		

A5.6.3 SUMMARY

In summary, the two main conclusions reached in Chapter 11 - delay between acquisition of assumed components and attainment of concept, and discontinuities in growth of the number and length concept domains - are essentially unaffected by adopting less stringent [illegible]

A5.7 CONCLUSION

Finally, on the basis of the evidence discussed in this Appendix, it is reasonable to conclude that the results and findings of the present research as presented in Chapters 10 and 11 of the main text, do not stem in any significant fashion from the use of a strict scoring criterion incorporating wide variation between tasks in the number and type of question asked.

APPENDIX 6.

SPECULATIVE OUTLINE OF FURTHER WORK ON PRODUCTION SYSTEM MODELLING OF THE DEVELOPMENT OF LINEAR MEASUREMENT.

A6.1 INTRODUCTION.

The production systems described in Chapter 13 were constructed largely as a means of testing a SIM hypothesis concerning developmental discontinuities. It was also stated that those systems constituted a beginning of a much larger project aimed at constructing a production system model of the development of iterative linear measurement. This Appendix outlines briefly the direction in which such further work could proceed.

A production system model of the development of linear measurement should exhibit the developmental patterns observed in this study. For example, the model should display a capacity to conserve length before it demonstrates a capacity to `construct' a unit of length for purposes of iteration. A separate sub-system would be constructed for each assumed component of linear measurement. These sub-systems would be state models constituting a developmental hierarchy. The major task would then be to construct a learning, or transition, model which accounted for that hierarchy, and, in particular, for the delay between acquisition of the assumed components and eventual attainment of iterative linear measurement.

A6.2 TYPES OF STATE MODEL REQUIRED.

The nature of these state models of the assumed components of linear measurement requires some elaboration. As indicated in Chapter 2, there is a sense in which, for both number and length, conservation implies transitive reasoning, and transitive reasoning implies conservation. Reflecting that observation, a state model of conservation could provide explicitly for the activation of productions encoding transitive reasoning rules, and a state model of transitive reasoning could provide explicitly for the processing of conservation rules.

Although such models might provide an admirable formal specification of conservation and transitivity, it is unlikely that they would provide an accurate description of the cognitive processes actually invoked by a subject in making a conservation response, or responding correctly to a transitive reasoning task. This is because it is not usually the case that the conserving subject, for example, resorts to 'first principles' in reaching his decision. Instead, it seems that he relies upon a general rule covering a certain class of transformation. First principle statements of the kind that are necessary in the formal derivation of, say, conservation are only given if the subject is required to justify, post hoc, his answer. Piaget and Inhelder (1969) capture this distinction by asserting that conservation is a matter of logical necessity not empirical determination.

Hence, for both number and length, it would seem that, in respect of both conservation and transitive reasoning, at least two types of state model are required: one that contains first principle rules; and one that contains only a general rule. The nature of the developmental relationship between these models should be consistent with the empirical evidence. For example, for both number and length, the first conservation model would incorporate explicit rules concerning transitive processing of equivalence relations (because, in this study, the order of appearance was the capacity assessed by N-TI-EQ then that assessed by N-CONS; LR-TI-EQ then L-CONS). The second conservation model would not contain these rules. In two senses, therefore, the second model would be more mature: firstly, it would appear later in development; and, secondly, it would have had redundant processing - the first principle rules - removed from it, and, to that extent, it would be more efficient. That greater efficiency would be expressed behaviourally in much faster response times. Consequently, the impression would be conveyed that the subject was making a logical assertion, rather than expressing a ponderously determined decision concerning an empirical regularity. It should also be noted that the first model - the one concerning the first principle rules - may not be expressed behaviourally before the subject's responses provide evidence that the second model has been constructed. This is so even though the former is cast as the developmental precursor of the latter. Indeed, the first model may only ever be expressed behaviourally when the subject is required to justify his answers. This point will be returned to in later paragraphs.

This speculation concerning the two-phase development of conservation is also relevant to the development of linear measurement, as assessed by LR-M-CARD. When a subject solves the linear measurement problem represented by that task, it is extremely unlikely that, in the absence of prior, detailed and complete instruction, he would invoke the conservation and/or transitive reasoning models as part of his solution strategy. That is to say, the subject would not ordinarily work out LR-M-CARD from first principles. The conservation and transitive reasoning models would only be invoked if the subject was required to justify his answer. Thus, as with conservation, development of linear measurement would proceed through two (at least) phases. The developmental relationship between the two production system models representing each phase should be consistent with the empirical evidence. That is, the first model would incorporate productions corresponding to conservation and transitivity rules, whilst the second model would not. The latter would be more mature than the former in-so-far as: (a), it would appear later in development; and (b), it would have had redundant processing - the conservation and transitive reasoning rules - removed from it. To that extent it would also be more efficient, and yield faster solution times.

Again, as was the case with conservation, the first model of linear measurement - the one containing the conservation and transitive reasoning rules - may only be expressed behaviourally when the subject is required to justify his answers. Additionally, the conservation and transitive reasoning capacities incorporated in that model would be the more mature, later-developing versions of those sub-systems, not the versions which incorporate first principle rules.

A6.3 PROPOSED DEVELOPMENTAL PROCESS.

The character of the proposed developmental process will be apparent from the above discussion. Development is seen as proceeding through the assembly of lower-level sub-systems into higher-level sub-systems which, in turn, are assembled into still higher-level sub-systems. This process of vertical integration is also characterised by the progressive refinement of newly-assembled sub-systems until appropriate levels of efficiency have been achieved. This gain in efficiency would be achieved essentially by removal of redundant processing, with a concomitant reduction in solution path step length and process supervisor complexity.

In addition to this vertical integration, development would also be characterised by a horizontal spread of the range of objects and events to which specific sub-systems can be applied, and a progressive differentiation of those objects and events to which specific sub-systems cannot by applied.

Temporal discontinuities could be expected to be manifested in both the vertical and horizontal patterns of development. The discontinuities in the vertical pattern - such as those reported in the present study - would be attributed to the processing involved in, firstly, the assembly, and, secondly, the refinement of the constituent sub-systems.

The proposed developmental processes are not new in character. They reflect the same kind of approach that characterises Piaget's ideas regarding differentiation and integration of schemes. They also underlie Gagne's notions of cumulative learning, and are at the heart of Pascual-Leone's

view of learning. The other notable factor all these theoretical formulations have in common is the absence of a detailed description of the mechanisms required to give effect to them. However, the longer term enterprise of constructing a production system model of the development of linear measurement must proceed within a theoretical context that provides for such mechanisms. Since, at the present time, the most detailed account of these mechanisms is that provided by Klahr and Wallace (1976), their theoretical proposals would appear to offer the most appropriate framework within which to proceed.

A6.4 KLAHR AND WALLACE'S MODEL.

A6.4.1 ORGANISATION OF MEMORY.

The organisation of Klahr and Wallaces's (1976) memory model, and the flow of information within the model, are illustrated in Figure A6.1.

The following list of features serves to elaborate the schematic information provided by Figure A6.1:

(i) All productions are stored in long term memory (LTM).

(ii) The condition elements of productions are tested against the contents of up to two of the short term memory buffers, and/or "echoic" and/or "iconic memory stores".

(iii) The action elements of productions are applied to the contents of the short term memory buffers or specified parts of LTM, but not to echoic or iconic memory stores.

(iv) Different timing and capacity parameters apply to each memory component.

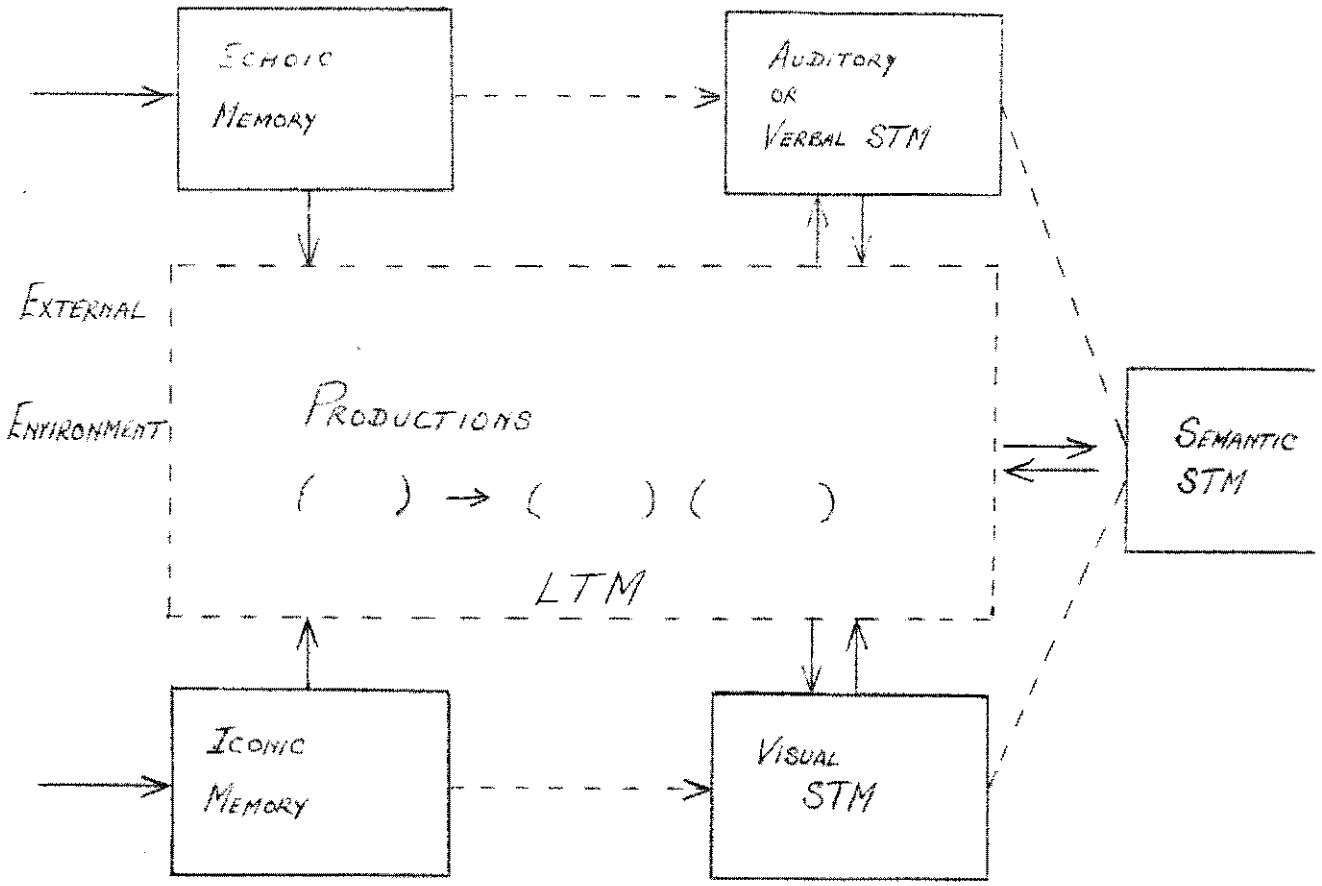


FIGURE A6.1: SCHEMATIC OUTLINE OF KLAHR AND WALLACE MEMORY MODEL. (FROM: KLAHR & WALLACE (1976) pp. 175).

NOTE: DASHED LINES REPRESENT FUNCTIONAL INFORMATION FLOW.

A6.4.2 REPRESENTATION IN LONG TERM MEMORY.

Klahr and WAllace (1976) represent LTM as a "disconnected graph structure". Each graph consists of a network of nodes. Each node consists of a production or production system; a description list connecting that node with other nodes; and an experience list that provides a trace of the involvement of that node in previous processing. The model also makes provision for an episodic component of LTM, called the "Time Line" (TL). Critical aspects of discrete processing sequences are stored in the Time Line. Essentially, they are the initiating conditions, the terminating condition, and production and production systems invoked during processing.

Klahr and Wallace (1976) propose that LTM is divided into "three tiers", with each tier having multiple levels. The tiers are searched in sequence and, within each tier, the levels are searched sequentially. However, within each level, parallel searching is carried out for the next appropriate production.

Productions and production systems relating to specific objects, events and situations previously encountered are stored at tier 1 of LTM. Productions controlling strategic procedures, such as "means-end analysis", and "predicate-led functioning", are stored at tier 2. Productions responsible for learning are held in tier 3.

A6.4.3 PROCESSING ECONOMY PRINCIPLE.

Klahr and Wallace (1976) see development largely in terms of rule generation and periodic LTM re-organisation. These processes operate in accordance with a "processing economy principle". That principle sets out the

conditions under which "specific situation sequences", encapsulated in time line episodic recording, become cast as "consistent sequence" productions and production systems. Concomitant with consistent sequence detection, the developing system also searches out, and eliminates, instances of redundancy.

These processes are not modelled by Klahr and Wallace in the form of executable computer programs. Rather, they posit the existence of certain innate systemic productions at each tier of LTM, but particularly at tier 3, which operate in accordance with the general systemic principles of consistency detection and redundancy elimination.

A6.5 APPLICATION OF KLAHR AND WALLACE'S MODEL.

The component of Klahr and Wallace's model that offers particular promise to the longer term project referred to earlier is their proposal concerning time line recording. Together with their proposed innate systemic productions, time line recording provides a mechanism that could give effect to developmental processes resulting in the kinds of delay and discontinuity observed in the present study.

The following example should be sufficient to convey a general understanding of the kind of processing envisaged. Consider the developing system that has already constructed general-rule models of conservation and transitive reasoning, for number and length. When faced with a problem of the kind represented by LR-M-CARD. the system would respond firstly by searching LTM for an appropriate common consistent sequence production system. This search would proceed under the control of a tier 2 innate

systemic production. After failing in the search, the initial goal might be replaced with a lesser goal, one concerning the length of one of the objects. This would initiate another search of LTM for a common consistent sequence appropriate to that new, lesser goal. If that search is also unsuccessful, the current goal would be replaced with one concerned with the properties of the length of an object. The next search of LTM might locate the production systems concerning the conservation of length and the transitive processing of length relations. (In this context, the properties of length could be construed as declarative or procedural knowledge. For present purposes, however, this distinction, and the consequences it implies for differing memorial representations, can be set aside). Though this processing would not initially result in a solution to the original problem, it does have the effect of constructing partial solutions. Moreover, the processing involved is recorded as a series of time line episodes. Further exposure to problems of the LR-M-CARD kind would result in additional time line recording. Subsequently examination of this time line information by the tier 3 innate systemic productions responsible for learning would identify these partial solutions, and recast them as consistent specific production systems that move progressively closer to eventual solutions.

Additionally, review of time line information in this manner would identify redundant processing of the kind represented in the first principle models discussed earlier in connection with conservation and linear measurement, and would lead to the eventual emergence of the more efficient, second phase models.

As Klahr and Wallace (1976) illustrate, using quantification processes as an example, this processing of experience, stored as time-sequenced information, by innate systemic productions can result in a pattern of development marked by temporal discontinuities. The main difficulty with the present proposal is that it could reasonably be expected that there would be behavioural evidence of the developing system assembling progressively closer approximations to the eventual solution strategy. That is, there should be behavioural evidence of faulty and incomplete execution of solution strategies. That was the case with the quantification evidence cited by Klahr and Wallace (1976). However, as far as the present study is concerned that was not the case - those subjects who could not solve LR-M-CARD did not even begin the task, while those who could solve the problem did so perfectly. On the surface this suggests a sudden insightful assembly of components into a solution strategy which is executed perfectly the first time it is applied. Intuitively, this seems less likely than the gradual - though marked by discontinuities - process sketched in the above paragraphs.

How then can the present study's lack of behavioural evidence of partial solutions be accounted for? Currently the most appealing suggestion is that, in the initial stages of the development of linear measurement, the components are still respresented in LTM by first principle systems. Because these first principle component systems would have longer step paths than the later-developing, general rule component systems, this would mean extended step paths at two (or more) levels. If that were the case, the strategy could impose supervisory processing demands beyond the developed capacity of the system. In that event, breakdown could occur before a point was reached where cognitive processes would be expressed behaviourally.

Clearly, these notions are somewhat imprecise. Though they carry some (usually dimly glimpsed) prospect of accounting for the developmental patterns observed in the present study, they will remain conjectural until they can be demonstrated in action. The most compelling demonstration would be observation of the behaviour of a production system model of the proposed system. A minimum outcome would be that examination of the trace of execution of a model of that kind would prompt the construction of other information-processing orientated hypotheses which would be both consistent with the behavioural data and amenable to being modelled as executable systems.

References.

Achenbach,T.M. & Weisz, J.R. A longitudinal study of developmental synchrony between conceptual identity, seriation, and transitivity of color, number, and length. Child Development, 1975, 46, 840-848.

Anderson, J.R. Arguments concerning representation for mental imagery. Psychological Review, 1978, 85, 249-277.

Anderson, J.R. Further arguments concerning representations for mental imagery: a response to Hayes-Roth and Pylyshyn. Psychological Review, 1979, 86(4), 395-406.

Ahr, P.R. & Youniss, J. Reasons for failure on the class inclusion problem. Child Development, 1970, 41, 131-143.

Anzai,Y. & Simon,H. The theory of learning by doing. Psychological Review, 1979, 86(2), 124-140.

Baylor, G.W. & Gascon, J. An information processing theory of aspects of the development of weight seriation in children. Cognitive Psychology, 1974, 6, 1-40.

Beard, R.M. The order of concept development-studies in two fields. Education Review, 1963, 15, 105-117.

Bearison, D.J. Role of measurement operations in the acquisition of conservation. Developmental Psychology, 1969, 1(6), 653-660.

Beilin, H. Stimulus and cognitive transformation in conservation. In D.Elkind and J.H. Flavell (Eds.), Studies in cognitive development. New York: Oxford University Press, 1969, 409-437.

Beth, E.W. & Piaget, J. Mathematical epistemology and psychology. Dordrecht-Holland: Reidel, 1966.

Braine, M.D.S. The ontogeny of certain logical operations: Piaget's formulation examined by non-verbal methods. Psychological Monographs, 1959, 73, (5,475).

Brainerd, C.J. Order of acquisition of transitivity, conservation and class inclusion of length and weight. Developmental Psychology, 1973, 8, 105-116.

Brainerd, C.J. Mathematical and behavioural foundations of number. Journal of General Psychology, 1973(a), 88, 221-281.

Brainerd, C.J. The origins of number concepts. Scientific American, 1973(b), 228(3), 101-109.

Brainerd, C.J. Inducing ordinal and cardinal representations of the first five natural numbers. Journal of Experimental Child Psychology, 1974, 18, 520-534.

Brainerd, C.J. Training and transfer of transitivity, conservation and class inclusion of length. Child Development, 1974, 45, 324-334.

Brainerd, C.J. Piaget's theory of intelligence. New Jersey: Prentice-Hall, 1978.

Brainerd, C.J. & Fraser, M. A further test of the ordinal theory of number development. Journal of Genetic Psychology, 1975, 127, 21-33.

Brainerd, C.J. & Kaszor, P. An analysis of two proposed sources of children's class inclusion errors. Developmental Psychology, 1974, 10, 633-643.

Brainerd, C.J. & Vanden Heuvel, K. Development of geometric imagery in five to eight year olds. Genetic Psychology Monographs, 1974, 98, 89-143.

Bryant, P.E. Perception and understanding in young children. London: Methuen, 1974.

Bryant, P.E. & Trabasso, T. Transitive inference and memory in young children. Nature, 1971, 232, 456-458.

Case, R. Learning and development: A neo-Piagetian interpretation. Human Development, 1972, 15, 339-358.

Case, R. Validation of a neo-Piagetian mental capacity construct. Journal of Experimental Child Psychology, 1972(a), 14, 287-302.

Dale, L.G. A neo-Piagetian investigation of some factors affecting success on Piaget's flexibility of rods problem. Unpublished doctoral dissertation, La Trobe University, 1975.

DeBoysson-Bardies, B. & O'Regan, K. What children do in spite of adults' hypotheses. Nature, 1973, 246, 531-534.

DeSoto, C.B., London, M., & Handel, S. Social reasoning and spatial paralogic. Journal of Personality and Social Psychology, 1965, (2,4), 513-521.

Dodwell, P.C. Children's understanding of number and related concepts. Canadian Journal of Psychology, 1960, 14, 191-205.

Dodwell, P.C. Children's understanding of number concepts: characteristics of an individual and of a group test. Canadian Jounral of Psychology, 1961, 15, 29-36.

Edwards, A.L. Techniques of attitude scale construction. New York: Appleton-Century-Crofts, 1957.

Elkind, D. Discrimination, seriation and numeration of size and dimensional differences in young children. Piaget replication study vi. Journal of Genetic Psychology, 1964, 104, 275-296.

Flavell, J.H. The developmental psychology of Jean Piaget. New York: Van Nostrand, 1963.

Flavell, J.H. Stage-related properties of cognitive development. Cognitive Psychology, 1971, 2, 421-453.

Flavell, J.H. An analysis of cognitive developmental sequences.

Genetic Psychology Monograph, 1972, 86, 279-350.

Gagne, R.M. Contributions of learning to human learning. Psychological Review, 1968, 75, 177-191.

Gelman, R. Logical capacity of very young children. Child Development, 1972, 43, 75-90.

Gelman, R. & Gallistel, C.R. The child's understanding of number. Cambridge, Mass: Harvard University Press, 1978.

Gelman, R. & Tucker, M.F. Further investigations of the young child's conception of number. Child Development, 1975, 46, 167-175.

Goldsmidt, M.L. Different types of conservation and non-conservation and their relation to age, sex, IQ, MA and vocabulary. Child Development, 1967, 38(2), 1229-1246.

Green, B.F. Attitude measurement. In: G.Lindzey (Ed.), Handbook of Social Psychology. Mass: Addison-Wesley, 1954.

Green, B.F. A method of scalogram analysis using summary statistics. Psychometrika, 1959, 21(1), 78-99.

Grieve, R. & Nesdale, A.R. Observations on a test of transitive inference in children. Australian Journal of Psychology, 1979, 31(1), 43-48.

Groen, G. & Parkman, J. A chromatic analysis of simple addition. Psychological Review, 1972, 79, 329-343.

Gruen, G.E. Experiences affecting the development of number conservation in children. Child Development, 1965, 36(4), 963-979.

Halford, G.S. & Galloway, W. Children who fail to make transitive inferences can remember comparisons. Australian journal of Psychology, 1977, 29, 1-5.

Halford, G.S. Measurement and memory in transitivity: a reply to Grieve and Nesdale. Australian Journal of Psychology, 1979, 31(1), 49-56.

Hays, W.L. Statistics. London: Holt, Rinehart, Winston, 1969.

Hood, H.B. An experimental study of Piaget's theory of the development of number in children. British Journal of Psychology, 1962, 53, 273-286.

Hunt, E.B. & Poltrock, S.E. The mechanics of thought. In. B.H. Kantowitz (ed.), Human Information Processing: Tutorials in Performance and Cognition. New Jersey: Lawrence Erlbaum Associates, 1974.

Inhelder, B. Some aspects of Piaget's genetic approach to cognition. Monograph of the Society for Research in Child Development, 1962, 27(2), 235-247.

Inhelder, B. & Piaget, J. The early growth of logic in the child. London: Routledge and Kegan Paul, 1964.

Jennings, J.R. The effect of verbal and pictorial presentation on class inclusion competence and performance. Psychonomic Science, 1970, 20, 357-358.

Klahr, D. & Wallace, J.G. Class inclusion processes. In. S. Farnham-Diggory (Ed.), Information Processing in Children. New York: Academic Press, 1972.

Klahr, D. & Wallace, J.G. Cognitive development: An information-processing view. New Jersey: Lawrence Erlbaum, 1976.

Kofsky, E. A scalogram study of classificatory development. Child Development, 1966, 37, 191-204.

Langford, P. Beyond Piaget: Recent theories of concept development and their significance for teaching. Melbourne: LaTrobe University, 1978.

Langford, P. A longitudinal study of children's understanding of logical laws in arithmetic and Boolean algebra. Educational Psychology 1981, 1(2), 119-139.

Lifschitz, M. & Langford, P. The role of counting and measurement in conservation learning. Archives de Psychologie, 1977, 173, 1-14.

Loevinger, J. A systematic approach to the construction and evaluation of tests of ability. Psychological Monographs. 1946-47, 60-61, whole No285(4).

Lovell, K. & Ogilvie, E. A study of the conservation of weight in the junior school. British Journal of Educational Psychology, 1961, 31, 138-144.

Mc.Mannis, D.L. Conservation and transitivity of weight and length by normals and retardates. Developmental Psychology, 1969, 1, 373-382.

Macnamara, J. A Note on Piaget and Number. Child Development, 1975, 46, 424-429.

Miller, S.A. Non verbal assessment of Piagetian concepts. Psychological Bulletin, 1976, 83, 405-430.

Neisser, U. Cognitive Psychology. New York: Appleton-Century-Crofts, 1967.

Newell, A. & Simon, H.A. Human problem solving. New Jersey: Prentice-Hall, 1972.

Ohlsson, S. The PSS program: User's manual and program descriptions. Stencil, Department of Psychology, University of Stockholm, 1980.

O'Shea, T. & Young, R. A production rule account of errors in children's subtraction. Working Paper No.42, Department of Artificial Intelligence, University of Edinburgh. 1978.

Osherson, D.M. Logical abilities in children. Potomac, Erlbaum, Vol.1. 1974.

Pascual-Leone, J. A mathematical model for the transition rule in Piaget's developmental stages. Acta Psychologica, 1970, 32, 301-345.

Pascual-Leone, J. Compounds, confounds, and models in developmental information processing: A reply to Trabasso and Foellinger. Journal of Experimental Child Psychology, 1978, 26, 18-40.

Pascual-Leone, J. & Smith, J. The encoding and decoding of symbols by children: A new experimental paradigm and a neo-Piagetian model. Journal of Experimental Child Psychology, 1969, 8, 328-355.

Piaget, J. The child's conception of number. New York: Humanities, 1952.

Piaget, J. Logic and psychology. Manchester: Manchester University Press, 1953.

Piaget, J. The construction of reality in the child. New York: Basic Books, 1954.

Piaget, J. The stages of intellectual development of the child. In: P.C. Wason, and P.N. Johnson-Laird (Eds.), Thinking and Reasoning. London: Penguin, 1968.

Piaget, J. The principles of genetic epistemology. London: Routledge and Kegan Paul, 1972.

Piaget, J. & Inhelder, B. The psychology of the child. New York: Basic Books, 1969.

Piaget, J. Inhelder, B. & Szeminska, A. The child's conception of geometry. London: Routledge and Kegan Paul, 1960.

Riley, C.A. & Trabasso, T. Comparatives, logical structure and encoding in a transitive inference task. Journal of Experimental Psychology, 1974, 17, 187-203.

Roodin, M.L. & Gruen, G.E. The role of memory in making transitive judgements. Journal of Experimental Child Psychology, 1970, 10, 264-275.

Schaeffer, B. An unbalanced view of numbers. Contemporary Psychology, 1980, 25(7), 556-557.

Schiff, W. & Saarni, C.I. Perception and conservation of length: Piaget and Taponeir revisited. Developmental Psychology, 1976, 12(2), 98-106.

Siegel, L.S. The sequence of development of certain number concepts in pre-school children. Developmental Psychology, 1971(a), 5, 357-361.

Siegel, L.S. The development of certain number concepts. Developmental Psychology, 1971(b), 5, 362-363.

Siegel, L.S. Development of number concepts: ordering and correspondence operations and the role of length cues. Developmental Psychology, 1974, 10(b), 907-912.

Smedslund, J. The acquisition of conservation of substance and weight in children. II. External re-inforcement of conservation of weight and of the operations of addition and subtraction. Scandanavian Journal of Psychology, 1961, 71-84.

Smedslund, J. Development of concrete transitivity of length in children. Child Development, 1963, 34, 389-405.

Strauss, S. & Ilan, J. Length conservation and speed concepts: organizational disequilibrium training between concepts. Journal of Educational Psychology, 1975, 67(4), 470-477.

Suppes, P. & Zinnes, J.L. Basic measurement theory. In. R.D.Luce, R.R.Bush, & E.Galanter (Eds.), Handbook of Mathematical Psychology. New York: Wiley, 1963.

Trabasso, T. On the evaluation of parameters and the evaluation of a mathematical model: A reply to Pascual-Leone. Journal of Experimental Child Psychology, 1978, 26, 41-45.

Trabasso, T. & Follinger, D.B. Information processing capacity in children: A test of Pascual-Leone's model. Journal of Experimental Child Psychology, 1978, 26, 1-17.

Wallace, J.G. Stages and transition in conceptual development: An experimental study. Slough, National Foundation for Educational Research in England and Wales, 1972.

Wallach, L. On the bases of conservation. In. D.Eklind and J.H. Flavell (Eds.), Studies in Cognitive Development. New York: Oxford University Press, 1969, 191-219.

Wallach, L. Wall, A.J. & Anderson, L. Number conservation: The roles of reversibility, addition-subtraction and misleading perceptual cues. Child Development, 1967, 38, 425-442.

Winer, G.A. An analysis of verbal facilitation of class inclusion reasoning. Child Development, 1974, 45, 224-227.

Winer, G.A. & Kronberg, D.D. Children's responses to verbally and pictorally presented class inclusion items and to a task of number conservation. Journal Genetic Psychology, 1974, 125, 141-152.

Winston, P.H. Artificial Intelligence. Mass: Addison-Wesley, 1979.

Witkin, H.A. The perception of the upright. Scientific American, 1959, 200(2), 50-56.

Wohlwill, J.F. A study of the development of the number concept by scalogram analysis. The Journal of Genetic Psychology, 1960, 97, 345-377.

Wohlwill, J.F. Responses to class inclusion questions with verbally and pictorially presented items. Child Development, 1968, 39, 449-465.

Wohlwill, J.F. & Lowe, R.C. Experimental analysis of the development of the conservation of number. Child Development, 1962, 33, 153-167.

Youniss, J. & Dennison, A. Figurative and operative aspects of children's inference. Child Development, 1971, 42, 1837-1847.

Youniss, J. & Murray, J.P. Transitive inference with non-transitive solutions controlled. Developmental Psychology, 1970, 2, 169-175.

DEVELOPMENT OF LINEAR MEASUREMENT.

LISTINGS AND EXECUTION TRACES OF SIX PRODUCTION-SYSTEM MODELS OF THREE NUMBER TASKS.

ADDENDA TO THE MAIN REPORT.

Kenneth Vine

January, 1983.

CONTENTS.

ADDENDUM 1.

LISTING AND EXECUTION TRACE FOR

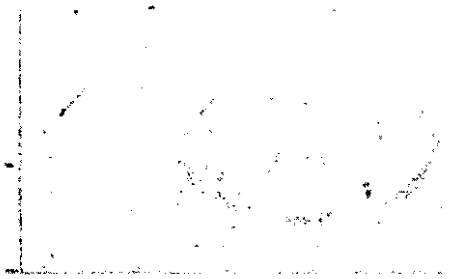
N-ADD-NV: TABLE-LOOK-UP METHOD.

```
 p 400
(VARIABLES;
(A GENC)
(B GENC)
(C GENC)
(D GENC)
(E GENC)
(<X> SEQC)
(<P> SEQC)
)
(PRODUCTIONS;
(P000 (GOAL * ATTEND)(OLD SEND C)
==>
REPL((GOAL * ATTEND)  ;  (GOAL + ATTEND));
DEL((OLD SEND C));
USER()   )
(P00  (GOAL * ATTEND)(GO)
==>
REPL((GOAL * ATTEND)  ;  (GOAL + ATTEND));
DEL((GO))  )
(P0 (GOAL * ATTEND)
==>
USER()  )
(P1 (GOAL + ATTEND)(TOP A)(BOTTOM B)(SEND C)
==>
REPL((GOAL + ATTEND)  ;  (GOAL *S* ATTEND));
INS((GOAL * SEND C));
INS((Y 1 Z))   )
(P2 (GOAL * SEND C)
==>
REPL((GOAL * SEND C)  ;  (GOAL *S* SEND C));
INS((GOAL * PUSH BUTTON))  )
(P3 (GOAL * PUSH BUTTON)
==>
DEL((GOAL * PUSH BUTTON));
INS((ELM A));
INS((GOAL * SUBIT))  )
(P4  (GOAL * SUBIT)(ELM A)
==>
DEL((GOAL * SUBIT));
INS((OS 1));
DEL((ELM A));
INS((GOAL * COUNT))  )
(P5  (GOAL * COUNT)(OS 1)(Y <X> Z)
==>
REPL((GOAL * COUNT)  ;  (GOAL + COUNT));
DEL((OS 1));
SAY(<X>);
REPL((Y <X> Z)  ;  (SAID <X>));
INS((GOAL * MARK))  )
(P6 (GOAL * MARK)(SAID 1)
==>
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y 2 Z))   )
(P6A  (GOAL * MARK)(SAID 2)
==>
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y 3 Z))   )
(P6B  (GOAL * MARK)(SAID 3)
==>
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y 4 Z))   )
(P6C  (GOAL * MARK)(SAID 4)
==>
REPL((GOAL * MARK)  ;  (GOAL + MARK))
```

```
INS((Y 5 Z))    )
(P7 (GOAL * MARK)(GOAL * COUNT)(GOAL *S* SEND C)
(SAID C)
==>
DEL((GOAL *S* SEND C))$
DEL((GOAL * MARK))$
DEL((GOAL * COUNT))$
DEL((SAID C))$
REPL((GOAL *S* ATTEND)  $  (GOAL * ATTEND))  )
(P8  (GOAL * MARK)(GOAL * COUNT)(GOAL *S* SEND C)(SAID D)
==>
REPL((GOAL *S* SEND C)  $  (GOAL * SEND C))$
DEL((GOAL * MARK))$
DEL((GOAL * COUNT))$
DEL((SAID D))    )
(P9 (GOAL * ATTEND)(MANY BOTTOM BEFORE)
==>
REPL((GOAL * ATTEND)  $  (GOAL *S* ATTEND))$
DEL((MANY BOTTOM BEFORE))$
INS((GOAL * RECALL BOTTOM))   )
(P10  (GOAL * RECALL BOTTOM)(BOTTOM B)
==>
SAY((BOTTOM D))$
DEL((GOAL * RECALL BOTTOM))$
REPL((GOAL *S* ATTEND)  $  (GOAL * ATTEND))  )
(P11 (GOAL * ATTEND)(MANY TUBE)
==>
REPL((GOAL * ATTEND)  $  (GOAL *S* ATTEND))$
DEL((MANY TUBE))$
INS((GOAL * RECALL  TUBE))   )
(P12  (GOAL * RECALL TUBE)(SEND C)
==>
DEL((GOAL * RECALL TUBE))$
REPL((GOAL *S* ATTEND)  $  (GOAL * ATTEND))$
SAY((SEND C))    )
(P13 (GOAL * ATTEND)(MANY BOTTOM NOW)
==>
REPL((GOAL * ATTEND)  $  (GOAL *S* ATTEND))$
DEL((MANY BOTTOM NOW))$
INS((GOAL * ADD))     )
(P14 (GOAL * ADD)(BOTTOM B)(ABS (GOAL * LIST B))
==>
REPL((GOAL * ADD)  $  (GOAL *S* ADD))$
INS((GOAL * LIST B))  )
(P15 (GOAL * LIST ZERO)
==>
REPL((GOAL *S* ADD)  $  (GOAL * ADD))$
INS((2 TWO Z))$
INS((3 THREE Z))   )
(P15A  (GOAL * LIST TWO)
==>
REPL((GOAL *S* ADD)  $  (GOAL * ADD))$
INS((3 FIVE Z))$
INS((4 SIX Z))   )
(P15B (GOAL * LIST SIX)
==>
REPL((GOAL *S* ADD)  $  (GOAL * ADD))$
INS((2 EIGHT Z))$
INS((3 NINE Z))   )
(P15C  (GOAL * LIST NINE)
==>
REPL((GOAL *S* ADD)  $  (GOAL * ADD))$
INS((0 NINE Z))$
INS((1 TEN Z))    )
(P16   (GOAL * ADD)(SEND C)(C <X> Z)(BOTTOM B)
==>
```

```
SAY(<X>);
DEL((GOAL * NEED));
DEL((GOAL * LIST B));
INS((GOAL X PURGE))    )
(P24A (GOAL * PURGE)(SEND C)(C <X> Z)(BOTTOM B)
==>
REPL((BOTTOM B)   ;   (BOTTOM <X>));
REPL((SEND C)   ;   (OLD SEND C));
DEL((C <X> Z))      )
(P24 (GOAL * PURGE)(Y <X> Z)
==>
DEL((Y <X> Z))    )
(P25   (GOAL * PURGE)(C <P> Z)
==>
DEL((C <P> Z))    )
(P26 (GOAL * PURGE)(BOTTOM 12)
==>
STOP)
(P27 (GOAL * PURGE)
==>
DEL((GOAL * PURGE));
REPL((GOAL *?? ATTEND)   ;   (GOAL * ATTEND))    )
)
(PDLISTS:
(M-ADD-NU1
P9 P11 P13
P000 P0C P0
P1 P2 P3 P4 P5 P6
P6A P6B P6C
P7 P8 P10 P12
P13 P14 P16 P15
P15A P15B P15C
P24A
P24 P25 P26 P27
)
)
(DEF: SYM/(GOAL * ATTEND)(TOP TWELVE)(BOTTOM ZERO)(SEND 2)))
%%%END
%
```

```
RUN BEGINS :

((GOAL * ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK J
UNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(GO)


STM: ((GO) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK)
P00


STM: ((GOAL ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUN
K JUNK JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y 1 Z) (GOAL * SEND 2) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 2) (Y 1 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 2) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (QS 1) (GOAL *S* SEND 2) (Y 1 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P5

*******  1


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 2) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK
)
P6


STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 2) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUN
K)
P0
```



```
STM: ((GOAL * SEND 2) (Y 2 Z) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 2) (Y 2 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 2) (Y 2 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (QS 1) (GOAL *S* SEND 2) (Y 2 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P5

*******  2


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 2) (GOAL *S* SEND 2) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK
)
P6A


STM: ((Y 3 Z) (GOAL MARK) (SAID 2) (GOAL COUNT) (GOAL *S* SEND 2) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUN
K)
P7


STM: ((Y 3 Z) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9
INPUT TO STM:
*(MANY BOTTOM BEFORE)


STM: ((MANY BOTTOM BEFORE) (GOAL * ATTEND) (Y 3 Z) (TOP TWELVE) (BOTT
OM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9


STM: ((GOAL * RECALL BOTTOM) (GOAL *S* ATTEND) (Y 3 Z) (TOP TWELVE) (
BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

*********  (BOTTOM ZERO)


STM: ((BOTTOM ZERO) (GOAL * ATTEND) (Y 3 Z) (TOP TWELVE) (SEND 2) JUN
```

```
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)


STM: ((MANY TUBE) (GOAL * ATTEND) (BOTTOM ZERO) (Y 3 Z) (TOP TWELVE)
(SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (BOTTOM ZERO) (Y 3 Z) (T
OP TWELVE) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

*******  (SEND 2)


STM: ((SEND 2) (GOAL * ATTEND) (BOTTOM ZERO) (Y 3 Z) (TOP TWELVE) JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM NOW)


STM: ((MANY BOTTOM NOW) (GOAL * ATTEND) (SEND 2) (BOTTOM ZERO) (Y 3 Z
) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((GOAL * ADD) (GOAL *S* ATTEND) (SEND 2) (BOTTOM ZERO) (Y 3 Z) (
TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P14


STM: ((GOAL * LIST ZERO) (GOAL *S* ADD) (BOTTOM ZERO) (GOAL *S* ATTEN
D) (SEND 2) (Y 3 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((3 THREE Z) (2 TWO Z) (GOAL * LIST ZERO) (GOAL * ADD) (BOTTOM Z
ERO) (GOAL *S* ATTEND) (SEND 2) (Y 3 Z) (TOP TWELVE) JUNK JUNK JUNK J
UNK)
P16

*******  TWO


STM: ((GOAL * PURGE) (SEND 2) (2 TWO Z) (BOTTOM ZERO) (3 THREE Z) (GO
AL *S* ATTEND) (Y 3 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P23A


STM: ((GOAL * PURGE) (OLD SEND 2) (BOTTOM TWO) (3 THREE Z) (GOAL *S*
ATTEND) (Y 3 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
```

```
P24


STM: ((GOAL * PURGE) (OLD SEND 2) (BOTTOM TWO) (3 THREE Z) (GOAL *S*
ATTEND) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (OLD SEND 2) (BOTTOM TWO) (GOAL *S* ATTEND) (TOP
 TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((OLD SEND 2) (BOTTOM TWO) (GOAL * ATTEND) (TOP TWELVE) JUNK JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*((SEND FOUR)\)RUOF DNES(\
(SEND 4)


STM: ((SEND 4) (GOAL ATTEND) (BOTTOM TWO) (TOP TWELVE) JUNK JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y 1 Z) (GOAL * SEND 4) (GOAL *S* ATTEND) (TOP TUCLVE) (BOTTOM
TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 1 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELN A) (GOAL *S* SEND 4) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 4) (Y 1 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

*1*****  1


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK
```

```
)
P8


STM: ((GOAL * SEND 4) (Y 2 Z) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 2 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTON TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y 2 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTON TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 4) (Y 2 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTON TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

******* 2


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 2) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTON TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P6A


STM: ((Y 3 Z) (GOAL MARK) (SAID 2) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTON TWO) (SEND 4) JUNK JUNK JUNK JUNK
)
P3


STM: ((GOAL * SEND 4) (Y 3 Z) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTON
TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 3 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTON TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y 3 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTON TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 4) (Y 3 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTON TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5
```

```
*******  3


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 3) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P68


STM: ((Y 4 Z) (GOAL MARK) (SAID 3) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK
)
P9


STM: ((GOAL * SEND 4) (Y 4 Z) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 4 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUMIT) (ELM A) (GOAL *S* SEND 4) (Y 4 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 4) (Y 4 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

*******  4


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 4) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P10


STM: ((Y 5 Z) (GOAL MARK) (SAID 4) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK
)
P2


STM: ((Y 5 Z) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK
 JUNK JUNK JUNK [illegible] JUNK JUNK)
P6
INPUT TO STM:
*(ANY BOTTOM SENSE)


STM: [illegible]
```

```
OM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9


STM: ((GOAL * RECALL BOTTOM) (GOAL *S* ATTEND) (Y 5 Z) (TOP TWELVE) (
BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

*******  (BOTTOM TWO)


STM: ((BOTTOM TWO) (GOAL * ATTEND) (Y 5 Z) (TOP TWELVE) (SEND 4) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)



STM: ((MANY TUBE) (GOAL * ATTEND) (BOTTOM TWO) (Y 5 Z) (TOP TWELVE) (
SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (BOTTOM TWO) (Y 5 Z) (TO
P TWELVE) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

*******  (SEND 4)


STM: ((SEND 4) (GOAL * ATTEND) (BOTTOM TWO) (Y 5 Z) (TOP TWELVE) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM NOW)



STM: ((MANY BOTTOM NOW) (GOAL * ATTEND) (SEND 4) (BOTTOM TWO) (Y 5 Z)
 (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((GOAL * ADD) (GOAL *S* ATTEND) (SEND 4) (BOTTOM TWO) (Y 5 Z) (T
OP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P14


STM: ((GOAL * LIST TWO) (GOAL *S* ADD) (BOTTOM TWO) (GOAL *S* ATTEND)
 (SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK)
P15A


STM: ((4 SIX Z) (3 FIVE Z) (GOAL * LIST TWO) (GOAL * ADD) (BOTTOM TWO
) (GOAL *S* ATTEND) (SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK
)
```

```
P16

********  SIX


STM: ((GOAL * PURGE) (SEND 4) (4 SIX Z) (BOTTOM TWO) (3 FIVE Z) (GOAL
 *S* ATTEND) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P24A


STM: ((GOAL * PURGE) (OLD SEND 4) (BOTTOM SIX) (3 FIVE Z) (GOAL *S* A
TTEND) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (OLD SEND 4) (BOTTOM SIX) (3 FIVE Z) (GOAL *S* A
TTEND) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (OLD SEND 4) (BOTTOM SIX) (GOAL *S* ATTEND) (TOP
 TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((OLD SEND 4) (BOTTOM SIX) (GOAL * ATTEND) (TOP TWELVE) JUNK JUN
K JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*(SEND 3)


STM: ((SEND 3) (GOAL ATTEND) (BOTTOM SIX) (TOP TWELVE) JUNK JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y 1 Z) (GOAL * SEND 3) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 3) (Y 1 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 3) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (US 1) (GOAL *S* SEND 3) (Y 1 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P5
```

```
*******  1


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 3) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 3) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK
)
P3


STM: ((GOAL * SEND 3) (Y 2 Z) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 3) (Y 2 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 3) (Y 2 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (DS 1) (GOAL *S* SEND 3) (Y 2 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P5

*******  2


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 2) (GOAL *S* SEND 3) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P6A


STM: ((Y 3 Z) (GOAL MARK) (SAID 2) (GOAL COUNT) (GOAL *S* SEND 3) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK
)
P8


STM: ((GOAL * SEND 3) (Y 3 Z) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 3) (Y 3 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK)
P3
```

```
STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 3) (Y 3 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (QS 1) (GOAL *S* SEND 3) (Y 3 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P5

*******  3


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 3) (GOAL *S* SEND 3) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P6B


STM: ((Y 4 Z) (GOAL MARK) (SAID 3) (GOAL COUNT) (GOAL *S* SEND 3) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK
)
P7


STM: ((Y 4 Z) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM BEFORE)


STM: ((MANY BOTTOM BEFORE) (GOAL * ATTEND) (Y 4 Z) (TOP TWELVE) (BOTT
OM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9


STM: ((GOAL * RECALL BOTTOM) (GOAL *S* ATTEND) (Y 4 Z) (TOP TWELVE) (
BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

*******  (BOTTOM SIX)


STM: ((BOTTOM SIX) (GOAL * ATTEND) (Y 4 Z) (TOP TWELVE) (SEND 3) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)


STM: ((MANY TUBE) (GOAL * ATTEND) (BOTTOM SIX) (Y 4 Z) (TOP TWELVE) (
SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11
```

```
STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (BOTTOM SIX) (Y 4 Z) (TO
P TWELVE) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

*******  (SEND 3)


STM: ((SEND 3) (GOAL * ATTEND) (BOTTOM SIX) (Y 4 Z) (TOP TWELVE) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM NOW)



STM: ((MANY BOTTOM NOW) (GOAL * ATTEND) (SEND 3) (BOTTOM SIX) (Y 4 Z)
 (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((GOAL * ADD) (GOAL *S* ATTEND) (SEND 3) (BOTTOM SIX) (Y 4 Z) (T
OP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P14


STM: ((GOAL * LIST SIX) (GOAL *S* ADD) (BOTTOM SIX) (GOAL *S* ATTEND)
 (SEND 3) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK)
P15B


STM: ((3 NINE Z) (2 EIGHT Z) (GOAL * LIST SIX) (GOAL * ADD) (BOTTOM S
IX) (GOAL *S* ATTEND) (SEND 3) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK JU
NK)
P16

*******  NINE


STM: ((GOAL * PURGE) (SEND 3) (3 NINE Z) (BOTTOM SIX) (2 EIGHT Z) (GO
AL *S* ATTEND) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P24A


STM: ((GOAL * PURGE) (OLD SEND 3) (BOTTOM NINE) (2 EIGHT Z) (GOAL *S*
 ATTEND) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (OLD SEND 3) (BOTTOM NINE) (2 EIGHT Z) (GOAL *S*
 ATTEND) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (OLD SEND 3) (BOTTOM NINE) (GOAL *S* ATTEND) (TO
P TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
```

```
P27


STM: ((OLD SEND 3) (BOTTOM NINE) (GOAL * ATTEND) (TOP TWELVE) JUNK JU
NK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*(SEND 1)



STM: ((SEND 1) (GOAL ATTEND) (BOTTOM NINE) (TOP TWELVE) JUNK JUNK JUN
K JUNK JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y 1 Z) (GOAL * SEND 1) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 1) (Y 1 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 1) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 1) (Y 1 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK)
P5

******** 1


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 1) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK
)
P6


STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 1) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM NINE) (SEND 1) JUNK JUNK JUNK JUN
K)
P7


STM: ((Y 2 Z) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM NINE) (SEND 1) JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P8
INPUT TO STM:
*(MARY BUTTON BEFORE)
```

```
STM: ((MANY BOTTOM BEFORE) (GOAL * ATTEND) (Y 2 Z) (TOP TWELVE) (BOTT
OM NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9


STM: ((GOAL * RECALL BOTTOM) (GOAL *S* ATTEND) (Y 2 Z) (TOP TWELVE) (
BOTTOM NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

********  (BOTTOM NINE)


STM: ((BOTTOM NINE) (GOAL * ATTEND) (Y 2 Z) (TOP TWELVE) (SEND 1) JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)



STM: ((MANY TUBE) (GOAL * ATTEND) (BOTTOM NINE) (Y 2 Z) (TOP TWELVE)
(SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (BOTTOM NINE) (Y 2 Z) (T
OP TWELVE) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

********  (SEND 1)


STM: ((SEND 1) (GOAL * ATTEND) (BOTTOM NINE) (Y 2 Z) (TOP TWELVE) JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM NOW)



STM: ((MANY BOTTOM NOW) (GOAL * ATTEND) (SEND 1) (BOTTOM NINE) (Y 2 Z
) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((GOAL * ADD) (GOAL *S* ATTEND) (SEND 1) (BOTTOM NINE) (Y 2 Z) (
TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P14


STM: ((GOAL * LIST NINE) (GOAL *S* ADD) (BOTTOM NINE) (GOAL *S* ATTEN
D) (SEND 1) (Y 2 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK)
P15C
```

```
STM: ((1 TEN Z) (0 NINE Z) (GOAL * LIST NINE) (GOAL * ADD) (BOTTOM NI
NE) (GOAL *S* ATTEND) (SEND 1) (Y 2 Z) (TOP TWELVE) JUNK JUNK JUNK JU
NK)
P16

*******  TEN


STM: ((GOAL * PURGE) (SEND 1) (1 TEN Z) (BOTTOM NINE) (0 NINE Z) (GOA
L *S* ATTEND) (Y 2 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P24A


STM: ((GOAL * PURGE) (OLD SEND 1) (BOTTOM TEN) (0 NINE Z) (GOAL *S* A
TTEND) (Y 2 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (OLD SEND 1) (BOTTOM TEN) (0 NINE Z) (GOAL *S* A
TTEND) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (OLD SEND 1) (BOTTOM TEN) (GOAL *S* ATTEND) (TOP
 TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((OLD SEND 1) (BOTTOM TEN) (GOAL * ATTEND) (TOP TWELVE) JUNK JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*
*
*()


STM: (NIL (GOAL ATTEND) (BOTTOM TEN) (TOP TWELVE) JUNK JUNK JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK)


END OF RUN

NUMBER OF PRODUCTIONS FIRED : 202

NUMBER OF OPERATIONS EXECUTED : 557

NUMBER OF PRODUCTIONS TRIED : 3310

NUMBER OF SINGLE-ELEMENT COMPARISONS : 52057

SIMULATED TIME : 0

COMPUTER TIME: 275731 MSECS

NUMBER OF CONS-OPERATIONS : 516AA1
```

ADDENDUM 2.

LISTING AND EXECUTION TRACE FOR N-ADD-NV: COUNTING-BASED METHOD.

```
 P 500
(VARIABLES:
(A GENC)
(B GENC)
(C GENC)
(D GENC)
(E GENC)
(<X> SEQC)
(<P> SEQC)
)
(PRODUCTIONS:
(P000 (GOAL * ATTEND)(OLD SEND C)
-->
REPL((GOAL * ATTEND)  ;  (GOAL + ATTEND));
DEL((OLD SEND C));
USER()    )
(P00  (GOAL * ATTEND)(GO)
-->
REPL((GOAL * ATTEND)  ;  (GOAL + ATTEND));
DEL((GO))  )
(P0 (GOAL * ATTEND)
-->
USER()  )
(P1 (GOAL + ATTEND)(TOP A)(BOTTOM B)(SEND C)
-->
REPL((GOAL + ATTEND)  ;  (GOAL *S* ATTEND));
INS((GOAL * SEND C));
INS((Y 1 Z))    )
(P2 (GOAL * SEND C)
-->
REPL((GOAL * SEND C)  ;  (GOAL *S* SEND C));
INS((GOAL * PUSH BUTTON))  )
(P3 (GOAL * PUSH BUTTON)
-->
DEL((GOAL * PUSH BUTTON));
INS((ELM A));
INS((GOAL * SUBIT))  )
(P4  (GOAL * SUBIT)(ELM A)
-->
DEL((GOAL * SUBIT));
INS((QS 1));
DEL((ELM A));
INS((GOAL * COUNT))  )
(P5  (GOAL * COUNT)(QS 1)(Y <X> Z)
-->
REPL((GOAL * COUNT)  ;  (GOAL + COUNT));
DEL((QS 1));
SAY(<X>);
REPL((Y <X> Z)  ;  (SAID <X>));
INS((GOAL * MARK))  )
(P6 (GOAL * MARK)(SAID 1)
-->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y 2 Z))    )
(P6A  (GOAL * MARK)(SAID 2)
-->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y 3 Z))    )
(P6B  (GOAL * MARK)(SAID 3)
-->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y 4 Z))    )
(P6C  (GOAL * MARK)(SAID 4)
-->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
```

```
INS((Y 5 Z))   )
(P7 (GOAL + MARK)(GOAL + COUNT)(GOAL *S* SEND C)
(SAID C)
--->
DEL((GOAL *S* SEND C))$
DEL((GOAL + MARK))$
DEL((GOAL + COUNT))$
DEL((SAID C))$
REPL((GOAL *S* ATTEND)  $  (GOAL * ATTEND))  )
(P8  (GOAL + MARK)(GOAL + COUNT)(GOAL *S* SEND C)(SAID D)
--->
REPL((GOAL *S* SEND C)  $  (GOAL * SEND C))$
DEL((GOAL + MARK))$
DEL((GOAL + COUNT))$
DEL((SAID D))    )
(P9 (GOAL * ATTEND)(MANY BOTTOM BEFORE)
--->
REPL((GOAL * ATTEND)  $  (GOAL *S* ATTEND))$
DEL((MANY BOTTOM BEFORE))$
INS((GOAL * RECALL BOTTOM))   )
(P10  (GOAL * RECALL BOTTOM)(BOTTOM B)
--->
SAY((BOTTOM B))$
DEL((GOAL * RECALL BOTTOM))$
REPL((GOAL *S* ATTEND)  $  (GOAL * ATTEND))  )
(P11 (GOAL * ATTEND)(MANY TUBE)
--->
REPL((GOAL * ATTEND)  $  (GOAL *S* ATTEND))$
DEL((MANY TUBE))$
INS((GOAL * RECALL  TUBE))   )
(P12  (GOAL * RECALL TUBE)(SEND C)
--->
DEL((GOAL * RECALL TUBE))$
REPL((GOAL *S* ATTEND)  $  (GOAL * ATTEND))$
SAY((SEND C))    )
(P13 (GOAL * ATTEND)(MANY BOTTOM NOW)(BOTTOM B)
--->
REPL((GOAL * ATTEND)  $  (GOAL *S* ATTEND))$
DEL((MANY BOTTOM NOW))$
INS((GOAL * ADD))$
INS((W 0 V))$
INS((Y P Z))    )
(P14 (GOAL * ADD)(W <P> V)(SEND <P>)(Y <X> Z)
--->
SAY(<X>)$
DEL((GOAL * ADD))$
INS((GOAL * PURGE))   )
(P15 (GOAL * ADD)(W <P> V)(Y <X> Z)
--->
REPL((GOAL * ADD)  $  (GOAL *S* ADD))$
INS((GOAL * NEXT ALONG))$
DO(GET-NEXT)  )
(P16 (GOAL * NEXT UP)
--->
DO(STEP1")  )
(P17 (GOAL * NEXT ALONG)(Y ZERO Z)
--->
REPL((Y ZERO Z)  $  (Y ONE Z))$
DEL((GOAL * NEXT ALONG))$
INS((GOAL * NEXT UP))  )
(P17A (GOAL * NEXT ALONG)(Y ONE Z)
--->
REPL((Y ONE Z)  $  (Y TWO Z))$
DEL((GOAL * NEXT ALONG))$
INS((GOAL * NEXT UP))   )
(P17B (GOAL * NEXT ALONG)(Y TWO Z)
```

```
====>
REPL((Y TWO Z)  ;  (Y THREE Z))$
DEL((GOAL * NEXT ALONG))$
INS((GOAL * NEXT UP))    )
(P17C (GOAL * NEXT ALONG)(Y THREE Z)
====>
REPL((Y THREE Z)  ;  (Y FOUR Z))$
DEL((GOAL * NEXT ALONG))$
INS((GOAL * NEXT UP))    )
(P17D (GOAL * NEXT ALONG)(Y FOUR Z)
====>
REPL((Y FOUR Z)  ;  (Y FIVE Z))$
DEL((GOAL * NEXT ALONG))$
INS((GOAL * NEXT UP))    )
(P17E (GOAL * NEXT ALONG)(Y FIVE Z)
====>
REPL((Y FIVE Z)  ;  (Y SIX Z))$
DEL((GOAL * NEXT ALONG))$
INS((GOAL * NEXT UP))    )
(P17F (GOAL * NEXT ALONG)(Y SIX Z)
====>
REPL((Y SIX Z)  ;  (Y SEVEN Z))$
DEL((GOAL * NEXT ALONG))$
INS((GOAL * NEXT UP))    )
(P17G (GOAL * NEXT ALONG)(Y SEVEN Z)
====>
REPL((Y SEVEN Z)  ;  (Y EIGHT Z))$
DEL((GOAL * NEXT ALONG))$
INS((GOAL * NEXT UP))    )
(P17H  (GOAL * NEXT ALONG)(Y EIGHT Z)
====>
REPL((Y EIGHT Z)  ;  (Y NINE Z))$
DEL((GOAL * NEXT ALONG))$
INS((GOAL * NEXT UP))    )
(P17I (GOAL * NEXT ALONG)(Y NINE Z)
====>
REPL((Y NINE Z)  ;  (Y TEN Z))$
DEL((GOAL * NEXT ALONG))$
INS((GOAL * NEXT UP))
(P17J (GOAL * NEXT ALONG)(Y TEN Z)
====>
REPL((Y TEN Z)  ;  (Y ELEVEN Z))$
DEL((GOAL * NEXT ALONG))$
INS((GOAL * NEXT UP))    )
(P18 (GOAL * NEXT UP)(W 0 V)
====>
REPL((W 0 V)  ;  (W 1 V))$
DEL((GOAL * NEXT UP))$
REPL((GOAL *S* ADD)  ;  (GOAL * ADD))    )
(P18A (GOAL * NEXT UP)(W 1 V)
====>
REPL((W 1 V)  ;  (W 2 V))$
DEL((GOAL * NEXT UP))$
REPL((GOAL *S* ADD)  ;  (GOAL * ADD))    )
(P18B (GOAL * NEXT UP)(W 2 V)
====>
REPL((W 2 V)  ;  (W 3 V))$
DEL((GOAL * NEXT UP))$
REPL((GOAL *S* ADD)  ;  (GOAL * ADD))    )
(P18C (GOAL * NEXT UP)(W 3 V)
====>
REPL((W 3 V)  ;  (W 4 V))$
DEL((GOAL * NEXT UP))$
REPL((GOAL *S* ADD)  ;  (GOAL * ADD))    )
(P24A  (GOAL * PURGE)(Y <Y> Z)(BOTTON B)(SEND C)
====>
```

```
REPL((BOTTOM B)  ;  (BOTTOM <X>));
REPL((SEND C)  ;  (OLD SEND C));
DEL((Y <X> Z))    )
(P24 (GOAL * PURGE)(Y <X> Z)
==>
DEL((Y <X> Z))    )
(P25 (GOAL * PURGE)(W <P> V)
==>
DEL((W <P> V))    )
(P26 (GOAL * PURGE)(BOTTOM 13)
==>
STOP)
(P27 (GOAL * PURGE)
==>
DEL((GOAL * PURGE)))
REPL((GOAL *G* ATTEND)  ;  (GOAL * ATTEND))    )
)
(PDLISTS:
(N-ADD-NV2
P9 P11 P13
P000 P00 P0
P1 P2 P3 P4 P5 P6
P6A P6B P6C
P7 P8 P10 P12
P13 P14 P15 P16
P24A
P24 P25 P26 P27
)
(GET-NEXT
P17 P17A P17B P17C P17D P17E P17F
P17G P17H P17I P17J
)
(STEPUP
P18 P18A P18B P18C
)
)
(DEF: STM((GOAL * ATTEND)(TOP TWELVE)(BOTTOM ZERO)(SEND 2)))
%%%END
%FILE
[EDIFIL Filed : DSKC:<SYPOS.PS3]
```

```
RUN BEGINS :

((GOAL * ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK J
UNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(GO)


STM: ((GO) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK J
UNK JUNK JUNK JUNK JUNK JUNK)
P00


STM: ((GOAL ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUN
K JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y 1 Z) (GOAL * SEND 2) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 2) (Y 1 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 2) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (NS 1) (GOAL *S* SEND 2) (Y 1 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P5

*******  1


STM: ((GOAL * NAME) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 2) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK
)
P6


STM: ((Y 2 Z) (GOAL NAME) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 2) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUN
K)
P7
```

```
STM: ((GOAL * SEND 2) (Y 2 Z) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 2) (Y 2 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 2) (Y 2 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 2) (Y 2 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P5

******* 2


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 2) (GOAL *S* SEND 2) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK
)
P6A


STM: ((Y 3 Z) (GOAL MARK) (SAID 2) (GOAL COUNT) (GOAL *S* SEND 2) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUN
K)
P7


STM: ((Y 3 Z) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P8
INPUT TO STM:
*(MANY BOTTOM BEFORE)


STM: ((MANY BOTTOM BEFORE) (GOAL * ATTEND) (Y 3 Z) (TOP TWELVE) (BOTT
OM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9


STM: ((GOAL * RECALL BOTTOM) (GOAL *S* ATTEND) (Y 3 Z) (TOP TWELVE) (
BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

******* (BOTTOM ZERO)
```

```
STM: ((BOTTOM ZERO) (GOAL * ATTEND) (Y 3 Z) (TOP TWELVE) (SEND 2) JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)


STM: ((MANY TUBE) (GOAL * ATTEND) (BOTTOM ZERO) (Y 3 Z) (TOP TWELVE)
(SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (BOTTOM ZERO) (Y 3 Z) (T
OP TWELVE) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

********  (SEND 2)


STM: ((SEND 2) (GOAL * ATTEND) (BOTTOM ZERO) (Y 3 Z) (TOP TWELVE) JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM NOW)


STM: ((MANY BOTTOM NOW) (GOAL * ATTEND) (SEND 2) (BOTTOM ZERO) (Y 3 Z
) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((Y ZERO Z) (W 0 V) (GOAL * ADD) (GOAL *S* ATTEND) (BOTTOM ZERO)
 (SEND 2) (Y 3 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((GOAL * NEXT ALONG) (GOAL *S* ADD) (W 0 V) (Y ZERO Z) (GOAL *S*
 ATTEND) (BOTTOM ZERO) (SEND 2) (Y 3 Z) (TOP TWELVE) JUNK JUNK JUNK J
UNK)
P17


STM: ((GOAL * NEXT UP) (Y ONE Z) (GOAL *S* ADD) (W 0 V) (GOAL *S* ATT
END) (BOTTOM ZERO) (SEND 2) (Y 3 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16


STM: ((GOAL * NEXT UP) (Y ONE Z) (GOAL *S* ADD) (W 0 V) (GOAL *S* ATT
END) (BOTTOM ZERO) (SEND 2) (Y 3 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P18


STM: ((W 1 V) (Y ONE Z) (GOAL * ADD) (GOAL *S* ATTEND) (BOTTOM ZERO)
(SEND 2) (Y 3 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P15
```

```
STM: ((GOAL * NEXT ALONG) (GOAL *S* ADD) (W 1 V) (Y ONE Z) (GOAL *S*
ATTEND) (BOTTOM ZERO) (SEND 2) (Y 3 Z) (TOP TWELVE) JUNK JUNK JUNK JU
NK)
P17A


STM: ((GOAL * NEXT UP) (Y TWO Z) (GOAL *S* ADD) (W 1 V) (GOAL *S* ATT
END) (BOTTOM ZERO) (SEND 2) (Y 3 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16


STM: ((GOAL * NEXT UP) (Y TWO Z) (GOAL *S* ADD) (W 1 V) (GOAL *S* ATT
END) (BOTTOM ZERO) (SEND 2) (Y 3 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16A


STM: ((W 2 V) (Y TWO Z) (GOAL * ADD) (GOAL *S* ATTEND) (BOTTOM ZERO)
(SEND 2) (Y 3 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P14

******** TWO


STM: ((GOAL * PURGE) (W 2 V) (SEND 2) (Y TWO Z) (GOAL *S* ATTEND) (BO
TTOM ZERO) (Y 3 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P24A


STM: ((GOAL * PURGE) (BOTTOM TWO) (OLD SEND 2) (W 2 V) (GOAL *S* ATTE
ND) (Y 3 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (BOTTOM TWO) (OLD SEND 2) (W 2 V) (GOAL *S* ATTE
ND) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (BOTTOM TWO) (OLD SEND 2) (GOAL *S* ATTEND) (TOP
 TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((BOTTOM TWO) (OLD SEND 2) (GOAL * ATTEND) (TOP TWELVE) JUNK JUN
K JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*(SEND 4)


STM: ((SEND 4) (GOAL ATTEND) (BOTTOM TWO) (TOP TWELVE) JUNK JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK)
P1
```

```
STM: ((Y 1 Z) (GOAL * SEND 4) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 1 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 4) (Y 1 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

******* 1


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P6


STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK
)
P8


STM: ((GOAL * SEND 4) (Y 2 Z) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 2 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y 2 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 4) (Y 2 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

******* 2
```

```
STM: ((GOAL * MARK) (GOAL COUNT) (SAID 2) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P6A


STM: ((Y 3 Z) (GOAL MARK) (SAID 2) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK
)
P8


STM: ((GOAL * SEND 4) (Y 3 Z) (GOAL *2* ATTEND) (TOP TWELVE) (BOTTOM
TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 3 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y 3 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 4) (Y 3 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

********  3


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 3) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P6B


STM: ((Y 4 Z) (GOAL MARK) (SAID 3) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK
)
P8


STM: ((GOAL * SEND 4) (Y 4 Z) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 4 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3
```

```
STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y A Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (GS 1) (GOAL *S* SEND 4) (Y A Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

*******  4


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 4) (GOAL *S* SEND 1) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P6C


STM: ((Y S Z) (GOAL MARK) (SAID 4) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK
)
P7


STM: ((Y S Z) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND 1) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM BEFORE)


STM: ((MANY BOTTOM BEFORE) (GOAL * ATTEND) (Y S Z) (TOP TWELVE) (BOTT
OM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9


STM: ((GOAL * RECALL BOTTOM) (GOAL *S* ATTEND) (Y S Z) (TOP TWELVE) (
BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

*******  (BOTTOM TWO)


STM: ((BOTTOM TWO) (GOAL * ATTEND) (Y S Z) (TOP TWELVE) (SEND 4) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)


STM: ((MANY TUBE) (GOAL * ATTEND) (BOTTOM TWO) (Y S Z) (TOP TWELVE) (
SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (BOTTOM TWO) (Y S Z) (TO
```

```
P TWELVE) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

*******  (SEND 4)


STM: ((SEND 4) (GOAL * ATTEND) (BOTTOM TWO) (Y 5 Z) (TOP TWELVE) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM NOW)


STM: ((MANY BOTTOM NOW) (GOAL * ATTEND) (SEND 4) (BOTTOM TWO) (Y 5 Z)
 (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((Y TWO Z) (W 0 V) (GOAL * ADD) (GOAL *S* ATTEND) (BOTTOM TWO) (
SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((GOAL * NEXT ALONG) (GOAL *S* ADD) (W 0 V) (Y TWO Z) (GOAL *S*
ATTEND) (BOTTOM TWO) (SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUN
K)
P17B


STM: ((GOAL * NEXT UP) (Y THREE Z) (GOAL *S* ADD) (W 0 V) (GOAL *S* A
TTEND) (BOTTOM TWO) (SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK
)
P16


STM: ((GOAL * NEXT UP) (Y THREE Z) (GOAL *S* ADD) (W 0 V) (GOAL *S* A
TTEND) (BOTTOM TWO) (SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK
)
P18


STM: ((W 1 V) (Y THREE Z) (GOAL * ADD) (GOAL *S* ATTEND) (BOTTOM TWO)
 (SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((GOAL * NEXT ALONG) (GOAL *S* ADD) (W 1 V) (Y THREE Z) (GOAL *S
* ATTEND) (BOTTOM TWO) (SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK J
UNK)
P17C


STM: ((GOAL * NEXT UP) (Y FOUR Z) (GOAL *S* ADD) (W 1 V) (GOAL *S* AT
TEND) (BOTTOM TWO) (SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16
```

```
STM: ((GOAL * NEXT UP) (Y FOUR Z) (GOAL *S* ADD) (W 1 V) (GOAL *S* AT
TEND) (BOTTOM TWO) (SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P18A


STM: ((W 2 V) (Y FOUR Z) (GOAL * ADD) (GOAL *S* ATTEND) (BOTTOM TWO)
(SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((GOAL * NEXT ALONG) (GOAL *S* ADD) (W 2 V) (Y FOUR Z) (GOAL *S*
 ATTEND) (BOTTOM TWO) (SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JU
NK)
P17D


STM: ((GOAL * NEXT UP) (Y FIVE Z) (GOAL *S* ADD) (W 2 V) (GOAL *S* AT
TEND) (BOTTOM TWO) (SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16


STM: ((GOAL * NEXT UP) (Y FIVE Z) (GOAL *S* ADD) (W 2 V) (GOAL *S* AT
TEND) (BOTTOM TWO) (SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P18B


STM: ((W 3 V) (Y FIVE Z) (GOAL * ADD) (GOAL *S* ATTEND) (BOTTOM TWO)
(SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((GOAL * NEXT ALONG) (GOAL *S* ADD) (W 3 V) (Y FIVE Z) (GOAL *S*
 ATTEND) (BOTTOM TWO) (SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JU
NK)
P17E


STM: ((GOAL * NEXT UP) (Y SIX Z) (GOAL *S* ADD) (W 3 V) (GOAL *S* ATT
END) (BOTTOM TWO) (SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16


STM: ((GOAL * NEXT UP) (Y SIX Z) (GOAL *S* ADD) (W 3 V) (GOAL *S* ATT
END) (BOTTOM TWO) (SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P18C


STM: ((W 4 V) (Y SIX Z) (GOAL * ADD) (GOAL *S* ATTEND) (BOTTOM TWO) (
SEND 4) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P14

******** SIX
```

```
STM: ((GOAL * PURGE) (W 4 V) (SEND 4) (Y SIX Z) (GOAL *S* ATTEND) (BO
TTOM TWO) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P24A


STM: ((GOAL * PURGE) (BOTTOM SIX) (OLD SEND 4) (W 4 V) (GOAL *S* ATTE
ND) (Y 5 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (BOTTOM SIX) (OLD SEND 4) (W 4 V) (GOAL *S* ATTE
ND) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (BOTTOM SIX) (OLD SEND 4) (GOAL *S* ATTEND) (TOP
 TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((BOTTOM SIX) (OLD SEND 4) (GOAL * ATTEND) (TOP TWELVE) JUNK JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*(SEND 3)


STM: ((SEND 3) (GOAL ATTEND) (BOTTOM SIX) (TOP TWELVE) JUNK JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y 1 Z) (GOAL * SEND 3) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 3) (Y 1 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 3) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 3) (Y 1 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P5

******** 1


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 3) (GOAL *S*
```

```
 ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P6


STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 3) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK
)
P8


STM: ((GOAL * SEND 3) (Y 2 Z) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 3) (Y 2 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 3) (Y 2 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (AS 1) (GOAL *S* SEND 3) (Y 2 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P5

******** 2


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 2) (GOAL *S* SEND 3) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P6A


STM: ((Y 3 Z) (GOAL MARK) (SAID 2) (GOAL COUNT) (GOAL *S* SEND 3) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK
)
P8


STM: ((GOAL * SEND 3) (Y 3 Z) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 3) (Y 3 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 3) (Y 3 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P4
```

```
STM: ((GOAL * COUNT) (@S 1) (GOAL *S* SEND 3) (Y 3 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P5

******** 3


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 3) (GOAL *S* SEND 3) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P6B


STM: ((Y 4 Z) (GOAL MARK) (SAID 3) (GOAL COUNT) (GOAL *S* SEND 3) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK
)
P7


STM: ((Y 4 Z) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND 3) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM BEFORE)


STM: ((MANY BOTTOM BEFORE) (GOAL * ATTEND) (Y 4 Z) (TOP TWELVE) (BOTT
OM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9


STM: ((GOAL * RECALL BOTTOM) (GOAL *S* ATTEND) (Y 4 Z) (TOP TWELVE) (
BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

********  (BOTTOM SIX)


STM: ((BOTTOM SIX) (GOAL * ATTEND) (Y 4 Z) (TOP TWELVE) (SEND 3) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)


STM: ((MANY TUBE) (GOAL * ATTEND) (BOTTOM SIX) (Y 4 Z) (TOP TWELVE) (
SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (BOTTOM SIX) (Y 4 Z) (TO
P TWELVE) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12
```

```
********  (SEND 3)


STM: ((SEND 3) (GOAL * ATTEND) (BOTTOM SIX) (Y 4 Z) (TOP TWELVE) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM NOW)


STM: ((MANY BOTTOM NOW) (GOAL * ATTEND) (SEND 3) (BOTTOM SIX) (Y 4 Z)
 (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((Y SIX Z) (W 0 V) (GOAL * ADD) (GOAL *S* ATTEND) (BOTTOM SIX) (
SEND 3) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((GOAL * NEXT ALONG) (GOAL *S* ADD) (W 0 V) (Y SIX Z) (GOAL *S*
ATTEND) (BOTTOM SIX) (SEND 3) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK JUN
K)
P17F


STM: ((GOAL * NEXT UP) (Y SEVEN Z) (GOAL *S* ADD) (W 0 V) (GOAL *S* A
TTEND) (BOTTOM SIX) (SEND 3) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK
)
P16


STM: ((GOAL * NEXT UP) (Y SEVEN Z) (GOAL *S* ADD) (W 0 V) (GOAL *S* A
TTEND) (BOTTOM SIX) (SEND 3) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK
)
P18


STM: ((W 1 V) (Y SEVEN Z) (GOAL * ADD) (GOAL *S* ATTEND) (BOTTOM SIX)
 (SEND 3) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((GOAL * NEXT ALONG) (GOAL *S* ADD) (W 1 V) (Y SEVEN Z) (GOAL *S
* ATTEND) (BOTTOM SIX) (SEND 3) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK J
UNK)
P17G


STM: ((GOAL * NEXT UP) (Y EIGHT Z) (GOAL *S* ADD) (W 1 V) (GOAL *S* A
TTEND) (BOTTOM SIX) (SEND 3) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK
)
P16
```

```
STM: ((GOAL * NEXT UP) (Y EIGHT Z) (GOAL *S* ADD) (W 1 V) (GOAL *S* A
TTEND) (BOTTOM SIX) (SEND 3) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK
)
P18A


STM: ((W 2 V) (Y EIGHT Z) (GOAL * ADD) (GOAL *S* ATTEND) (BOTTOM SIX)
 (SEND 3) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((GOAL * NEXT ALONG) (GOAL *S* ADD) (W 2 V) (Y EIGHT Z) (GOAL *S
* ATTEND) (BOTTOM SIX) (SEND 3) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK J
UNK)
P17H


STM: ((GOAL * NEXT UP) (Y NINE Z) (GOAL *S* ADD) (W 2 V) (GOAL *S* AT
TEND) (BOTTOM SIX) (SEND 3) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16


STM: ((GOAL * NEXT UP) (Y NINE Z) (GOAL *S* ADD) (W 2 V) (GOAL *S* AT
TEND) (BOTTOM SIX) (SEND 3) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P18B


STM: ((W 3 V) (Y NINE Z) (GOAL * ADD) (GOAL *S* ATTEND) (BOTTOM SIX)
(SEND 3) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P14

********  NINE


STM: ((GOAL * PURGE) (W 3 V) (SEND 3) (Y NINE Z) (GOAL *S* ATTEND) (B
OTTOM SIX) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P24A


STM: ((GOAL * PURGE) (BOTTOM NINE) (OLD SEND 3) (W 3 V) (GOAL *S* ATT
END) (Y 4 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (BOTTOM NINE) (OLD SEND 3) (W 3 V) (GOAL *S* ATT
END) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (BOTTOM NINE) (OLD SEND 3) (GOAL *S* ATTEND) (TO
P TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((BOTTOM NINE) (OLD SEND 3) (GOAL * ATTEND) (TOP TWELVE) JUNK JU
NK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
```

```
FOCO
INPUT TO STM:
*(SEND 1)


STM: ((SEND 1) (GOAL ATTEND) (BOTTOM NINE) (TOP TWELVE) JUNK JUNK JUN
K JUNK JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y 1 Z) (GOAL * SEND 1) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 1) (Y 1 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 1) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 1) (Y 1 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK)
P5

******** 1


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 1) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK
)
P6


STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 1) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM NINE) (SEND 1) JUNK JUNK JUNK JUN
K)
P7


STM: ((Y 2 Z) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM NINE) (SEND 1) JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P8
INPUT TO STM:
*(MANY BOTTOM BEFORE)


STM: ((MANY BOTTOM BEFORE) (GOAL * ATTEND) (Y 2 7) (TOP TWELVE) (BOTT
OM NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9
```

```
STM: ((GOAL * RECALL BOTTOM) (GOAL *3* ATTEND) (Y 2 Z) (TOP TWELVE) (
BOTTOM NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

********  (BOTTOM NINE)


STM: ((BOTTOM NINE) (GOAL * ATTEND) (Y 2 Z) (TOP TWELVE) (SEND 1) JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)



STM: ((MANY TUBE) (GOAL * ATTEND) (BOTTOM NINE) (Y 2 Z) (TOP TWELVE)
(SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *5* ATTEND) (BOTTOM NINE) (Y 2 Z) (T
OP TWELVE) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

********  (SEND 1)


STM: ((SEND 1) (GOAL * ATTEND) (BOTTOM NINE) (Y 2 Z) (TOP TWELVE) JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM NOW)



STM: ((MANY BOTTOM NOW) (GOAL * ATTEND) (SEND 1) (BOTTOM NINE) (Y 2 Z
) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((Y NINE Z) (W 0 V) (GOAL * ADD) (GOAL *5* ATTEND) (BOTTOM NINE)
 (SEND 1) (Y 2 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((GOAL * NEXT ALONG) (GOAL *5* ADD) (W 0 V) (Y NINE Z) (GOAL *3*
 ATTEND) (BOTTOM NINE) (SEND 1) (Y 2 Z) (TOP TWELVE) JUNK JUNK JUNK J
UNK)
P17I


STM: ((GOAL * NEXT UP) (Y TEN Z) (GOAL *5* ADD) (W 0 V) (GOAL *5* ATT
END) (BOTTOM NINE) (SEND 1) (Y 2 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16
```

```
STM: ((GOAL * NEXT UP) (Y TEN Z) (GOAL *S* ADD) (W 0 V) (GOAL *S* ATT
END) (BOTTOM NINE) (SEND 1) (Y 2 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P18


STM: ((W 1 V) (Y TEN Z) (GOAL * ADD) (GOAL *S* ATTEND) (BOTTOM NINE)
(SEND 1) (Y 2 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P14

*******  TEN


STM: ((GOAL * PURGE) (W 1 V) (SEND 1) (Y TEN Z) (GOAL *S* ATTEND) (BO
TTOM NINE) (Y 2 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P24A


STM: ((GOAL * PURGE) (BOTTOM TEN) (OLD SEND 1) (W 1 V) (GOAL *S* ATTE
ND) (Y 2 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (BOTTOM TEN) (OLD SEND 1) (W 1 V) (GOAL *S* ATTE
ND) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (BOTTOM TEN) (OLD SEND 1) (GOAL *S* ATTEND) (TOP
 TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((BOTTOM TEN) (OLD SEND 1) (GOAL * ATTEND) (TOP TWELVE) JUNK JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*()


STM: (NIL (GOAL ATTEND) (BOTTOM TEN) (TOP TWELVE) JUNK JUNK JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK)


END OF RUN

*
*^C
```

ADDENDUM 3.

LISTING AND EXECUTION TRACE FOR

N-ADD-NV: MODIFIED COUNTING METHOD.

```
 P 500
(VARIABLES:
(A GENC)
(B GENC)
(C GENC)
(D GENC)
(E GENC)
(<X> SEQC)
(<P> SEQC)
)
(PRODUCTIONS:
(P000 (GOAL * ATTEND)(OLD SEND C)
-->
REPL((GOAL * ATTEND)  ;  (GOAL + ATTEND));
DEL((OLD SEND C));
USER()    )
(P00  (GOAL * ATTEND)(GO)
-->
REPL((GOAL * ATTEND)  ;  (GOAL + ATTEND));
DEL((GO))  )
(P0 (GOAL * ATTEND)
-->
USER()  )
(P1 (GOAL + ATTEND)(TOP A)(BOTTOM B)(SEND C)
-->
REPL((GOAL + ATTEND)  ;  (GOAL *S* ATTEND));
INS((GOAL * SEND C));
INS((Y ONE Z))    )
(P2 (GOAL * SEND C)
-->
REPL((GOAL * SEND C)  ;  (GOAL *S* SEND C));
INS((GOAL * PUSH BUTTON))  )
(P3 (GOAL * PUSH BUTTON)
-->
DEL((GOAL * PUSH BUTTON));
INS((ELM A));
INS((GOAL * SUBIT))  )
(P4  (GOAL * SUBIT)(ELM A)
-->
DEL((GOAL * SUBIT));
INS((QS 1));
DEL((ELM A));
INS((GOAL * COUNT))  )
(P5  (GOAL * COUNT)(QS 1)(Y <X> Z)
-->
REPL((GOAL * COUNT)  ;  (GOAL + COUNT));
DEL((QS 1));
SAY(<X>);
REPL((Y <X> Z)  ;  (SAID <X>));
INS((GOAL * MARK))  )
(P6 (GOAL * MARK)(SAID ONE)
-->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y TWO Z))    )
(P6A  (GOAL * MARK)(SAID TWO)
-->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y THREE Z))    )
(P6B  (GOAL * MARK)(SAID THREE)
-->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y FOUR Z))    )
(P6C  (GOAL * MARK)(SAID FOUR)
-->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y FIVE Z))    )
```

```
(P7 (GOAL * MARK)(GOAL * COUNT)(GOAL *S* SEND C)
(SAID C)
==>
DEL((GOAL *S* SEND C));
DEL((GOAL * MARK));
DEL((GOAL * COUNT));
DEL (SAID C));
REPL((GOAL *S* ATTEND)  ;  (GOAL * ATTEND))  )
(P8  (GOAL * MARK)(GOAL * COUNT)(GOAL *S* SEND C)(SAID D)
==>
REPL((GOAL *S* SEND C)  ;  (GOAL * SEND C));
DEL((GOAL * MARK));
DEL((GOAL * COUNT));
DEL((SAID D))     )
(P9 (GOAL * ATTEND)(MANY BOTTOM BEFORE)
==>
REPL((GOAL * ATTEND)  ;  (GOAL *S* ATTEND));
DEL((MANY BOTTOM BEFORE));
INS((GOAL * RECALL BOTTOM))   )
(P10  (GOAL * RECALL BOTTOM)(BOTTOM B)
==>
SAY((BOTTOM B));
DEL((GOAL * RECALL BOTTOM));
REPL((GOAL *S* ATTEND)  ;  (GOAL * ATTEND))  )
(P11 (GOAL * ATTEND)(MANY TUBE)
==>
REPL((GOAL * ATTEND)  ;  (GOAL *S* ATTEND));
DEL((MANY TUBE));
INS((GOAL * RECALL  TUBE))   )
(P12  (GOAL * RECALL TUBE)(SEND C)
==>
DEL((GOAL * RECALL TUBE));
REPL((GOAL *S* ATTEND)  ;  (GOAL * ATTEND));
SAY((SEND C))   )
(P13  (GOAL * ATTEND)(MANY BOTTOM NOW)
==>
REPL((GOAL * ATTEND)  ;  (GOAL *S* ATTEND));
DEL((MANY BOTTOM NOW));
INS((GOAL * SMALL));
INS((GOAL *S* MOVE));
INS((GOAL *S* MARK));
INS((GOAL *S* COMPARE));
INS((GOAL *S* ADD));
INS((Y ONE Z))   )
(P14  (GOAL * SMALL)
==>
REPL((GOAL * SMALL)  ;  (GOAL *S* SMALL));
REPL((GOAL *S* MOVE)  ;  (GOAL * MOVE))   )
(P15 (GOAL * MOVE)(Y <X> Z)
==>
REPL((GOAL * MOVE)  ;  (GOAL *S* MOVE));
REPL((GOAL *S* MARK)  ;  (GOAL * MARK));
REPL((Y <X> Z)  ;  (NUMBER <X>));
DO(MARKER)   )
(P17  (GOAL * MARK)(NUMBER ONE)
==>
REPL((GOAL * MARK)  ;  (GOAL *S* MARK));
REPL((GOAL *S* COMPARE)  ;  (GOAL * COMPARE));
INS((Y TWO Z))   )
(P17A (GOAL * MARK)(NUMBER TWO)
==>
REPL((GOAL * MARK)  ;  (GOAL *S* MARK));
REPL((GOAL *S* COMPARE)  ;  (GOAL * COMPARE));
INS((Y THREE Z))   )
(P17B  (GOAL * MARK)(NUMBER THREE)
==>
```

```
REPL((GOAL * MARK)  ;  (GOAL *C* MARK))$
REPL((GOAL *S* COMPARE)  ;  (GOAL * COMPARE))$
INS((Y FOUR Z))    )
(P17C  (GOAL * MARK)(NUMBER FOUR)
-->
REPL((GOAL * MARK)  ;  (GOAL *S* MARK))$
REPL((GOAL *S* COMPARE)  ;  (GOAL * COMPARE))$
INS((Y FIVE Z))    )
(P17D  (GOAL * MARK)(NUMBER FIVE)
-->
REPL((GOAL * MARK)  ;  (GOAL *S* MARK))$
REPL((GOAL *S* COMPARE)  ;  (GOAL * COMPARE))$
INS((Y SIX Z))    )
(P17E  (GOAL * MARK)(NUMBER SIX)
-->
REPL((GOAL * MARK)  ;  (GOAL *S* MARK))$
REPL((GOAL *S* COMPARE)  ;  (GOAL * COMPARE))$
INS((Y SEVEN Z))    )
(P17F  (GOAL * MARK)(NUMBER SEVEN)
-->
REPL((GOAL * MARK)  ;  (GOAL *S* MARK))$
REPL((GOAL *S* COMPARE)  ;  (GOAL * COMPARE))$
INS((Y EIGHT Z))    )
(P17G  (GOAL * MARK)(NUMBER EIGHT)
-->
REPL((GOAL * MARK)  ;  (GOAL *S* MARK))$
REPL((GOAL *S* COMPARE)  ;  (GOAL * COMPARE))$
INS((Y NINE Z))    )
(P17H  (GOAL * MARK)(NUMBER NINE)
-->
REPL((GOAL * MARK)  ;  (GOAL *S* MARK))$
REPL((GOAL *S* COMPARE)  ;  (GOAL * COMPARE))$
INS((Y TEN Z))    )
(P17I  (GOAL * MARK)(NUMBER TEN)
-->
REPL((GOAL * MARK)  ;  (GOAL *S* MARK))$
REPL((GOAL *S* COMPARE)  ;  (GOAL * COMPARE))$
INS((Y ELEVEN Z))    )
(P17J  (GOAL * MARK)(NUMBER ELEVEN)
-->
REPL((GOAL * MARK)  ;  (GOAL *S* MARK))$
REPL((GOAL *S* COMPARE)  ;  (GOAL * COMPARE))$
INS((Y TWELVE Z))    )
(P17K  (GOAL * MARK)(NUMBER TWELVE)
-->
REPL((GOAL * MARK)  ;  (GOAL *S* MARK))$
REPL((GOAL *S* COMPARE)  ;  (GOAL * COMPARE))$
INS((Y THIRTEEN Z))    )
(P24A  (GOAL * PURGE)(Y <X> Z)(BOTTOM B)(SEND C)
-->
REPL((BOTTOM B)  ;  (BOTTOM <X>))$
REPL((SEND C)  ;  (OLD SEND C))$
DEL((Y <X> Z))    )
(P24 (GOAL * PURGE)(Y <X> Z)
-->
DEL((Y <X> Z))    )
(P25 (GOAL * PURGE)(D <X> Z)
-->
DEL((D <X> Z))    )
(P26 (GOAL * PURGE)(BOTTOM TWELVE)
-->
STOP)
(P27 (GOAL * PURGE)
-->
DEL((GOAL * PURGE))$
REPL((GOAL *S* ATTEND)  ;  (GOAL * ATTEND))    )
```

```
(P130A  (GOAL * SMALL)(BOTTOM B)(SEND B)
-->
DEL((GOAL * SMALL));
REPL((GOAL *S* ADD)  ;  (GOAL * ADD));
INS((SMALLER B));
INS((LARGER B))   )
(P130B  (GOAL * SMALL)(BOTTOM ZERO)(SEND C)
-->
DEL((GOAL * SMALL));
DEL((GOAL *S* MOVE));
DEL((GOAL *S* MARK));
DEL((GOAL *S* COMPARE));
DEL((GOAL *S* ADD));
INS((GOAL * PURGE));
REPL((BOTTOM ZERO  ;  (BOTTOM C));
REPL((SEND C)  ;  (OLD SEND C));
SAY(C)    )
(P180  (GOAL * COMPARE)(GOAL *S* SMALL)(BOTTOM B)(NUMBER B)
-->
REPL((GOAL * COMPARE)  ;  (GOAL *S* COMPARE));
REPL((GOAL *S* SMALL)  ;  (GOAL *S* LARGE));
REPL((GOAL *S* MOVE)  ;  (GOAL * MOVE));
REPL((NUMBER B)  ;  (SMALLER B))    )
(P190  (GOAL * COMPARE)(GOAL *S* SMALL)(SEND C)(NUMBER C)
-->
REPL((GOAL * COMPARE)  ;  (GOAL *S* COMPARE));
REPL((GOAL *S* SMALL)  ;  (GOAL *S* LARGE));
REPL((GOAL *S* MOVE)  ;  (GOAL * MOVE));
REPL((NUMBER C)  ;  (SMALLER C))    )
(P200  (GOAL * COMPARE)(GOAL *S* LARGE)
(BOTTOM B)(NUMBER B)
-->
REPL((GOAL * COMPARE)  ;  (GOAL *S* COMPARE));
DEL((GOAL *S* LARGE));
REPL((GOAL *S* ADD)  ;  (GOAL * ADD));
REPL((NUMBER B)  ;  (LARGER B));
INS((W ONE V))    )
(P210  (GOAL * COMPARE)(GOAL *S* LARGE)(SEND C)(NUMBER C)
-->
REPL((GOAL * COMPARE)  ;  (GOAL *S* COMPARE));
DEL((GOAL *S* LARGE));
REPL((GOAL *S* ADD)  ;  (GOAL * ADD));
REPL((NUMBER C)  ;  (LARGER C));
INS((W ONE V))    )
(P210A  (GOAL * COMPARE)(GOAL *S* ADD)(NUMBER E)
(ABS (W <P> V))(ABS (GOAL *S* FIND PLACE))
-->
REPL((GOAL * COMPARE)  ;  (GOAL *S* COMPARE));
REPL((GOAL *S* MOVE)  ;  (GOAL * MOVE));
DEL((NUMBER E))    )
(P220  (GOAL * ADD)(W <P> V)(SMALLER <P>)(Y <X> Z)
(LARGER E)
-->
DEL((GOAL * ADD));
DEL((GOAL *S* MOVE));
DEL((GOAL *S* MARK));
DEL((GOAL *S* COMPARE));
DEL((SMALLER <P>));
DEL((LARGER E));
INS((GOAL * PURGE));
DEL((W <P> V));
SAY(<X>)    )
(P230  (GOAL * ADD)(W ONE V)
-->
REPL((GOAL * ADD)  ;  (GOAL *S* ADD));
REPL((GOAL *S* MOVE)  ;  (GOAL * MOVE));
```

```
REPL ((W ONE V)   ;  (W TWO V))     )
(P230A  (GOAL * ADD)(W TWO V)
   -->
REPL((GOAL * ADD)  ;  (GOAL *S* ADD));
REPL ((GOAL *S* MOVE)  ;  (GOAL * MOVE));
REPL((W TWO V)  ;  (W THREE V))    )
(P230B  (GOAL * ADD)(W THREE V)
   -->
REPL((GOAL * ADD)  ;  (GOAL *S* ADD));
REPL((GOAL *S* MOVE)  ;  (GOAL * MOVE));
REPL((W THREE V)  ;  (W FOUR V))  )
(P230C  (GOAL * ADD)(W FOUR V)
   -->
REPL((GOAL * ADD)  ;  (GOAL *S* ADD));
REPL((GOAL *S* MOVE)  ;  (GOAL * MOVE));
REPL((W FOUR V)  ;  (W FIVE V))   )
(P240 (GOAL * COMPARE)(W <I> V)
(NUMBER E)
   -->
DEL((NUMBER E));
REPL((GOAL * COMPARE)  ;  (GOAL *S* COMPARE));
REPL((GOAL *S* ADD)  ;  (GOAL * ADD)   )
(P250  (GOAL * ADD)(LARGER ONE)
   -->
REPL((Y ONE 2)  ;  (Y TWO 2));
INS((W ONE V))   )
(P260  (GOAL * COMPARE)(GOAL *S* ADD)(GOAL *S* FIND PLACE)
(LARGER E)(NUMBER E)
   -->
REPL((GOAL * COMPARE)  ;  (GOAL *S* COMPARE));
REPL((GOAL *S* ADD)  ;  (GOAL * ADD));
DEL((GOAL *S* FIND PLACE));
DEL((NUMBER E));
INS((W ONE V))     )
(P270  (GOAL * ADD)(LARGER E)(SMALLER E)
   -->
REPL((GOAL * ADD)  ;  (GOAL *S* ADD));
REPL((GOAL *S* MOVE)  ;  (GOAL * MOVE));
INS((GOAL *S* FIND PLACE))   )
(P280  (GOAL * COMPARE)(GOAL *S* ADD)(GOAL *S* FIND PLACE)
   -->
REPL((GOAL * COMPARE)  ;  (GOAL *S* COMPARE));
REPL((GOAL *S* MOVE)  ;  (GOAL * MOVE));
DEL((NUMBER <X>))    )
)
(PDLISTS
(N-ADDRESS
P9 P11 P13
P000 P00 P0
P1 P2 P3 P4 P5 P6
P6A P6B P6C
P7 P8 P10 P12
P130A P130B
P14      P16
P180 P190 P200 P210 P210A
P220 P230 P230A P230B P230C
P240 P250 P260 P270 P280)
P24A
P24 P25 P26 P27
)
(MARKER
P17 P17A P17B P17C P17D P17E P17F
P17G P17H P17I
)
)
(DEF: STM((GOAL * ATTEND)(TOP TWELVE)(BOTTOM ZERO)(SEND TWO)))
```

```
RUN BEGINS :

((GOAL * ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND TWO) JUNK JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(GO)


STM: ((GO) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND TWO) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P00


STM: ((GOAL ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND TWO) JUNK JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y ONE Z) (GOAL * SEND TWO) (GOAL *S* ATTEND) (TOP TWELVE) (BOT
TOM ZERO) (SEND TWO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUN
K)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND TWO) (Y ONE Z) (GOAL *S* AT
TEND) (TOP TWELVE) (BOTTOM ZERO) (SEND TWO) JUNK JUNK JUNK JUNK JUNK
JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND TWO) (Y ONE Z) (GOAL *S*
ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND TWO) JUNK JUNK JUNK JUNK JUN
K JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND TWO) (Y ONE Z) (GOAL *S* A
TTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND TWO) JUNK JUNK JUNK JUNK JUNK
 JUNK JUNK JUNK)
P5

********  ONE


STM: ((GOAL * MARK) (GOAL COUNT) (SAID ONE) (GOAL *S* SEND TWO) (GOAL
 *S* ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND TWO) JUNK JUNK JUNK JUN
K JUNK JUNK JUNK JUNK)
P6


STM: ((Y TWO Z) (GOAL MARK) (SAID ONE) (GOAL COUNT) (GOAL *S* SEND TW
```

```
0) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND TWO) JUNK JUNK
JUNK JUNK JUNK JUNK JUNK)
P8


STM: ((GOAL * SEND TWO) (Y TWO Z) (GOAL *S* ATTEND) (TOP TWELVE) (BOT
TOM ZERO) (SEND TWO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUN
K)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND TWO) (Y TWO Z) (GOAL *S* AT
TEND) (TOP TWELVE) (BOTTOM ZERO) (SEND TWO) JUNK JUNK JUNK JUNK JUNK
JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND TWO) (Y TWO Z) (GOAL *S*
ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND TWO) JUNK JUNK JUNK JUNK JUN
K JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND TWO) (Y TWO Z) (GOAL *S* A
TTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND TWO) JUNK JUNK JUNK JUNK JUNK
 JUNK JUNK JUNK)
P5

*******  TWO


STM: ((GOAL * MARK) (GOAL COUNT) (SAID TWO) (GOAL *S* SEND TWO) (GOAL
 *S* ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND TWO) JUNK JUNK JUNK JUN
K JUNK JUNK JUNK JUNK)
P6A


STM: ((Y THREE Z) (GOAL MARK) (SAID TWO) (GOAL COUNT) (GOAL *S* SEND
TWO) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND TWO) JUNK JUN
K JUNK JUNK JUNK JUNK JUNK)
P7


STM: ((Y THREE Z) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND TW
O) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9
INPUT TO STM:
*(MANY BOTTOM BEFORE)


STM: ((MANY BOTTOM BEFORE) (GOAL * ATTEND) (Y THREE Z) (TOP TWELVE) (
BOTTOM ZERO) (SEND TWO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK
JUNK)
P9
```

```
STM: ((GOAL * RECALL BOTTOM) (GOAL *S* ATTEND) (Y THREE Z) (TOP TWELV
E) (BOTTOM ZERO) (SEND TWO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK J
UNK JUNK)
P10

********  (BOTTOM ZERO)


STM: ((BOTTOM ZERO) (GOAL * ATTEND) (Y THREE Z) (TOP TWELVE) (SEND TW
O) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)


STM: ((MANY TUBE) (GOAL * ATTEND) (BOTTOM ZERO) (Y THREE Z) (TOP TWEL
VE) (SEND TWO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (BOTTOM ZERO) (Y THREE Z
) (TOP TWELVE) (SEND TWO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUN
K JUNK)
P12

********  (SEND TWO)


STM: ((SEND TWO) (GOAL * ATTEND) (BOTTOM ZERO) (Y THREE Z) (TOP TWELV
E) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM NOW)


STM: ((MANY BOTTOM NOW) (GOAL * ATTEND) (SEND TWO) (BOTTOM ZERO) (Y T
HREE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUN
K)
P13


STM: ((Y ONE Z) (GOAL *S* ADD) (GOAL *S* COMPARE) (GOAL *S* MARK) (GO
AL *S* MOVE) (GOAL * SMALL) (GOAL *S* ATTEND) (SEND TWO) (BOTTOM ZERO
) (Y THREE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P130B

********  TWO


STM: ((GOAL * PURGE) (BOTTOM TWO) (OLD SEND TWO) (Y ONE Z) (GOAL *S*
ATTEND) (Y THREE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK J
UNK JUNK)
P24


STM: ((GOAL * PURGE) (BOTTOM TWO) (OLD SEND TWO) (GOAL *S* ATTEND) (Y
```

THREE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK J
UNK)
P24

STM: ((GOAL * PURGE) (BOTTOM TWO) (OLD SEND TWO) (GOAL *S* ATTEND) (T
OP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27

STM: ((BOTTOM TWO) (OLD SEND TWO) (GOAL * ATTEND) (TOP TWELVE) JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*(SEND FOUR)

STM: ((SEND FOUR) (GOAL ATTEND) (BOTTOM TWO) (TOP TWELVE) JUNK JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P1

STM: ((Y ONE Z) (GOAL * SEND FOUR) (GOAL *S* ATTEND) (TOP TWELVE) (BO
TTOM TWO) (SEND FOUR) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JU
NK)
P2

STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND FOUR) (Y ONE Z) (GOAL *S* A
TTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JUNK JUNK JUNK JUNK
JUNK JUNK JUNK JUNK JUNK)
P3

STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND FOUR) (Y ONE Z) (GOAL *S*
ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JUNK JUNK JUNK JU
NK JUNK JUNK JUNK)
P4

STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND FOUR) (Y ONE Z) (GOAL *S*
ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JUNK JUNK JUNK JUN
K JUNK JUNK JUNK)
P5

******** ONE

STM: ((GOAL * MARK) (GOAL COUNT) (SAID ONE) (GOAL *S* SEND FOUR) (GOA
L *S* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JUNK JUNK JU
NK JUNK JUNK JUNK JUNK)
P6

STM: ((Y TWO Z) (GOAL MARK) (SAID ONE) (GOAL COUNT) (GOAL *S* SEND FO
UR) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JUNK

```
 JUNK JUNK JUNK JUNK JUNK)
P8


STM: ((GOAL * SEND FOUR) (Y TWO Z) (GOAL *S* ATTEND) (TOP TWELVE) (BO
TTOM TWO) (SEND FOUR) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JU
NK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND FOUR) (Y TWO Z) (GOAL *S* A
TTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JUNK JUNK JUNK JUNK
 JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBTT) (CLM A) (GOAL *S* SEND FOUR) (Y TWO Z) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JUNK JUNK JUNK JU
NK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND FOUR) (Y TWO Z) (GOAL *S*
ATTEND) (TOP TWELVE) (BUTTON TWO) (SEND FOUR) JUNK JUNK JUNK JUNK JUN
K JUNK JUNK JUNK)
P5

*******  TWO


STM: ((GOAL * MARK) (GOAL COUNT) (SAID TWO) (GOAL *S* SEND FOUR) (GOA
L *S* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JUNK JUNK JU
NK JUNK JUNK JUNK JUNK)
P6A


STM: ((Y THREE Z) (GOAL MARK) (SAID TWO) (GOAL COUNT) (GOAL *S* SEND
 FOUR) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JU
NK JUNK JUNK JUNK JUNK JUNK)
P8


STM: ((GOAL * SEND FOUR) (Y THREE Z) (GOAL *S* ATTEND) (TOP TWELVE) (
BOTTOM TWO) (SEND FOUR) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK
 JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND FOUR) (Y THREE Z) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JUNK JUNK JUNK JU
NK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBTT) (CLM A) (GOAL *S* SEND FOUR) (Y THREE Z) (GOAL *
S* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JUNK JUNK JUNK
JUNK JUNK JUNK JUNK)
```

P4

STM: ((GOAL * COUNT) (QS 1) (GOAL *S* SEND FOUR) (Y THREE Z) (GOAL *S
* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JUNK JUNK JUNK J
UNK JUNK JUNK JUNK)
P5

******* THREE

STM: ((GOAL * MARK) (GOAL COUNT) (SAID THREE) (GOAL *S* SEND FOUR) (G
OAL *S* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JUNK JUNK
JUNK JUNK JUNK JUNK JUNK)
P6B

STM: ((Y FOUR Z) (GOAL MARK) (SAID THREE) (GOAL COUNT) (GOAL *S* SEND
FOUR) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK J
UNK JUNK JUNK JUNK JUNK JUNK)
P8

STM: ((GOAL * SEND FOUR) (Y FOUR Z) (GOAL *S* ATTEND) (TOP TWELVE) (B
OTTOM TWO) (SEND FOUR) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK J
UNK)
P2

STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND FOUR) (Y FOUR Z) (GOAL *S*
ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JUNK JUNK JUNK JUN
K JUNK JUNK JUNK JUNK)
P3

STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND FOUR) (Y FOUR Z) (GOAL *S
* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JUNK JUNK JUNK J
UNK JUNK JUNK JUNK)
P4

STM: ((GOAL * COUNT) (QS 1) (GOAL *S* SEND FOUR) (Y FOUR Z) (GOAL *S*
ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JUNK JUNK JUNK JU
NK JUNK JUNK JUNK)
P5

******* FOUR

STM: ((GOAL * MARK) (GOAL COUNT) (SAID FOUR) (GOAL *S* SEND FOUR) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JUNK JUNK J
UNK JUNK JUNK JUNK JUNK)
P6C

STM: ((Y FIVE Z) (GOAL MARK) (SAID FOUR) (GOAL COUNT) (GOAL *S* SEND
FOUR) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR) JUNK JU

```
NK JUNK JUNK JUNK JUNK JUNK)
P7


STM: ((Y FIVE Z) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM TWO) (SEND FOUR
) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM BEFORE)


STM: ((MANY BOTTOM BEFORE) (GOAL * ATTEND) (Y FIVE Z) (TOP TWELVE) (B
OTTOM TWO) (SEND FOUR) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK J
UNK)
P9


STM: ((GOAL * RECALL BOTTOM) (GOAL *S* ATTEND) (Y FIVE Z) (TOP TWELVE
) (BOTTOM TWO) (SEND FOUR) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JU
NK JUNK)
P10

********  (BOTTOM TWO)


STM: ((BOTTOM TWO) (GOAL * ATTEND) (Y FIVE Z) (TOP TWELVE) (SEND FOUR
) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)


STM: ((MANY TUBE) (GOAL * ATTEND) (BOTTOM TWO) (Y FIVE Z) (TOP TWELVE
) (SEND FOUR) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (BOTTOM TWO) (Y FIVE Z)
(TOP TWELVE) (SEND FOUR) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK
 JUNK)
P12

********  (SEND FOUR)


STM: ((SEND FOUR) (GOAL * ATTEND) (BOTTOM TWO) (Y FIVE Z) (TOP TWELVE
) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM NOW)


STM: ((MANY BOTTOM NOW) (GOAL * ATTEND) (SEND FOUR) (BOTTOM TWO) (Y F
IVE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK
)
```

```
P13


STM: ((Y ONE Z) (GOAL *S* ADD) (GOAL *S* COMPARE) (GOAL *S* MARK) (GO
AL *S* MOVE) (GOAL * SMALL) (GOAL *S* ATTEND) (SEND FOUR) (BOTTOM TWO
) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P14


STM: ((GOAL *S* SMALL) (Y ONE Z) (GOAL *S* ADD) (GOAL *S* COMPARE) (G
OAL *S* MARK) (GOAL * MOVE) (GOAL *S* ATTEND) (SEND FOUR) (BOTTOM TWO
) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P16


STM: ((GOAL *S* MOVE) (NUMBER ONE) (GOAL *S* SMALL) (GOAL *S* ADD) (G
OAL *S* COMPARE) (GOAL * MARK) (GOAL *S* ATTEND) (SEND FOUR) (BOTTOM
TWO) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P17


STM: ((Y TWO Z) (GOAL *S* MARK) (NUMBER ONE) (GOAL *S* MOVE) (GOAL *S
* SMALL) (GOAL *S* ADD) (GOAL * COMPARE) (GOAL *S* ATTEND) (SEND FOUR
) (BOTTOM TWO) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P210A


STM: ((GOAL *S* COMPARE) (GOAL *S* ADD) (Y TWO Z) (GOAL *S* MARK) (GO
AL * MOVE) (GOAL *S* SMALL) (GOAL *S* ATTEND) (SEND FOUR) (BOTTOM TWO
) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P16


STM: ((GOAL *S* MOVE) (NUMBER TWO) (GOAL *S* COMPARE) (GOAL *S* ADD)
(GOAL * MARK) (GOAL *S* SMALL) (GOAL *S* ATTEND) (SEND FOUR) (BOTTOM
TWO) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P17A


STM: ((Y THREE Z) (GOAL *S* MARK) (NUMBER TWO) (GOAL *S* MOVE) (GOAL
* COMPARE) (GOAL *S* ADD) (GOAL *S* SMALL) (GOAL *S* ATTEND) (SEND FO
UR) (BOTTOM TWO) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P130


STM: ((GOAL *S* COMPARE) (GOAL *S* LARGE) (BOTTOM TWO) (SMALLER TWO)
(Y THREE Z) (GOAL *S* MARK) (GOAL * MOVE) (GOAL *S* ADD) (GOAL *S* AT
TEND) (SEND FOUR) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16


STM: ((GOAL *S* MOVE) (NUMBER THREE) (GOAL *S* COMPARE) (GOAL *S* LAR
GE) (BOTTOM TWO) (SMALLER TWO) (GOAL * MARK) (GOAL *S* ADD) (GOAL *S*
 ATTEND) (SEND FOUR) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P17B
```

```
STM: ((Y FOUR Z) (GOAL *S* MARK) (NUMBER THREE) (GOAL *S* MOVE) (GOAL
 * COMPARE) (GOAL *S* LARGE) (BOTTOM TWO) (SMALLER TWO) (GOAL *S* ADD
) (GOAL *S* ATTEND) (SEND FOUR) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUN
K)
P210A


STM: ((GOAL *S* COMPARE) (GOAL *S* ADD) (Y FOUR Z) (GOAL *S* MARK) (G
OAL * MOVE) (GOAL *S* LARGE) (BOTTOM TWO) (SMALLER TWO) (GOAL *S* ATT
END) (SEND FOUR) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16


STM: ((GOAL *S* MOVE) (NUMBER FOUR) (GOAL *S* COMPARE) (GOAL *S* ADD)
 (GOAL * MARK) (GOAL *S* LARGE) (BOTTOM TWO) (SMALLER TWO) (GOAL *S*
ATTEND) (SEND FOUR) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P17C


STM: ((Y FIVE Z) (GOAL *S* MARK) (NUMBER FOUR) (GOAL *S* MOVE) (GOAL
* COMPARE) (GOAL *S* ADD) (GOAL *S* LARGE) (BOTTOM TWO) (SMALLER TWO)
 (GOAL *S* ATTEND) (SEND FOUR) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK
)
P210


STM: ((W ONE V) (GOAL *S* COMPARE) (SEND FOUR) (LARGER FOUR) (Y FIVE
Z) (GOAL *S* MARK) (GOAL *S* MOVE) (GOAL * ADD) (BOTTOM TWO) (SMALLER
 TWO) (GOAL *S* ATTEND) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK)
P230


STM: ((GOAL *S* ADD) (W TWO V) (GOAL *S* COMPARE) (SEND FOUR) (LARGER
 FOUR) (Y FIVE Z) (GOAL *S* MARK) (GOAL * MOVE) (BOTTOM TWO) (SMALLER
 TWO) (GOAL *S* ATTEND) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK)
P16


STM: ((GOAL *S* MOVE) (NUMBER FIVE) (GOAL *S* ADD) (W TWO V) (GOAL *S
* COMPARE) (SEND FOUR) (LARGER FOUR) (GOAL * MARK) (BOTTOM TWO) (SMAL
LER TWO) (GOAL *S* ATTEND) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK)
P170


STM: ((Y SIX Z) (GOAL *S* MARK) (NUMBER FIVE) (GOAL *S* MOVE) (GOAL *
S* ADD) (W TWO V) (GOAL * COMPARE) (SEND FOUR) (LARGER FOUR) (BOTTOM
TWO) (SMALLER TWO) (GOAL *S* ATTEND) (Y FIVE Z) (TOP TWELVE) JUNK JUN
K)
P240


STM: ((GOAL *S* COMPARE) (W TWO V) (Y SIX Z) (GOAL *S* MARK) (GOAL *S
* MOVE) (GOAL * ADD) (SEND FOUR) (LARGER FOUR) (BOTTOM TWO) (SMALLER
TWO) (GOAL *S* ATTEND) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK)
P220

********  SIX
```

```
STM: ((GOAL * PURGE) (Y SIX Z) (SEND FOUR) (BOTTOM TWO) (GOAL *S* ATT
END) (Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK
JUNK)
P24A


STM: ((GOAL * PURGE) (BOTTOM SIX) (OLD SEND FOUR) (GOAL *S* ATTEND) (
Y FIVE Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK J
UNK)
P24


STM: ((GOAL * PURGE) (BOTTOM SIX) (OLD SEND FOUR) (GOAL *S* ATTEND) (
TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((BOTTOM SIX) (OLD SEND FOUR) (GOAL * ATTEND) (TOP TWELVE) JUNK
JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*(SEND THREE)



STM: ((SEND THREE) (GOAL ATTEND) (BOTTOM SIX) (TOP TWELVE) JUNK JUNK
JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y ONE Z) (GOAL * SEND THREE) (GOAL *S* ATTEND) (TOP TWELVE) (B
OTTOM SIX) (SEND THREE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK
JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND THREE) (Y ONE Z) (GOAL *S*
ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND THREE) JUNK JUNK JUNK JUNK JU
NK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND THREE) (Y ONE Z) (GOAL *S
* ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND THREE) JUNK JUNK JUNK JUNK
JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND THREE) (Y ONE Z) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND THREE) JUNK JUNK JUNK JUNK J
UNK JUNK JUNK JUNK)
P5

*******  ONE
```

```
STM: ((GOAL * MARK) (GOAL COUNT) (SAID ONE) (GOAL *S* SEND THREE) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND THREE) JUNK JUNK JUNK
JUNK JUNK JUNK JUNK JUNK)
P6


STM: ((Y TWO Z) (GOAL MARK) (SAID ONE) (GOAL COUNT) (GOAL *S* SEND TH
REE) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND THREE) JUNK JU
NK JUNK JUNK JUNK JUNK JUNK)
P8


STM: ((GOAL * SEND THREE) (Y TWO Z) (GOAL *S* ATTEND) (TOP TWELVE) (B
OTTOM SIX) (SEND THREE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK
JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND THREE) (Y TWO Z) (GOAL *S*
ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND THREE) JUNK JUNK JUNK JUNK JU
NK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND THREE) (Y TWO Z) (GOAL *S
* ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND THREE) JUNK JUNK JUNK JUNK
JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (GS 1) (GOAL *S* SEND THREE) (Y TWO Z) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND THREE) JUNK JUNK JUNK JUNK J
UNK JUNK JUNK JUNK)
P5

********  TWO


STM: ((GOAL * MARK) (GOAL COUNT) (SAID TWO) (GOAL *S* SEND THREE) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND THREE) JUNK JUNK JUNK
JUNK JUNK JUNK JUNK JUNK)
P66


STM: ((Y THREE Z) (GOAL MARK) (SAID TWO) (GOAL COUNT) (GOAL *S* SEND
THREE) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND THREE) JUNK
JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SEND THREE) (Y THREE Z) (GOAL *S* ATTEND) (TOP TWELVE)
(BOTTOM SIX) (SEND THREE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUN
K JUNK)
P2
```

```
STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND THREE) (Y THREE Z) (GOAL *S
* ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND THREE) JUNK JUNK JUNK JUNK
JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUNIT) (ELM A) (GOAL *S* SEND THREE) (Y THREE Z) (GOAL
*S* ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND THREE) JUNK JUNK JUNK JUN
K JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND THREE) (Y THREE Z) (GOAL *
S* ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND THREE) JUNK JUNK JUNK JUNK
 JUNK JUNK JUNK JUNK)
P5

*******  THREE


STM: ((GOAL * MARK) (GOAL COUNT) (SAID THREE) (GOAL *S* SEND THREE) (
GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND THREE) JUNK JUNK JUN
K JUNK JUNK JUNK JUNK JUNK)
P6B


STM: ((Y FOUR Z) (GOAL MARK) (SAID THREE) (GOAL COUNT) (GOAL *S* SEND
 THREE) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND THREE) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK)
P7


STM: ((Y FOUR Z) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM SIX) (SEND THRE
E) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P8
INPUT TO STM:
*(MANY BOTTOM BEFORE)


STM: ((MANY BOTTOM BEFORE) (GOAL * ATTEND) (Y FOUR Z) (TOP TWELVE) (B
OTTOM SIX) (SEND THREE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK
JUNK)
P9


STM: ((GOAL * RECALL BOTTOM) (GOAL *S* ATTEND) (Y FOUR Z) (TOP TWELVE
) (BOTTOM SIX) (SEND THREE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK J
UNK JUNK)
P10

*******  (BOTTOM SIX)


STM: ((BOTTOM SIX) (GOAL * ATTEND) (Y FOUR Z) (TOP TWELVE) (SEND THRE
E) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9
```

```
INPUT TO STM:
*(MANY TUBE)


STM: ((MANY TUBE) (GOAL * ATTEND) (BOTTOM SIX) (Y FOUR Z) (TOP TWELVE
) (SEND THREE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (BOTTOM SIX) (Y FOUR Z)
(TOP TWELVE) (SEND THREE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUN
K JUNK)
P12

*******  (SEND THREE)


STM: ((SEND THREE) (GOAL * ATTEND) (BOTTOM SIX) (Y FOUR Z) (TOP TWELV
E) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM NOW)


STM: ((MANY BOTTOM NOW) (GOAL * ATTEND) (SEND THREE) (BOTTOM SIX) (Y
FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUN
K)
P13


STM: ((Y ONE Z) (GOAL *S* ADD) (GOAL *S* COMPARE) (GOAL *S* MARK) (GO
AL *S* MOVE) (GOAL * SMALL) (GOAL *S* ATTEND) (SEND THREE) (BOTTOM SI
X) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P14


STM: ((GOAL *S* SMALL) (Y ONE Z) (GOAL *S* ADD) (GOAL *S* COMPARE) (G
OAL *S* MARK) (GOAL * MOVE) (GOAL *S* ATTEND) (SEND THREE) (BOTTOM SI
X) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P16


STM: ((GOAL *S* MOVE) (NUMBER ONE) (GOAL *S* SMALL) (GOAL *S* ADD) (G
OAL *S* COMPARE) (GOAL * MARK) (GOAL *S* ATTEND) (SEND THREE) (BOTTOM
 SIX) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P17


STM: ((Y TWO Z) (GOAL *S* MARK) (NUMBER ONE) (GOAL *S* MOVE) (GOAL *S
* SMALL) (GOAL *S* ADD) (GOAL * COMPARE) (GOAL *S* ATTEND) (SEND THRE
E) (BOTTOM SIX) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P210A


STM: ((GOAL *S* COMPARE) (GOAL *S* ADD) (Y TWO Z) (GOAL *S* MARK) (GO
```

AL * MOVE) (GOAL *S* SMALL) (GOAL *S* ATTEND) (SEND THREE) (BOTTOM SI
X) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P16

STM: ((GOAL *S* MOVE) (NUMBER TWO) (GOAL *S* COMPARE) (GOAL *S* ADD)
(GOAL * MARK) (GOAL *S* SMALL) (GOAL *S* ATTEND) (SEND THREE) (BOTTOM
 SIX) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P17A

STM: ((Y THREE Z) (GOAL *S* MARK) (NUMBER TWO) (GOAL *S* MOVE) (GOAL
* COMPARE) (GOAL *S* ADD) (GOAL *S* SMALL) (GOAL *S* ATTEND) (SEND TH
REE) (BOTTOM SIX) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P210A

STM: ((GOAL *S* COMPARE) (GOAL *S* ADD) (Y THREE Z) (GOAL *S* MARK) (
GOAL * MOVE) (GOAL *S* SMALL) (GOAL *S* ATTEND) (SEND THREE) (BOTTOM
SIX) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P16

STM: ((GOAL *S* MOVE) (NUMBER THREE) (GOAL *S* COMPARE) (GOAL *S* ADD
) (GOAL * MARK) (GOAL *S* SMALL) (GOAL *S* ATTEND) (SEND THREE) (BOTT
OM SIX) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P17B

STM: ((Y FOUR Z) (GOAL *S* MARK) (NUMBER THREE) (GOAL *S* MOVE) (GOAL
 * COMPARE) (GOAL *S* ADD) (GOAL *S* SMALL) (GOAL *S* ATTEND) (SEND T
HREE) (BOTTOM SIX) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P190

STM: ((GOAL *S* COMPARE) (GOAL *S* LARGE) (SEND THREE) (SMALLER THREE
) (Y FOUR Z) (GOAL *S* MARK) (GOAL * MOVE) (GOAL *S* ADD) (GOAL *S* A
TTEND) (BOTTOM SIX) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16

STM: ((GOAL *S* MOVE) (NUMBER FOUR) (GOAL *S* COMPARE) (GOAL *S* LARG
E) (SEND THREE) (SMALLER THREE) (GOAL * MARK) (GOAL *S* ADD) (GOAL *S
* ATTEND) (BOTTOM SIX) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P17C

STM: ((Y FIVE Z) (GOAL *S* MARK) (NUMBER FOUR) (GOAL *S* MOVE) (GOAL
* COMPARE) (GOAL *S* LARGE) (SEND THREE) (SMALLER THREE) (GOAL *S* AD
D) (GOAL *S* ATTEND) (BOTTOM SIX) (Y FOUR Z) (TOP TWELVE) JUNK JUNK J
UNK)
P210A

STM: ((GOAL *S* COMPARE) (GOAL *S* ADD) (Y FIVE Z) (GOAL *S* MARK) (G
OAL * MOVE) (GOAL *S* LARGE) (SEND THREE) (SMALLER THREE) (GOAL *S* A
TTEND) (BOTTOM SIX) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)

P16

STM: ((GOAL *S* MOVE) (NUMBER FIVE) (GOAL *S* COMPARE) (GOAL *S* ADD) (GOAL * MARK) (GOAL *S* LARGE) (SEND THREE) (SMALLER THREE) (GOAL *S* ATTEND) (BOTTOM SIX) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P17D

STM: ((Y SIX Z) (GOAL *S* MARK) (NUMBER FIVE) (GOAL *S* MOVE) (GOAL * COMPARE) (GOAL *S* ADD) (GOAL *S* LARGE) (SEND THREE) (SMALLER THREE) (GOAL *S* ATTEND) (BOTTOM SIX) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK)
P210A

STM: ((GOAL *S* COMPARE) (GOAL *S* ADD) (Y SIX Z) (GOAL *S* MARK) (GOAL * MOVE) (GOAL *S* LARGE) (SEND THREE) (SMALLER THREE) (GOAL *S* ATTEND) (BOTTOM SIX) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16

STM: ((GOAL *S* MOVE) (NUMBER SIX) (GOAL *S* COMPARE) (GOAL *S* ADD) (GOAL * MARK) (GOAL *S* LARGE) (SEND THREE) (SMALLER THREE) (GOAL *S* ATTEND) (BOTTOM SIX) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P17E

STM: ((Y SEVEN Z) (GOAL *S* MARK) (NUMBER SIX) (GOAL *S* MOVE) (GOAL * COMPARE) (GOAL *S* ADD) (GOAL *S* LARGE) (SEND THREE) (SMALLER THREE) (GOAL *S* ATTEND) (BOTTOM SIX) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK)
P200

STM: ((W ONE V) (GOAL *S* COMPARE) (BOTTOM SIX) (LARGER SIX) (Y SEVEN Z) (GOAL *S* MARK) (GOAL *S* MOVE) (GOAL * ADD) (SEND THREE) (SMALLER THREE) (GOAL *S* ATTEND) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK)
P230

STM: ((GOAL *S* ADD) (W TWO V) (GOAL *S* COMPARE) (BOTTOM SIX) (LARGER SIX) (Y SEVEN Z) (GOAL *S* MARK) (GOAL * MOVE) (SEND THREE) (SMALLER THREE) (GOAL *S* ATTEND) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK)
P16

STM: ((GOAL *S* MOVE) (NUMBER SEVEN) (GOAL *S* ADD) (W TWO V) (GOAL * S* COMPARE) (BOTTOM SIX) (LARGER SIX) (GOAL * MARK) (SEND THREE) (SMALLER THREE) (GOAL *S* ATTEND) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK)
P17F

STM: ((Y EIGHT Z) (GOAL *S* MARK) (NUMBER SEVEN) (GOAL *S* MOVE) (GOAL *S* ADD) (W TWO V) (GOAL * COMPARE) (BOTTOM SIX) (LARGER SIX) (SEND THREE) (SMALLER THREE) (GOAL *S* ATTEND) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK)

P240

STM: ((GOAL *S* COMPARE) (W TWO V) (Y EIGHT Z) (GOAL *S* MARK) (GOAL *S* MOVE) (GOAL * ADD) (BOTTOM SIX) (LARGER SIX) (SEND THREE) (SMALLER THREE) (GOAL *S* ATTEND) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK)
P230A

STM: ((GOAL *S* ADD) (W THREE V) (GOAL *S* COMPARE) (Y EIGHT Z) (GOAL *S* MARK) (GOAL * MOVE) (BOTTOM SIX) (LARGER SIX) (SEND THREE) (SMALLER THREE) (GOAL *S* ATTEND) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK)
P16

STM: ((GOAL *S* MOVE) (NUMBER EIGHT) (GOAL *S* ADD) (W THREE V) (GOAL *S* COMPARE) (GOAL * MARK) (BOTTOM SIX) (LARGER SIX) (SEND THREE) (SMALLER THREE) (GOAL *S* ATTEND) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK)
P17G

STM: ((Y NINE Z) (GOAL *S* MARK) (NUMBER EIGHT) (GOAL *S* MOVE) (GOAL *S* ADD) (W THREE V) (GOAL * COMPARE) (BOTTOM SIX) (LARGER SIX) (SEND THREE) (SMALLER THREE) (GOAL *S* ATTEND) (Y FOUR Z) (TOP TWELVE) JUNK JUNK)
P240

STM: ((GOAL *S* COMPARE) (W THREE V) (Y NINE Z) (GOAL *S* MARK) (GOAL *S* MOVE) (GOAL * ADD) (BOTTOM SIX) (LARGER SIX) (SEND THREE) (SMALLER THREE) (GOAL *S* ATTEND) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK)
P220

******** NINE

STM: ((GOAL * PURGE) (Y NINE Z) (BOTTOM SIX) (SEND THREE) (GOAL *S* ATTEND) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P24A

STM: ((GOAL * PURGE) (BOTTOM NINE) (OLD SEND THREE) (GOAL *S* ATTEND) (Y FOUR Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P24

STM: ((GOAL * PURGE) (BOTTOM NINE) (OLD SEND THREE) (GOAL *S* ATTEND) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27

STM: ((BOTTOM NINE) (OLD SEND THREE) (GOAL * ATTEND) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P000

```
INPUT TO STM:
*(SEND ONE)


STM: ((SEND ONE) (GOAL ATTEND) (BOTTOM NINE) (TOP TWELVE) JUNK JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y ONE Z) (GOAL * SEND ONE) (GOAL *S* ATTEND) (TOP TWELVE) (BOT
TOM NINE) (SEND ONE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUN
K)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND ONE) (Y ONE Z) (GOAL *S* AT
TEND) (TOP TWELVE) (BOTTOM NINE) (SEND ONE) JUNK JUNK JUNK JUNK JUNK
JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND ONE) (Y ONE Z) (GOAL *S*
ATTEND) (TOP TWELVE) (BOTTOM NINE) (SEND ONE) JUNK JUNK JUNK JUNK JUN
K JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND ONE) (Y ONE Z) (GOAL *S* A
TTEND) (TOP TWELVE) (BOTTOM NINE) (SEND ONE) JUNK JUNK JUNK JUNK JUNK
 JUNK JUNK JUNK)
P5

*******  ONE


STM: ((GOAL * MARK) (GOAL COUNT) (SAID ONE) (GOAL *S* SEND ONE) (GOAL
 *S* ATTEND) (TOP TWELVE) (BOTTOM NINE) (SEND ONE) JUNK JUNK JUNK JUN
K JUNK JUNK JUNK JUNK)
P6


STM: ((Y TWO Z) (GOAL MARK) (SAID ONE) (GOAL COUNT) (GOAL *S* SEND ON
E) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM NINE) (SEND ONE) JUNK JUNK
JUNK JUNK JUNK JUNK JUNK)
P7


STM: ((Y TWO Z) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM NINE) (SEND ONE)
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P8
INPUT TO STM:
*(MANY BOTTOM BEFORE)


STM: ((MANY BOTTOM BEFORE) (GOAL * ATTEND) (Y TWO Z) (TOP TWELVE) (BO
```

```
TTOM NINE) (SEND ONE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JU
NK)
P9


STM: ((GOAL * RECALL BOTTOM) (GOAL *S* ATTEND) (Y TWO Z) (TOP TWELVE)
 (BOTTOM NINE) (SEND ONE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUN
K JUNK)
P10

*******  (BOTTOM NINE)


STM: ((BOTTOM NINE) (GOAL * ATTEND) (Y TWO Z) (TOP TWELVE) (SEND ONE)
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE()\)(\)


STM: ((MANY TUBE) (GOAL * ATTEND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE
) (SEND ONE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (BOTTOM NINE) (Y TWO Z)
(TOP TWELVE) (SEND ONE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK
JUNK)
P12

*******  (SEND ONE)


STM: ((SEND ONE) (GOAL * ATTEND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE)
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM NOW)


STM: ((MANY BOTTOM NOW) (GOAL * ATTEND) (SEND ONE) (BOTTOM NINE) (Y T
WO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((Y ONE Z) (GOAL *S* ADD) (GOAL *S* COMPARE) (GOAL *S* MARK) (GO
AL *S* MOVE) (GOAL * SMALL) (GOAL *S* ATTEND) (SEND ONE) (BOTTOM NINE
) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P14


STM: ((GOAL *S* SMALL) (Y ONE Z) (GOAL *S* ADD) (GOAL *S* COMPARE) (G
OAL *S* MARK) (GOAL * MOVE) (GOAL *S* ATTEND) (SEND ONE) (BOTTOM NINE
) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P15
```

STM: ((GOAL *S* MOVE) (NUMBER ONE) (GOAL *S* SMALL) (GOAL *S* ADD) (G
OAL *S* COMPARE) (GOAL * MARK) (GOAL *S* ATTEND) (SEND ONE) (BOTTOM N
INE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK)
P17

STM: ((Y TWO Z) (GOAL *S* MARK) (NUMBER ONE) (GOAL *S* MOVE) (GOAL *S
* SMALL) (GOAL *S* ADD) (GOAL * COMPARE) (GOAL *S* ATTEND) (SEND ONE)
 (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P190

STM: ((GOAL *S* COMPARE) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (Y
 TWO Z) (GOAL *S* MARK) (GOAL * MOVE) (GOAL *S* ADD) (GOAL *S* ATTEND
) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16

STM: ((GOAL *S* MOVE) (NUMBER TWO) (GOAL *S* COMPARE) (GOAL *S* LARGE
) (SEND ONE) (SMALLER ONE) (GOAL * MARK) (GOAL *S* ADD) (GOAL *S* ATT
END) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P17A

STM: ((Y THREE Z) (GOAL *S* MARK) (NUMBER TWO) (GOAL *S* MOVE) (GOAL
* COMPARE) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (GOAL *S* ADD) (
GOAL *S* ATTEND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK)
P210A

STM: ((GOAL *S* COMPARE) (GOAL *S* ADD) (Y THREE Z) (GOAL *S* MARK) (
GOAL * MOVE) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (GOAL *S* ATTE
ND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16

STM: ((GOAL *S* MOVE) (NUMBER THREE) (GOAL *S* COMPARE) (GOAL *S* ADD
) (GOAL * MARK) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (GOAL *S* A
TTEND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P17B

STM: ((Y FOUR Z) (GOAL *S* MARK) (NUMBER THREE) (GOAL *S* MOVE) (GOAL
 * COMPARE) (GOAL *S* ADD) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE)
(GOAL *S* ATTEND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK
)
P210A

STM: ((GOAL *S* COMPARE) (GOAL *S* ADD) (Y FOUR Z) (GOAL *S* MARK) (G
OAL * MOVE) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (GOAL *S* ATTEN
D) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16

STM: ((GOAL *S* MOVE) (NUMBER FOUR) (GOAL *S* COMPARE) (GOAL *S* ADD)
(GOAL * MARK) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (GOAL *S* AT
TEND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P17C

STM: ((Y FIVE Z) (GOAL *S* MARK) (NUMBER FOUR) (GOAL *S* MOVE) (GOAL
* COMPARE) (GOAL *S* ADD) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (
GOAL *S* ATTEND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK)
P210A

STM: ((GOAL *S* COMPARE) (GOAL *S* ADD) (Y FIVE Z) (GOAL *S* MARK) (G
OAL * MOVE) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (GOAL *S* ATTEN
D) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16

STM: ((GOAL *S* MOVE) (NUMBER FIVE) (GOAL *S* COMPARE) (GOAL *S* ADD)
(GOAL * MARK) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (GOAL *S* AT
TEND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P17D

STM: ((Y SIX Z) (GOAL *S* MARK) (NUMBER FIVE) (GOAL *S* MOVE) (GOAL *
COMPARE) (GOAL *S* ADD) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (G
OAL *S* ATTEND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK)
P210A

STM: ((GOAL *S* COMPARE) (GOAL *S* ADD) (Y SIX Z) (GOAL *S* MARK) (GO
AL * MOVE) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (GOAL *S* ATTEND
) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16

STM: ((GOAL *S* MOVE) (NUMBER SIX) (GOAL *S* COMPARE) (GOAL *S* ADD)
(GOAL * MARK) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (GOAL *S* ATT
END) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P17E

STM: ((Y SEVEN Z) (GOAL *S* MARK) (NUMBER SIX) (GOAL *S* MOVE) (GOAL
* COMPARE) (GOAL *S* ADD) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (
GOAL *S* ATTEND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK)
P210A

STM: ((GOAL *S* COMPARE) (GOAL *S* ADD) (Y SEVEN Z) (GOAL *S* MARK) (
GOAL * MOVE) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (GOAL *S* ATTE
ND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16

STM: ((GOAL *S* MOVE) (NUMBER SEVEN) (GOAL *S* COMPARE) (GOAL *S* ADD
) (GOAL * MARK) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (GOAL *S* A
TTEND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)

P17F

STM: ((Y EIGHT Z) (GOAL *S* MARK) (NUMBER SEVEN) (GOAL *S* MOVE) (GOA
L * COMPARE) (GOAL *S* ADD) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE)
 (GOAL *S* ATTEND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUN
K)
P210A

STM: ((GOAL *S* COMPARE) (GOAL *S* ADD) (Y EIGHT Z) (GOAL *S* MARK) (
GOAL * MOVE) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (GOAL *S* ATTE
ND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16

STM: ((GOAL *S* MOVE) (NUMBER EIGHT) (GOAL *S* COMPARE) (GOAL *S* ADD
) (GOAL * MARK) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (GOAL *S* A
TTEND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P17G

STM: ((Y NINE Z) (GOAL *S* MARK) (NUMBER EIGHT) (GOAL *S* MOVE) (GOAL
 * COMPARE) (GOAL *S* ADD) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE)
(GOAL *S* ATTEND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK
)
P210A

STM: ((GOAL *S* COMPARE) (GOAL *S* ADD) (Y NINE Z) (GOAL *S* MARK) (G
OAL * MOVE) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (GOAL *S* ATTEN
D) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P16

STM: ((GOAL *S* MOVE) (NUMBER NINE) (GOAL *S* COMPARE) (GOAL *S* ADD)
 (GOAL * MARK) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (GOAL *S* AT
TEND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK)
P17H

STM: ((Y TEN Z) (GOAL *S* MARK) (NUMBER NINE) (GOAL *S* MOVE) (GOAL *
 COMPARE) (GOAL *S* ADD) (GOAL *S* LARGE) (SEND ONE) (SMALLER ONE) (G
OAL *S* ATTEND) (BOTTOM NINE) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK)
P200

STM: ((W ONE V) (GOAL *S* COMPARE) (BOTTOM NINE) (LARGER NINE) (Y TEN
 Z) (GOAL *S* MARK) (GOAL *S* MOVE) (GOAL * ADD) (SEND ONE) (SMALLER
ONE) (GOAL *S* ATTEND) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK)
P220

******** TEN

STM: ((GOAL * PURGE) (Y TEN Z) (BOTTOM NINE) (SEND ONE) (GOAL *S* ATT
END) (Y TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)

```
UNK)
P24A


STM: ((GOAL * PURGE) (BOTTOM TEN) (OLD SEND ONE) (GOAL *S* ATTEND) (Y
 TWO Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUN
K)
P24


STM: ((GOAL * PURGE) (BOTTOM TEN) (OLD SEND ONE) (GOAL *S* ATTEND) (T
OP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((BOTTOM TEN) (OLD SEND ONE) (GOAL * ATTEND) (TOP TWELVE) JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*()



STM: (NIL (GOAL ATTEND) (BOTTOM TEN) (TOP TWELVE) JUNK JUNK JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)


END OF RUN

NUMBER OF PRODUCTIONS FIRED : 276

NUMBER OF OPERATIONS EXECUTED : 1161

NUMBER OF PRODUCTIONS TRIED : 5661

NUMBER OF SINGLE-ELEMENT COMPARISONS : 135141

SIMULATED TIME : 0

COMPUTER TIME: 652045 MSECS

NUMBER OF CONS-OPERATIONS : 1521701


*
*^C
```

ADDENDUM 4.

LISTING AND EXECUTION TRACE FOR N-SUB-NV: TABLE-LOOK-UP METHOD.

```
P 500
(VARIABLES;
(A GENC)
(B GENC)
(C GENC)
(D GENC)
(E GENC)
(<X> SEQC)
(<P> SEQC)
)
(PRODUCTIONS;
(P000 (GOAL * ATTEND)(OLD SEND C)
-==->
REPL((GOAL * ATTEND)  ;  (GOAL + ATTEND));
DEL((OLD SEND C));
USER()    )
(P00  (GOAL * ATTEND)(GO)
-==->
REPL((GOAL * ATTEND)  ;  (GOAL + ATTEND));
DEL((GO))   )
(P0 (GOAL * ATTEND)
-==->
USER()   )
(P1 (GOAL + ATTEND)(TOP A)(BOTTOM B)(SEND C)
-==->
REPL((GOAL + ATTEND)  ;  (GOAL *S* ATTEND));
INS((GOAL * SEND C));
INS((Y 1 Z))    )
(P2 (GOAL * SEND C)
-==->
REPL((GOAL * SEND C)  ;  (GOAL *S* SEND C));
INS((GOAL * PUSH BUTTON))   )
(P3 (GOAL * PUSH BUTTON)
-==->
DEL((GOAL * PUSH BUTTON));
INS((ELM A));
INS((GOAL * SUBIT))   )
(P4   (GOAL * SUBIT)(ELM A)
-==->
DEL((GOAL * SUBIT));
INS((QS 1));
DEL((ELM A));
INS((GOAL * COUNT))   )
(P5   (GOAL * COUNT)(QS 1)(Y <X> Z)
-==->
REPL((GOAL * COUNT)  ;  (GOAL + COUNT));
DEL((QS 1));
SAY(<X>);
REPL((Y <X> Z)  ;  (SAID <X>));
INS((GOAL * MARK))   )
(P6 (GOAL * MARK)(SAID 1)
-==->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y 2 Z))    )
(P66  (GOAL * MARK)(SAID 2)
-==->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y 3 Z))    )
(P6B  (GOAL * MARK)(SAID 3)
-==->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y 4 Z))    )
(P66  (GOAL * MARK)(SAID 4)
-==->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS(Y 5 Z))   )
```

```
(P7 (GOAL + MARK)(GOAL + COUNT)(GOAL *S* SEND C)
(SAID C)
==>
DEL((GOAL *S* SEND C));
DEL((GOAL + MARK));
DEL((GOAL + COUNT));
DEL((SAID C));
REPL((GOAL *S* ATTEND)  ;  (GOAL * ATTEND))  )
(P8  (GOAL + MARK)(GOAL + COUNT)(GOAL *S* SEND C)(SAID D)
==>
REPL((GOAL *S* SEND C)  ;  (GOAL * SEND C));
DEL((GOAL + MARK));
DEL((GOAL + COUNT));
DEL((SAID D))     )
(P9 (GOAL * ATTEND)(MANY TOP BEFORE)
==>
REPL((GOAL * ATTEND)  ;  (GOAL *S* ATTEND));
DEL((MANY TOP BEFORE));
INS((GOAL * RECALL TOP))    )
(P10  (GOAL * RECALL TOP)(TOP A)
==>
SAY((TOP A));
DEL((GOAL * RECALL TOP));
REPL((GOAL *S* ATTEND)  ;  (GOAL * ATTEND))  )
(P11 (GOAL * ATTEND)(MANY TUBE)
==>
REPL((GOAL * ATTEND)  ;  (GOAL *S* ATTEND));
DEL((MANY TUBE));
INS((GOAL * RECALL  TUBE))    )
(P12  (GOAL * RECALL TUBE)(SEND C)
==>
DEL((GOAL * RECALL TUBE));
REPL((GOAL *S* ATTEND)  ;  (GOAL * ATTEND));
SAY((SEND C))     )
(P13 (GOAL * ATTEND)(MANY TOP NOW)
==>
REPL((GOAL * ATTEND)  ;  (GOAL *S* ATTEND));
DEL((MANY TOP NOW));
INS((GOAL * SUBTRACT))      )
(P14 (GOAL * SUBTRACT)(TOP A)(ABS (GOAL * LIST A))
==>
REPL((GOAL * SUBTRACT)  ;  (GOAL *S* SUBTRACT));
INS((GOAL * LIST A))  )
(P15 (GOAL * LIST THREE)
==>
REPL((GOAL *S* SUBTRACT)  ;  (GOAL * SUBTRACT));
INS((1 TWO Z));
INS((2 ONE Z))    )
(P15A  (GOAL * LIST SIX)
==>
REPL((GOAL *S* SUBTRACT)  ;  (GOAL * SUBTRACT));
INS((2 FOUR Z));
INS((3 THREE Z))    )
(P15B (GOAL * LIST TEN)
==>
REPL((GOAL *S* SUBTRACT)  ;  (GOAL * SUBTRACT));
INS((3 SEVEN Z));
INS((4 SIX Z))    )
(P15C  (GOAL * LIST TWELVE)
==>
REPL((GOAL *S* SUBTRACT)  ;  (GOAL * SUBTRACT));
INS((1 ELEVEN Z));
INS((2 TEN Z))     )
(P16   (GOAL * SUBTRACT)(SEND C)(C <X> Z)(TOP A)
==>
SAY(<X>));
```

```
DEL((GOAL * SUBTRACT));
DEL((GOAL * LIST A));
INS((GOAL * PURGE))    )
(P24A (GOAL * PURGE)(SEND C)(C <X> Z)(TOP A)
==>
REPL((TOP A)  ;  (TOP <X>));
REPL((SEND C)  ;  (OLD SEND C));
DEL((C <X> Z))    )
(P24 (GOAL * PURGE)(Y <X> Z)
==>
DEL((Y <X> Z))    )
(P25  (GOAL * PURGE)(C <P> Z)
==>
DEL((C <P> Z))    )
(P26 ((GOAL * PURGE)(BOTTOM 12)
==>
STOP)
(P27 (GOAL * PURGE)
==>
DEL((GOAL * PURGE));
REPL((GOAL $S* ATTEND)  ;  (GOAL * ATTEND))    )
)
(PSLISTS:
(N-SUB-N1
P9 P11 P13
P000 P00 P0
P1 P2 P3 P4 P5 P6
P6A P6B P6C
P7 P8 P10 P12
P13 P14 P16 P15
P15A P15B P15C
P24A
P24 P25 P26 P27
)
)
(DEF! STM((GOAL * ATTEND)(TOP TWELVE)(BOTTOM ZERO)(SEND 2)))
**END
*FILE
[EDIFIL filed : DSKC:KVADD5.PSS]
```

```
RUN BEGINS :

((GOAL * ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK J
UNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(GO)


STM: ((GO) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK)
P00


STM: ((GOAL ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUN
K JUNK JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y 1 Z) (GOAL * SEND 2) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 2) (Y 1 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 2) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 2) (Y 1 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P5

*******  1


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 2) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK
)
P6


STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 2) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUN
K)
P8
```

```
STM: ((GOAL * SEND 2) (Y 2 Z) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 2) (Y 2 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (FLN A) (GOAL *S* SEND 2) (Y 2 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (QS 1) (GOAL *S* SEND 2) (Y 2 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P5

******** 2


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 2) (GOAL *S* SEND 2) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK
)
P6A


STM: ((Y 3 Z) (GOAL MARK) (SAID 2) (GOAL COUNT) (GOAL *S* SEND 2) (GOA
L *S* ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK
K)
P7


STM: ((Y 3 Z) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P8
INPUT TO STM:
*((MANY TOP BEFORE)


STM: ((MANY TOP BEFORE) (GOAL * ATTEND) (Y 3 Z) (TOP TWELVE) (BOTTOM
ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9


STM: ((GOAL * RECALL (OP) (GOAL *S* ATTEND) (Y 3 Z) (TOP TWELVE) (BOT
TOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

********  (TOP TWELVE)


STM: ((TOP TWELVE) (GOAL * ATTEND) (Y 3 Z) (BOTTOM ZERO) (SEND 2) JUN
```

```
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)


STM: ((MANY TUBE) (GOAL * ATTEND) (TOP TWELVE) (Y 3 Z) (BOTTOM ZERO)
(SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (TOP TWELVE) (Y 3 Z) (BO
TTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

******* (SEND 2)


STM: ((SEND 2) (GOAL * ATTEND) (TOP TWELVE) (Y 3 Z) (BOTTOM ZERO) JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TOP NOW)


STM: ((MANY TOP NOW) (GOAL * ATTEND) (SEND 2) (TOP TWELVE) (Y 3 Z) (B
OTTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((GOAL * SUBTRACT) (GOAL *S* ATTEND) (SEND 2) (TOP TWELVE) (Y 3
Z) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P14


STM: ((GOAL * LIST TWELVE) (GOAL *S* SUBTRACT) (TOP TWELVE) (GOAL *S*
 ATTEND) (SEND 2) (Y 3 Z) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK
)
P15C


STM: ((2 TEN Z) (1 ELEVEN Z) (GOAL * LIST TWELVE) (GOAL * SUBTRACT) (
TOP TWELVE) (GOAL *S* ATTEND) (SEND 2) (Y 3 Z) (BOTTOM ZERO) JUNK JUN
K JUNK JUNK)
P16

******* TEN


STM: ((GOAL * PURGE) (SEND 2) (2 TEN Z) (TOP TWELVE) (1 ELEVEN Z) (GO
AL *S* ATTEND) (Y 3 Z) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK)
P24A


STM: ((GOAL * PURGE) (OLD CLOD 2) (TOP TEN) (1 ELEVEN Z) (GOAL *S* AT
```

```
TEND) (Y 3 Z) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (OLD SEND 2) (TOP TEN) (1 ELEVEN Z) (GOAL *S* AT
TEND) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (OLD SEND 2) (TOP TEN) (GOAL *S* ATTEND) (BOTTOM
 ZERO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((OLD SEND 2) (TOP TEN) (GOAL * ATTEND) (BOTTOM ZERO) JUNK JUNK
JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*((SEND 4)



STM: ((SEND 4) (GOAL ATTEND) (TOP TEN) (BOTTOM ZERO) JUNK JUNK JUNK J
UNK JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y 1 Z) (GOAL * SEND 4) (GOAL *S* ATTEND) (TOP TEN) (BOTTOM ZER
O) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 1 Z) (GOAL *S* ATTEND
) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 4) (Y 1 Z) (GOAL *S* ATTEN
D) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

******** 1


STM: ((GOAL * RANK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P6


STM: ((Y 2 Z) (GOAL RANK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK)
```

```
P8


STM: ((GOAL * SEND 4) (Y 2 Z) (GOAL *S* ATTEND) (TOP TEN) (BOTTOM ZER
O) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 2 Z) (GOAL *S* ATTEND
) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y 2 Z) (GOAL *S* ATTE
ND) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 4) (Y 2 Z) (GOAL *S* ATTEN
D) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

******* 2


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 2) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P6A


STM: ((Y 3 Z) (GOAL MARK) (SAID 2) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK)
P8


STM: ((GOAL * SEND 4) (Y 3 Z) (GOAL *S* ATTEND) (TOP TEN) (BOTTOM ZER
O) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 3 Z) (GOAL *S* ATTEND
) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y 3 Z) (GOAL *S* ATTE
ND) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 4) (Y 3 Z) (GOAL *S* ATTEN
D) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

******* 3
```

```
STM: ((GOAL * MARK) (GOAL COUNT) (SAID 3) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P6B


STM: ((Y 4 Z) (GOAL MARK) (SAID 3) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK)
P8


STM: ((GOAL * SEND 4) (Y 4 Z) (GOAL *S* ATTEND) (TOP TEN) (BOTTOM ZER
O) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 4 Z) (GOAL *S* ATTEND
) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y 4 Z) (GOAL *S* ATTE
ND) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (QS 1) (GOAL *S* SEND 4) (Y 4 Z) (GOAL *S* ATTEN
D) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

******* 4


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 4) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P6C


STM: ((Y 5 Z) (GOAL MARK) (SAID 4) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK JUNK JUNK JUNK)
P7


STM: ((Y 5 Z) (GOAL * ATTEND) (TOP TEN) (BOTTOM ZERO) (SEND 4) JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
#(MANY TOP BEFORE)


STM: ((MANY TOP BEFORE) (GOAL * ATTEND) (Y 5 Z) (TOP TEN) (BOTTOM ZER
O) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9
```

```
STM: ((GOAL * RECALL TOP) (GOAL *S* ATTEND) (Y 5 Z) (TOP TEN) (BOTTOM
 ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

********  (TOP TEN)


STM: ((TOP TEN) (GOAL * ATTEND) (Y 5 Z) (BOTTOM ZERO) (SEND 4) JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)



STM: ((MANY TUBE) (GOAL * ATTEND) (TOP TEN) (Y 5 Z) (BOTTOM ZERO) (SE
ND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (TOP TEN) (Y 5 Z) (BOTTO
M ZERO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

********  (SEND 4)


STM: ((SEND 4) (GOAL * ATTEND) (TOP TEN) (Y 5 Z) (BOTTOM ZERO) JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TOP NOW)



STM: ((MANY TOP NOW) (GOAL * ATTEND) (SEND 4) (TOP TEN) (Y 5 Z) (BOTT
OM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((GOAL * SUBTRACT) (GOAL *S* ATTEND) (SEND 4) (TOP TEN) (Y 5 Z)
(BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P14


STM: ((GOAL * LIST TEN) (GOAL *S* SUBTRACT) (TOP TEN) (GOAL *S* ATTEN
D) (SEND 4) (Y 5 Z) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK)
P15B


STM: ((4 SIX Z) (3 SEVEN Z) (GOAL * LIST TEN) (GOAL * SUBTRACT) (TOP
TEN) (GOAL *S* ATTEND) (SEND 4) (Y 5 Z) (BOTTOM ZERO) JUNK JUNK JUNK
JUNK)
P16

********  SIX
```

```
STM: ((GOAL * PURGE) (SEND 4) (4 SIX Z) (TOP TEN) (3 SEVEN Z) (GOAL *
S* ATTEND) (Y 5 Z) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK)
P24A


STM: ((GOAL * PURGE) (OLD SEND 4) (TOP SIX) (3 SEVEN Z) (GOAL *S* ATT
END) (Y 5 Z) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (OLD SEND 4) (TOP SIX) (3 SEVEN Z) (GOAL *S* ATT
END) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (OLD SEND 4) (TOP SIX) (GOAL *S* ATTEND) (BOTTON
 ZERO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((OLD SEND 4) (TOP SIX) (GOAL * ATTEND) (BOTTOM ZERO) JUNK JUNK
JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*(SEND 3)


STM: ((SEND 3) (GOAL ATTEND) (TOP SIX) (BOTTOM ZERO) JUNK JUNK JUNK J
UNK JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y 1 Z) (GOAL * SEND 3) (GOAL *S* ATTEND) (TOP SIX) (BOTTON ZER
O) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 3) (Y 1 Z) (GOAL *S* ATTEND
) (TOP SIX) (BOTTOM ZERO) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 3) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP SIX) (BOTTON ZERO) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 3) (Y 1 Z) (GOAL *S* ATTEN
D) (TOP SIX) (BOTTON ZERO) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P5

******** 1
```

```
STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 3) (GOAL *S*
 ATTEND) (TOP SIX) (BOTTOM ZERO) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P6


STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 3) (GO
AL *S* ATTEND) (TOP SIX) (BOTTOM ZERO) (SEND 3) JUNK JUNK JUNK JUNK)
P8


STM: ((GOAL * SEND 3) (Y 2 Z) (GOAL *S* ATTEND) (TOP SIX) (BOTTOM ZER
O) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 3) (Y 2 Z) (GOAL *S* ATTEND
) (TOP SIX) (BOTTOM ZERO) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 3) (Y 2 Z) (GOAL *S* ATTE
ND) (TOP SIX) (BOTTOM ZERO) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 3) (Y 2 Z) (GOAL *S* ATTEN
D) (TOP SIX) (BOTTOM ZERO) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P5

******* 2


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 2) (GOAL *S* SEND 3) (GOAL *S*
 ATTEND) (TOP SIX) (BOTTOM ZERO) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P6A


STM: ((Y 3 Z) (GOAL MARK) (SAID 2) (GOAL COUNT) (GOAL *S* SEND 3) (GO
AL *S* ATTEND) (TOP SIX) (BOTTOM ZERO) (SEND 3) JUNK JUNK JUNK JUNK)
P8


STM: ((GOAL * SEND 3) (Y 3 Z) (GOAL *S* ATTEND) (TOP SIX) (BOTTOM ZER
O) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 3) (Y 3 Z) (GOAL *S* ATTEND
) (TOP SIX) (BOTTOM ZERO) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 3) (Y 3 Z) (GOAL *S* ATTE
ND) (TOP SIX) (BOTTOM ZERO) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P1
```

```
STM: ((GOAL * COUNT) (QS 1) (GOAL *S* SEND 3) (Y 3 Z) (GOAL *S* ATTEN
D) (TOP SIX) (BOTTOM ZERO) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P5

*******  3


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 3) (GOAL *S* SEND 3) (GOAL *S*
 ATTEND) (TOP SIX) (BOTTOM ZERO) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P6B


STM: ((Y 4 Z) (GOAL MARK) (SAID 3) (GOAL COUNT) (GOAL *S* SEND 3) (GO
AL *S* ATTEND) (TOP SIX) (BOTTOM ZERO) (SEND 3) JUNK JUNK JUNK JUNK)
P7


STM: ((Y 4 Z) (GOAL * ATTEND) (TOP SIX) (BOTTOM ZERO) (SEND 3) JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TOP BEFORE)


STM: ((MANY TOP BEFORE) (GOAL * ATTEND) (Y 4 Z) (TOP SIX) (BOTTOM ZER
O) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9


STM: ((GOAL * RECALL TOP) (GOAL *S* ATTEND) (Y 4 Z) (TOP SIX) (BOTTOM
 ZERO) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

*******  (TOP SIX)


STM: ((TOP SIX) (GOAL * ATTEND) (Y 4 Z) (BOTTOM ZERO) (SEND 3) JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)


STM: ((MANY TUBE) (GOAL * ATTEND) (TOP SIX) (Y 4 Z) (BOTTOM ZERO) (SE
ND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (TOP SIX) (Y 4 Z) (BOTTO
M ZERO) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

*******  (SAID 3)
```

```
STM: ((SEND 3) (GOAL * ATTEND) (TOP SIX) (Y 4 Z) (BOTTOM ZERO) JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TOP NOW)


STM: ((MANY TOP NOW) (GOAL * ATTEND) (SEND 3) (TOP SIX) (Y 4 Z) (BOTT
OM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((GOAL * SUBTRACT) (GOAL *S* ATTEND) (SEND 3) (TOP SIX) (Y 4 Z)
(BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P14


STM: ((GOAL * LIST SIX) (GOAL *S* SUBTRACT) (TOP SIX) (GOAL *S* ATTEN
D) (SEND 3) (Y 4 Z) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK)
P15A


STM: ((3 THREE Z) (2 FOUR Z) (GOAL * LIST SIX) (GOAL * SUBTRACT) (TOP
 SIX) (GOAL *S* ATTEND) (SEND 3) (Y 4 Z) (BOTTOM ZERO) JUNK JUNK JUNK
 JUNK)
P16

*******  THREE


STM: ((GOAL * PURGE) (SEND 3) (3 THREE Z) (TOP SIX) (2 FOUR Z) (GOAL
*S* ATTEND) (Y 4 Z) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK)
P24A


STM: ((GOAL * PURGE) (OLD SEND 3) (TOP THREE) (2 FOUR Z) (GOAL *S* AT
TEND) (Y 4 Z) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (OLD SEND 3) (TOP THREE) (2 FOUR Z) (GOAL *S* AT
TEND) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (OLD SEND 3) (TOP THREE) (GOAL *S* ATTEND) (BOTT
OM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((OLD SEND 3) (TOP THREE) (GOAL * ATTEND) (BOTTOM ZERO) JUNK JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P400
```

```
INPUT TO STM:
*(SEND 1)


STM: ((SEND 1) (GOAL ATTEND) (TOP THREE) (BOTTOM ZERO) JUNK JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y 1 Z) (GOAL * SEND 1) (GOAL *S* ATTEND) (TOP THREE) (BOTTOM Z
ERO) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 1) (Y 1 Z) (GOAL *S* ATTEND
) (TOP THREE) (BOTTOM ZERO) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 1) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP THREE) (BOTTOM ZERO) (SEND 1) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 1) (Y 1 Z) (GOAL *S* ATTEN
D) (TOP THREE) (BOTTOM ZERO) (SEND 1) JUNK JUNK JUNK JUNK JUNK)
P5

******** 1


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 1) (GOAL *S*
 ATTEND) (TOP THREE) (BOTTOM ZERO) (SEND 1) JUNK JUNK JUNK JUNK JUNK)
P6


STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 1) (GO
AL *S* ATTEND) (TOP THREE) (BOTTOM ZERO) (SEND 1) JUNK JUNK JUNK JUNK
)
P7


STM: ((Y 2 Z) (GOAL * ATTEND) (TOP THREE) (BOTTOM ZERO) (SEND 1) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*((MANY TOP BEFORE)


STM: ((MANY TOP BEFORE) (GOAL * ATTEND) (Y 2 Z) (TOP THREE) (BOTTOM Z
ERO) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9
```

```
STM: ((GOAL * RECALL TOP) (GOAL *S* ATTEND) (Y 2 Z) (TOP THREE) (BOTT
OM ZERO) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

********  (TOP THREE)


STM: ((TOP THREE) (GOAL * ATTEND) (Y 2 Z) (BOTTOM ZERO) (SEND 1) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)


STM: ((MANY TUBE) (GOAL * ATTEND) (TOP THREE) (Y 2 Z) (BOTTOM ZERO) (
SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (TOP THREE) (Y 2 Z) (BOT
TOM ZERO) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

********  (SEND 1)


STM: ((SEND 1) (GOAL * ATTEND) (TOP THREE) (Y 2 Z) (BOTTOM ZERO) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TOP NOW)


STM: ((MANY TOP NOW) (GOAL * ATTEND) (SEND 1) (TOP THREE) (Y 2 Z) (BO
TTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((GOAL * SUBTRACT) (GOAL *S* ATTEND) (SEND 1) (TOP THREE) (Y 2 Z
) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P14


STM: ((GOAL * LIST THREE) (GOAL *S* SUBTRACT) (TOP THREE) (GOAL *S* A
TTEND) (SEND 1) (Y 2 Z) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((2 ONE Z) (1 TWO Z) (GOAL * LIST THREE) (GOAL * SUBTRACT) (TOP
THREE) (GOAL *S* ATTEND) (SEND 1) (Y 2 Z) (BOTTOM ZERO) JUNK JUNK JUN
K JUNK)
P16

********  TWO
```

```
STM: ((GOAL * PURGE) (SEND 1) (1 TWO Z) (TOP THREE) (2 ONE Z) (GOAL *
S* ATTEND) (Y 2 Z) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK)
P24A


STM: ((GOAL * PURGE) (OLD SEND 1) (TOP TWO) (2 ONE Z) (GOAL *S* ATTEN
D) (Y 2 Z) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (OLD SEND 1) (TOP TWO) (2 ONE Z) (GOAL *S* ATTEN
D) (BOTTOM ZERO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (OLD SEND 1) (TOP TWO) (GOAL *S* ATTEND) (BOTTOM
 ZERO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((OLD SEND 1) (TOP TWO) (GOAL * ATTEND) (BOTTOM ZERO) JUNK JUNK
JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*()


STM: (NIL (GOAL ATTEND) (TOP TWO) (BOTTOM ZERO) JUNK JUNK JUNK JUNK J
UNK JUNK JUNK JUNK JUNK)


END OF RUN

NUMBER OF PRODUCTIONS FIRED : 202

NUMBER OF OPERATIONS EXECUTED : 557

NUMBER OF PRODUCTIONS TRIED : 3310

NUMBER OF SINGLE-ELEMENT COMPARISONS : 52047

SIMULATED TIME : 0

COMPUTER TIME: 274100 MSECS

NUMBER OF CONS-OPERATIONS : 516576


*
*^C
```

ADDENDUM 5.

LISTING AND EXECUTION TRACE FOR N-SUB-NV: COUNTING-BASED METHOD.

```
 P SOO
(VARIABLES;
(A GENC)
(B GENC)
(C GENC)
(D GENC)
(E GENC)
(<X> GENC)
(<F> SEQC)
)
(PRODUCTIONS;
(P000 (GOAL * ATTEND)(OLD SEND C)
-->
REPL((GOAL * ATTEND)  ;  (GOAL + ATTEND));
DEL((OLD SEND C));
USER()    )
(P00  (GOAL * ATTEND)(GO)
-->
REPL((GOAL * ATTEND)  ;  (GOAL + ATTEND));
DEL((GO))   )
(P0 (GOAL * ATTEND)
-->
USER()   )
(P1 (GOAL + ATTEND)(TOP A)(BOTTOM B)(SEND C)
-->
REPL((GOAL + ATTEND)  ;  (GOAL *S* ATTEND));
INS((GOAL * SEND C));
INS((Y 1 Z))    )
(P2 (GOAL * SEND C)
-->
REPL((GOAL * SEND C)  ;  (GOAL *S* SEND C));
INS((GOAL * PUSH BUTTON))   )
(P3 (GOAL * PUSH BUTTON)
-->
DEL((GOAL * PUSH BUTTON));
INS((ELM A));
INS((GOAL * SUBIT))   )
(P4  (GOAL * SUBIT)(ELM A)
-->
DEL((GOAL * SUBIT));
INS((QS 1));
DEL((ELM A));
INS((GOAL * COUNT))   )
(P5  (GOAL * COUNT)(QS 1)(Y <X> Z)
-->
REPL((GOAL * COUNT)  ;  (GOAL + COUNT));
DEL((QS 1));
SAY(<X>);
REPL((Y <X> Z)  ;  (SAID <X>));
INS((GOAL * MARK))   )
(P6 (GOAL * MARK)(SAID 1)
-->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y 2 3))    )
(P6A  (GOAL * MARK)(SAID 2)
-->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y 3 2))    )
(P6B  (GOAL * MARK)(SAID 3)
-->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y 4 5))    )
(P6C  (GOAL * MARK)(SAID 4)
-->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INSC(Y 5 Z))   )
```

```
(P7 (GOAL * MARK)(GOAL * COUNT)(GOAL *S* SEND C)
(SAID C)
==>
DEL((GOAL *S* SEND C));
DEL((GOAL * MARK));
DEL((GOAL * COUNT));
DEL((SAID C));
REPL((GOAL *S* ATTEND)  ;  (GOAL * ATTEND))  )
(P8  (GOAL * MARK)(GOAL * COUNT)(GOAL *S* SEND C)(SAID D)
==>
REPL((GOAL *S* SEND C)  ;  (GOAL * SEND C));
DEL((GOAL * MARK));
DEL((GOAL * COUNT));
DEL((SAID D))     )
(P9 (GOAL * ATTEND)(MANY TOP BEFORE)
==>
REPL((GOAL * ATTEND)  ;  (GOAL *S* ATTEND));
DEL((MANY TOP BEFORE));
INS((GOAL * RECALL TOP))    )
(P10  (GOAL * RECALL TOP)(TOP A)
==>
SAY((TOP A));
DEL((GOAL * RECALL TOP));
REPL((GOAL *S* ATTEND)  ;  (GOAL * ATTEND))  )
(P11 (GOAL * ATTEND)(MANY TUBE)
==>
REPL((GOAL * ATTEND)  ;  (GOAL *S* ATTEND));
DEL((MANY TUBE));
INS((GOAL * RECALL  TUBE))    )
(P12  (GOAL * RECALL TUBE)(SEND C)
==>
DEL((GOAL * RECALL TUBE));
REPL((GOAL *S* ATTEND)  ;  (GOAL * ATTEND));
SAY((SEND C))     )
(P13 (GOAL * ATTEND)(MANY TOP NOW)(TOP A)
==>
REPL((GOAL * ATTEND)  ;  (GOAL *S* ATTEND));
DEL((MANY TOP NOW));
INS((GOAL * SUBTRACT));
INS((U A V));
INS((Y A Z))     )
(P14 (GOAL * SUBTRACT)(W <P> V)(SEND <P>)(Y <X> Z)
==>
SAY(<X>);
DEL((GOAL * SUBTRACT));
INS((GOAL * PURGE))    )
(P15 (GOAL * SUBTRACT)(W <P> V)(Y <X> Z)
==>
REPL((GOAL * SUBTRACT)  ;  (GOAL *S* SUBTRACT));
INS((GOAL * NEXT DOWN));
DO(STEPDOWN)  )
(P16 (GOAL * NEXT UP)
==>
DO(STEPUP)  )
(P17 (GOAL * NEXT DOWN)(Y TWELVE Z)
==>
REPL((Y TWELVE Z)  ;  (Y ELEVEN Z));
DEL((GOAL * NEXT DOWN));
INS((GOAL * NEXT UP))  )
(P17A (GOAL * NEXT DOWN)(Y ELEVEN Z)
==>
REPL((Y ELEVEN Z)  ;  (Y TEN Z));
DEL((GOAL * NEXT DOWN));
INS((GOAL * NEXT UP))    )
(P17B (GOAL * NEXT DOWN)(Y TEN Z)
==>
```

```
REPL((Y TEN Z)  ;  (Y NINE Z))$
DEL((GOAL * NEXT DOWN))$
INS((GOAL * NEXT UP))    )
(P17C (GOAL * NEXT DOWN)(Y NINE Z)
-->
REPL((Y NINE Z)  ;  (Y EIGHT Z))$
DEL((GOAL * NEXT DOWN))$
INS((GOAL * NEXT UP))    )
(P17D (GOAL * NEXT DOWN)(Y EIGHT Z)
-->
REPL((Y EIGHT Z)  ;  (Y SEVEN Z))$
DEL((GOAL * NEXT DOWN))$
INS((GOAL * NEXT UP))    )
(P17E (GOAL * NEXT DOWN)(Y SEVEN Z)
-->
REPL((Y SEVEN Z)  ;  (Y SIX Z))$
DEL((GOAL * NEXT DOWN))$
INS((GOAL * NEXT UP))    )
(P17F (GOAL * NEXT DOWN)(Y SIX Z)
-->
REPL((Y SIX Z)  ;  (Y FIVE Z))$
DEL((GOAL * NEXT DOWN))$
INS((GOAL * NEXT UP))    )
(P17G (GOAL * NEXT DOWN)(Y FIVE Z)
-->
REPL((Y FIVE Z)  ;  (Y FOUR Z))$
DEL((GOAL * NEXT DOWN))$
INS((GOAL * NEXT UP))    )
(P17H  (GOAL * NEXT DOWN)(Y FOUR Z)
-->
REPL((Y FOUR Z)  ;  (Y THREE Z))$
DEL((GOAL * NEXT DOWN))$
INS((GOAL * NEXT UP))    )
(P17I (GOAL * NEXT DOWN)(Y THREE Z)
-->
REPL((Y THREE Z)  ;  (Y TWO Z))$
DEL((GOAL * NEXT DOWN))$
INS((GOAL * NEXT UP))    )
(P18 (GOAL * NEXT UP)(W 0 V)
-->
REPL((W 0 V)  ;  (W 1 V))$
DEL((GOAL * NEXT UP))$
REPL((GOAL *S* SUBTRACT)  ;  (GOAL * SUBTRACT))    )
(P19A (GOAL * NEXT UP)(W 1 V)
-->
REPL((W 1 V)  ;  (W 2 V))$
DEL((GOAL * NEXT UP))$
REPL((GOAL *S* SUBTRACT)  ;  (GOAL * SUBTRACT))    )
(P19B (GOAL * NEXT UP)(W 2 V)
-->
REPL((W 2 V)  ;  (W 3 V))$
DEL((GOAL * NEXT UP))$
REPL((GOAL *S* SUBTRACT)  ;  (GOAL * SUBTRACT))    )
(P19C (GOAL * NEXT UP)(W 3 V)
-->
REPL((W 4 V)  ;  (W 4 V))$
DEL((GOAL * NEXT UP))$
REPL((GOAL *S* SUBTRACT)  ;  (GOAL * SUBTRACT))    )
(P24A  (GOAL * PURGE)(Y <X> Z)(TOP A)(SEND C)
-->
REPL((TOP A)  ;  (TOP <X>))$
REPL((SEND C)  ;  (OLD SEND C))$
DEL((Y <X> Z))    )
(P24 (GOAL * PURGE)(Y <X> Z)
-->
DEL((Y <X> Z))    )
```

```
(P25 (GOAL * PURGE)(W <P> V)
===>
DEL((W <P> V))    )
(P26 (GOAL * PURGE)(BOTTOM 12)
===>
STOP)
(P27 (GOAL * PURGE)
===>
DEL((GOAL * PURGE));
REPL((GOAL *S* ATTEND)  ;  (GOAL * ATTEND))    )
)
(PDLISTS:
(N-SUB-NV2
P9 P11 P13
P000 P00 P0
P1 P2 P3 P4 P5 P6
P6A P6B P6C
P7 P8 P10 P12
P13 P14 P15 P16
P24A
P24 P25 P26 P27
)
(STEPDOWN
P17 P17A P17B P17C P17D P17E P17F
P17G P17H P17I
)
(STEPUP
P18 P18A P18B P18C
)
)
(DEF: STM((GOAL * ATTEND)(TOP TWELVE)(BOTTOM 0)(SEND 2)))
***END
*^C
```

```
RUN BEGINS :

((GOAL * ATTEND) (TOP TWELVE) (BOTTOM 0) (SEND 2) JUNK JUNK JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(GO)


STM: ((GO) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM 0) (SEND 2) JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK)
P00


STM: ((GOAL ATTEND) (TOP TWELVE) (BOTTOM 0) (SEND 2) JUNK JUNK JUNK J
UNK JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y 1 Z) (GOAL * SEND 2) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTON
0) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 2) (Y 1 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM 0) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 2) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM 0) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 2) (Y 1 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM 0) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P5

*******  1


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 2) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM 0) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P6


STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 2) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM 0) (SEND 2) JUNK JUNK JUNK JUNK)
P8
```

```
STM: ((GOAL * SEND 2) (Y 2 Z) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
0) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 2) (Y 2 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM 0) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 2) (Y 2 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM 0) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 2) (Y 2 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM 0) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P5

*******  2


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 2) (GOAL *S* SEND 2) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM 0) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P6A


STM: ((Y 3 Z) (GOAL MARK) (SAID 2) (GOAL COUNT) (GOAL *S* SEND 2) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM 0) (SEND 2) JUNK JUNK JUNK JUNK)
P7


STM: ((Y 3 Z) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM 0) (SEND 2) JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TOP BEFORE)


STM: ((MANY TOP BEFORE) (GOAL * ATTEND) (Y 3 Z) (TOP TWELVE) (BOTTOM
0) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9


STM: ((GOAL * RECALL TOP) (GOAL *S* ATTEND) (Y 3 Z) (TOP TWELVE) (BOT
TOM 0) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

*******  (TOP TWELVE)


STM: ((TOP TWELVE) (GOAL * ATTEND) (Y 3 Z) (BOTTOM 0) (SEND 2) JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(HAPPY TUNE)
```

```
STM: ((MANY TUBE) (GOAL * ATTEND) (TOP TWELVE) (Y 3 Z) (BOTTOM 0) (SE
ND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (TOP TWELVE) (Y 3 Z) (BO
TTOM 0) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

*******  (SEND 2)


STM: ((SEND 2) (GOAL * ATTEND) (TOP TWELVE) (Y 3 Z) (BOTTOM 0) JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TOP NOW)


STM: ((MANY TOP NOW) (GOAL * ATTEND) (SEND 2) (TOP TWELVE) (Y 3 Z) (B
OTTOM 0) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((Y TWELVE Z) (W 0 V) (GOAL * SUBTRACT) (GOAL *S* ATTEND) (TOP T
WELVE) (SEND 2) (Y 3 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((GOAL * NEXT DOWN) (GOAL *S* SUBTRACT) (W 0 V) (Y TWELVE Z) (GO
AL *S* ATTEND) (TOP TWELVE) (SEND 2) (Y 3 Z) (BOTTOM 0) JUNK JUNK JUN
K JUNK)
P17


STM: ((GOAL * NEXT UP) (Y ELEVEN Z) (GOAL *S* SUBTRACT) (W 0 V) (GOAL
 *S* ATTEND) (TOP TWELVE) (SEND 2) (Y 3 Z) (BOTTOM 0) JUNK JUNK JUNK
JUNK)
P16


STM: ((GOAL * NEXT UP) (Y ELEVEN Z) (GOAL *S* SUBTRACT) (W 0 V) (GOAL
 *S* ATTEND) (TOP TWELVE) (SEND 2) (Y 3 Z) (BOTTOM 0) JUNK JUNK JUNK
JUNK)
P18


STM: ((W 1 V) (Y ELEVEN Z) (GOAL * SUBTRACT) (GOAL *S* ATTEND) (TOP T
WELVE) (SEND 2) (Y 3 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P15
```

```
STM: ((GOAL * NEXT DOWN) (GOAL *S* SUBTRACT) (W 1 V) (Y ELEVEN Z) (GO
AL *S* ATTEND) (TOP TWELVE) (SEND 2) (Y 3 Z) (BOTTOM 0) JUNK JUNK JUN
K JUNK)
P17A


STM: ((GOAL * NEXT UP) (Y TEN Z) (GOAL *S* SUBTRACT) (W 1 V) (GOAL *S
* ATTEND) (TOP TWELVE) (SEND 2) (Y 3 Z) (BOTTOM 0) JUNK JUNK JUNK JUN
K)
P16


STM: ((GOAL * NEXT UP) (Y TEN Z) (GOAL *S* SUBTRACT) (W 1 V) (GOAL *S
* ATTEND) (TOP TWELVE) (SEND 2) (Y 3 Z) (BOTTOM 0) JUNK JUNK JUNK JUN
K)
P18A


STM: ((W 2 V) (Y TEN Z) (GOAL * SUBTRACT) (GOAL *S* ATTEND) (TOP TWEL
VE) (SEND 2) (Y 3 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P14

********  TEN


STM: ((GOAL * PURGE) (W 2 V) (SEND 2) (Y TEN Z) (GOAL *S* ATTEND) (TO
P TWELVE) (Y 3 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P24A


STM: ((GOAL * PURGE) (TOP TEN) (OLD SEND 2) (W 2 V) (GOAL *S* ATTEND)
 (Y 3 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (TOP TEN) (OLD SEND 2) (W 2 V) (GOAL *S* ATTEND)
 (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (TOP TEN) (OLD SEND 2) (GOAL *S* ATTEND) (BOTTOM
 0) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((TOP TEN) (OLD SEND 2) (GOAL * ATTEND) (BOTTOM 0) JUNK JUNK JUN
K JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*(SEND 4)


STM: ((SEND 4) (GOAL ATTEND) (TOP TEN) (BOTTOM 0) JUNK JUNK JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK)
P1
```

```
STM: ((Y 1 Z) (GOAL * SEND 4) (GOAL *S* ATTEND) (TOP TEN) (BOTTOM 0)
(SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 1 Z) (GOAL *S* ATTEND
) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 4) (Y 1 Z) (GOAL *S* ATTEN
D) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

******* 1


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P6


STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK)
P8


STM: ((GOAL * SEND 4) (Y 2 Z) (GOAL *S* ATTEND) (TOP TEN) (BOTTOM 0)
(SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 2 Z) (GOAL *S* ATTEND
) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y 2 Z) (GOAL *S* ATTE
ND) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 4) (Y 2 Z) (GOAL *S* ATTEN
D) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

******* 2
```

```
STM: ((GOAL * MARK) (GOAL COUNT) (SAID 2) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P60


STM: ((Y 3 Z) (GOAL MARK) (SAID 2) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK)
P8


STM: ((GOAL * SEND 4) (Y 3 Z) (GOAL *S* ATTEND) (TOP TEN) (BOTTOM 0)
(SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 3 Z) (GOAL *S* ATTEND
) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y 3 Z) (GOAL *S* ATTE
ND) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 4) (Y 3 Z) (GOAL *S* ATTEN
D) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

******** 3


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 3) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P65


STM: ((Y 4 Z) (GOAL MARK) (SAID 3) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK)
P8


STM: ((GOAL * SEND 4) (Y 4 Z) (GOAL *S* ATTEND) (TOP TEN) (BOTTOM 0)
(SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 4 Z) (GOAL *S* ATTEND
) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y 4 Z) (GOAL *S* ATTE
ND) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4
```

```
STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 4) (Y 4 Z) (GOAL *S* ATTEN
D) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

*******  4


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 4) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P6C


STM: ((Y 5 Z) (GOAL MARK) (SAID 4) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK JUNK JUNK)
P7


STM: ((Y 5 Z) (GOAL * ATTEND) (TOP TEN) (BOTTOM 0) (SEND 4) JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TOP BEFORE)


STM: ((MANY TOP BEFORE) (GOAL * ATTEND) (Y 5 Z) (TOP TEN) (BOTTOM 0)
(SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9


STM: ((GOAL * RECALL TOP) (GOAL *S* ATTEND) (Y 5 Z) (TOP TEN) (BOTTOM
 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

*******  (TOP TEN)


STM: ((TOP TEN) (GOAL * ATTEND) (Y 5 Z) (BOTTOM 0) (SEND 4) JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)


STM: ((MANY TUBE) (GOAL * ATTEND) (TOP TEN) (Y 5 Z) (BOTTOM 0) (SEND
4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (TOP TEN) (Y 5 Z) (BOTTO
M 0) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

*******  (SEND 4)
```

```
STM: ((SEND 4) (GOAL * ATTEND) (TOP TEN) (Y 5 Z) (BOTTOM 0) JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*((MANY TOP NOW)


STM: ((MANY TOP NOW) (GOAL * ATTEND) (SEND 4) (TOP TEN) (Y 5 Z) (BOTT
OM 0) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((Y TEN Z) (W 0 V) (GOAL * SUBTRACT) (GOAL *S* ATTEND) (TOP TEN)
 (SEND 4) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((GOAL * NEXT DOWN) (GOAL *S* SUBTRACT) (W 0 V) (Y TEN Z) (GOAL
*S* ATTEND) (TOP TEN) (SEND 4) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK
)
P17B


STM: ((GOAL * NEXT UP) (Y NINE Z) (GOAL *S* SUBTRACT) (W 0 V) (GOAL *
S* ATTEND) (TOP TEN) (SEND 4) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK)
P16


STM: ((GOAL * NEXT UP) (Y NINE Z) (GOAL *S* SUBTRACT) (W 0 V) (GOAL *
S* ATTEND) (TOP TEN) (SEND 4) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK)
P18


STM: ((W 1 V) (Y NINE Z) (GOAL * SUBTRACT) (GOAL *S* ATTEND) (TOP TEN
) (SEND 4) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((GOAL * NEXT DOWN) (GOAL *S* SUBTRACT) (W 1 V) (Y NINE Z) (GOAL
 *S* ATTEND) (TOP TEN) (SEND 4) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JUN
K)
P17C


STM: ((GOAL * NEXT UP) (Y EIGHT Z) (GOAL *S* SUBTRACT) (W 1 V) (GOAL
*S* ATTEND) (TOP TEN) (SEND 4) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK
)
P16


STM: ((GOAL * NEXT UP) (Y EIGHT Z) (GOAL *S* SUBTRACT) (W 1 V) (GOAL
*S* ATTEND) (TOP TEN) (SEND 4) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK
)
P19A
```

```
STM: ((W 2 V) (Y EIGHT Z) (GOAL * SUBTRACT) (GOAL *S* ATTEND) (TOP TE
N) (SEND 4) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((GOAL * NEXT DOWN) (GOAL *S* SUBTRACT) (W 2 V) (Y EIGHT Z) (GOA
L *S* ATTEND) (TOP TEN) (SEND 4) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JU
NK)
P17D


STM: ((GOAL * NEXT UP) (Y SEVEN Z) (GOAL *S* SUBTRACT) (W 2 V) (GOAL
*S* ATTEND) (TOP TEN) (SEND 4) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK
)
P16


STM: ((GOAL * NEXT UP) (Y SEVEN Z) (GOAL *S* SUBTRACT) (W 2 V) (GOAL
*S* ATTEND) (TOP TEN) (SEND 4) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK
)
P188


STM: ((W 3 V) (Y SEVEN Z) (GOAL * SUBTRACT) (GOAL *S* ATTEND) (TOP TE
N) (SEND 4) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((GOAL * NEXT DOWN) (GOAL *S* SUBTRACT) (W 3 V) (Y SEVEN Z) (GOA
L *S* ATTEND) (TOP TEN) (SEND 4) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JU
NK)
P17E


STM: ((GOAL * NEXT UP) (Y SIX Z) (GOAL *S* SUBTRACT) (W 3 V) (GOAL *S
* ATTEND) (TOP TEN) (SEND 4) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK)
P16


STM: ((GOAL * NEXT UP) (Y SIX Z) (GOAL *S* SUBTRACT) (W 3 V) (GOAL *S
* ATTEND) (TOP TEN) (SEND 4) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK)
P18C


STM: ((W 4 V) (Y SIX Z) (GOAL * SUBTRACT) (GOAL *S* ATTEND) (TOP TEN)
 (SEND 4) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P14

********  SIX


STM: ((GOAL * PURGE) (W 4 V) (SEND 4) (Y SIX Z) (GOAL *S* ATTEND) (TO
P TEN) (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P23A
```

```
STM: ((GOAL * PURGE) (TOP SIX) (OLD SEND 4) (W 4 V) (GOAL *S* ATTEND)
 (Y 5 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (TOP SIX) (OLD SEND 4) (W 4 V) (GOAL *S* ATTEND)
 (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (TOP SIX) (OLD SEND 4) (GOAL *S* ATTEND) (BOTTOM
 0) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((TOP SIX) (OLD SEND 4) (GOAL * ATTEND) (BOTTOM 0) JUNK JUNK JUN
K JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*(SEND 3)


STM: ((SEND 3) (GOAL ATTEND) (TOP SIX) (BOTTOM 0) JUNK JUNK JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y 1 Z) (GOAL * SEND 3) (GOAL *S* ATTEND) (TOP SIX) (BOTTOM 0)
(SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 3) (Y 1 Z) (GOAL *S* ATTEND
) (TOP SIX) (BOTTOM 0) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 3) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP SIX) (BOTTOM 0) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 3) (Y 1 Z) (GOAL *S* ATTEN
D) (TOP SIX) (BOTTOM 0) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P5

*******  1


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 3) (GOAL *S*
 ATTEND) (TOP SIX) (BOTTOM 0) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P6
```

```
STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 3) (GO
AL *S* ATTEND) (TOP SIX) (BOTTOM 0) (SEND 3) JUNK JUNK JUNK JUNK)
P8


STM: ((GOAL * SEND 3) (Y 2 3) (GOAL *S* ATTEND) (TOP SIX) (BOTTON 0)
(SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 3) (Y 2 Z) (GOAL *S* ATTEND
) (TOP SIX) (BOTTOM 0) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 3) (Y 2 Z) (GOAL *S* ATTE
ND) (TOP SIX) (BOTTOM 0) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 3) (Y 2 Z) (GOAL *S* ATTEN
D) (TOP SIX) (BOTTON 0) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P5

*******  2


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 2) (GOAL *S* SEND 3) (GOAL *S*
 ATTEND) (TOP SIX) (BOTTOM 0) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P6A


STM: ((Y 3 Z) (GOAL MARK) (SAID 2) (GOAL COUNT) (GOAL *S* SEND 3) (GO
AL *S* ATTEND) (TOP SIX) (BOTTOM 0) (SEND 3) JUNK JUNK JUNK JUNK)
P8


STM: ((GOAL * SEND 3) (Y 3 Z) (GOAL *S* ATTEND) (TOP SIX) (BOTTOM 0)
(SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 3) (Y 3 Z) (GOAL *S* ATTEND
) (TOP SIX) (BOTTOM 0) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 3) (Y 3 Z) (GOAL *S* ATTE
ND) (TOP SIX) (BOTTOM 0) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 3) (Y 3 Z) (GOAL *S* ATTEN
D) (TOP SIX) (BOTTON 0) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
```

```
P5

*******  3


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 3) (GOAL *S* SEND 3) (GOAL *S*
 ATTEND) (TOP SIX) (BOTTOM 0) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P6B


STM: ((Y 4 Z) (GOAL MARK) (SAID 3) (GOAL COUNT) (GOAL *S* SEND 3) (GO
AL *S* ATTEND) (TOP SIX) (BOTTOM 0) (SEND 3) JUNK JUNK JUNK JUNK)
P7


STM: ((Y 4 Z) (GOAL * ATTEND) (TOP SIX) (BOTTOM 0) (SEND 3) JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TOP BEFORE)


STM: ((MANY TOP BEFORE) (GOAL * ATTEND) (Y 4 Z) (TOP SIX) (BOTTOM 0)
(SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9


STM: ((GOAL * RECALL TOP) (GOAL *S* ATTEND) (Y 4 Z) (TOP SIX) (BOTTOM
 0) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

*******  (TOP SIX)


STM: ((TOP SIX) (GOAL * ATTEND) (Y 4 Z) (BOTTOM 0) (SEND 3) JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)


STM: ((MANY TUBE) (GOAL * ATTEND) (TOP SIX) (Y 4 Z) (BOTTOM 0) (SEND
3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (TOP SIX) (Y 4 Z) (BOTTO
M 0) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

*******  (SEND 3)


STM: ((SEND 3) (GOAL * ATTEND) (TOP SIX) (Y 4 Z) (BOTTOM 0) JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK)
```

```
P0
INPUT TO STM:
*((MANY TOP NOW))


STM: ((MANY TOP NOW) (GOAL * ATTEND) (SEND 3) (TOP SIX) (Y 4 Z) (BOTT
OM 0) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((Y SIX Z) (W 0 V) (GOAL * SUBTRACT) (GOAL *S* ATTEND) (TOP SIX)
 (SEND 3) (Y 4 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((GOAL * NEXT DOWN) (GOAL *S* SUBTRACT) (W 0 V) (Y SIX Z) (GOAL
*S* ATTEND) (TOP SIX) (SEND 3) (Y 4 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK
)
P17F


STM: ((GOAL * NEXT UP) (Y FIVE Z) (GOAL *S* SUBTRACT) (W 0 V) (GOAL *
S* ATTEND) (TOP SIX) (SEND 3) (Y 4 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK)
P16


STM: ((GOAL * NEXT UP) (Y FIVE Z) (GOAL *S* SUBTRACT) (W 0 V) (GOAL *
S* ATTEND) (TOP SIX) (SEND 3) (Y 4 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK)
P18


STM: ((W 1 V) (Y FIVE Z) (GOAL * SUBTRACT) (GOAL *S* ATTEND) (TOP SIX
) (SEND 3) (Y 4 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((GOAL * NEXT DOWN) (GOAL *S* SUBTRACT) (W 1 V) (Y FIVE Z) (GOAL
 *S* ATTEND) (TOP SIX) (SEND 3) (Y 4 Z) (BOTTOM 0) JUNK JUNK JUNK JUN
K)
P170


STM: ((GOAL * NEXT UP) (Y FOUR Z) (GOAL *S* SUBTRACT) (W 1 V) (GOAL *
S* ATTEND) (TOP SIX) (SEND 3) (Y 4 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK)
P16


STM: ((GOAL * NEXT UP) (Y FOUR Z) (GOAL *S* SUBTRACT) (W 1 V) (GOAL *
S* ATTEND) (TOP SIX) (SEND 3) (Y 4 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK)
P18A


STM: ((W 2 V) (Y FOUR Z) (GOAL * SUBTRACT) (GOAL *S* ATTEND) (TOP SIX
) (SEND 3) (Y 4 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P17
```

```
STM: ((GOAL * NEXT DOWN) (GOAL *S* SUBTRACT) (W 2 V) (Y FOUR Z) (GOAL
 *S* ATTEND) (TOP SIX) (SEND 3) (Y 4 Z) (BOTTOM 0) JUNK JUNK JUNK JUN
K)
P17H


STM: ((GOAL * NEXT UP) (Y THREE Z) (GOAL *S* SUBTRACT) (W 2 V) (GOAL
*S* ATTEND) (TOP SIX) (SEND 3) (Y 4 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK
)
P16


STM: ((GOAL * NEXT UP) (Y THREE Z) (GOAL *S* SUBTRACT) (W 2 V) (GOAL
*S* ATTEND) (TOP SIX) (SEND 3) (Y 4 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK
)
P18B


STM: ((W 3 V) (Y THREE Z) (GOAL * SUBTRACT) (GOAL *S* ATTEND) (TOP SI
X) (SEND 3) (Y 4 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P14

*******  THREE


STM: ((GOAL * PURGE) (W 3 V) (SEND 3) (Y THREE Z) (GOAL *S* ATTEND) (
TOP SIX) (Y 4 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P24A


STM: ((GOAL * PURGE) (TOP THREE) (OLD SEND 3) (W 3 V) (GOAL *S* ATTEN
D) (Y 4 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (TOP THREE) (OLD SEND 3) (W 3 V) (GOAL *S* ATTEN
D) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (TOP THREE) (OLD SEND 3) (GOAL *S* ATTEND) (BOTT
OM 0) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((TOP THREE) (OLD SEND 3) (GOAL * ATTEND) (BOTTOM 0) JUNK JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*(SEND 1)


STM: ((SEND 1) (GOAL ATTEND) (TOP THREE) (BOTTOM 0) JUNK JUNK JUNK JU
```

```
NK JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y 1 Z) (GOAL * SEND 1) (GOAL *S* ATTEND) (TOP THREE) (BOTTOM 0
) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 1) (Y 1 Z) (GOAL *S* ATTEND
) (TOP THREE) (BOTTOM 0) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 1) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP THREE) (BOTTOM 0) (SEND 1) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 1) (Y 1 Z) (GOAL *S* ATTEN
D) (TOP THREE) (BOTTOM 0) (SEND 1) JUNK JUNK JUNK JUNK JUNK)
P5

*******  1


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 1) (GOAL *S*
 ATTEND) (TOP THREE) (BOTTOM 0) (SEND 1) JUNK JUNK JUNK JUNK JUNK)
P6


STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 1) (GO
AL *S* ATTEND) (TOP THREE) (BOTTOM 0) (SEND 1) JUNK JUNK JUNK JUNK)
P7


STM: ((Y 2 Z) (GOAL * ATTEND) (TOP THREE) (BOTTOM 0) (SEND 1) JUNK JU
NK JUNK JUNK JUNK JUNK JUNK JUNK)
P8
INPUT TO STM:
*(MANY TOP BEFORE)


STM: ((MANY TOP BEFORE) (GOAL * ATTEND) (Y 2 Z) (TOP THREE) (BOTTOM 0
) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9


STM: ((GOAL * RECALL TOP) (GOAL *S* ATTEND) (Y 2 Z) (TOP THREE) (BOTT
OM 0) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

*******  (TOP THREE)
```

```
STM: ((TOP THREE) (GOAL * ATTEND) (Y 2 Z) (BOTTOM 0) (SEND 1) JUNK JU
NK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)


STM: ((MANY TUBE) (GOAL * ATTEND) (TOP THREE) (Y 2 Z) (BOTTOM 0) (SEN
D 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (TOP THREE) (Y 2 Z) (BOT
TOM 0) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

*******  (SEND 1)


STM: ((SEND 1) (GOAL * ATTEND) (TOP THREE) (Y 2 Z) (BOTTOM 0) JUNK JU
NK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TOP NOW)


STM: ((MANY TOP NOW) (GOAL * ATTEND) (SEND 1) (TOP THREE) (Y 2 Z) (BO
TTOM 0) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((Y THREE Z) (W 0 V) (GOAL * SUBTRACT) (GOAL *S* ATTEND) (TOP TH
REE) (SEND 1) (Y 2-Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P15


STM: ((GOAL * NEXT DOWN) (GOAL *S* SUBTRACT) (W 0 V) (Y THREE Z) (GOA
L *S* ATTEND) (TOP THREE) (SEND 1) (Y 2 Z) (BOTTOM 0) JUNK JUNK JUNK
JUNK)
P17I


STM: ((GOAL * NEXT UP) (Y TWO Z) (GOAL *S* SUBTRACT) (W 0 V) (GOAL *S
* ATTEND) (TOP THREE) (SEND 1) (Y 2 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK
)
P16


STM: ((GOAL * NEXT UP) (Y TWO Z) (GOAL *S* SUBTRACT) (W 0 V) (GOAL *S
* ATTEND) (TOP THREE) (SEND 1) (Y 2 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK
)
P18


STM: ((W 1 V) (Y TWO Z) (GOAL * SUBTRACT) (GOAL *S* ATTEND) (TOP THRE
```

```
E) (SEND 1) (Y 2 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P14

*******  TWO


STM: ((GOAL * PURGE) (W 1 V) (SEND 1) (Y TWO Z) (GOAL *S* ATTEND) (TO
P THREE) (Y 2 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK)
P24A


STM: ((GOAL * PURGE) (TOP TWO) (OLD SEND 1) (W 1 V) (GOAL *S* ATTEND)
 (Y 2 Z) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (TOP TWO) (OLD SEND 1) (W 1 V) (GOAL *S* ATTEND)
 (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (TOP TWO) (OLD SEND 1) (GOAL *S* ATTEND) (BOTTOM
 0) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((TOP TWO) (OLD SEND 1) (GOAL * ATTEND) (BOTTOM 0) JUNK JUNK JUN
K JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*()


STM: (NIL (GOAL ATTEND) (TOP TWO) (BOTTOM 0) JUNK JUNK JUNK JUNK JUNK
 JUNK JUNK JUNK JUNK)


END OF RUN

NUMBER OF PRODUCTIONS FIRED : 242

NUMBER OF OPERATIONS EXECUTED : 703

NUMBER OF PRODUCTIONS TRIED : 3750

NUMBER OF SINGLE-ELEMENT COMPARISONS : 60666

SIMULATED TIME : 0

COMPUTER TIME: 340332 MSECS

NUMBER OF CONS-OPERATIONS : 614353

*
^C
```

ADDENDUM 6.

LISTING AND EXECUTION TRACE FOR

N-CYC-NV: TABLE-LOOK-UP METHOD.

```
^C

.
.EDIT KVADD7.PSS
(VARIABLES:
*P 500
(VARIABLES:
(A GENC)
(B GENC)
(C GENC)
(D GENC)
(E GENC)
(<X> SEQC)
(<P> SEQC)
)
(PRODUCTIONS:
(P000 (GOAL * ATTEND)(OLD SEND C)
-->
REPL((GOAL * ATTEND)  ;  (GOAL + ATTEND));
DEL((OLD SEND C));
USER()   )
(P00  (GOAL * ATTEND)(GO)
-->
REPL((GOAL * ATTEND)  ;  (GOAL + ATTEND));
DEL((GO))  )
(P0 (GOAL * ATTEND)
-->
USER()  )
(P1 (GOAL + ATTEND)(TOP A)(BOTTOM B)(SEND C)
-->
REPL((GOAL + ATTEND)  ;  (GOAL *S* ATTEND));
INS((GOAL * SEND C));
INS((Y 1 Z))   )
(P2 (GOAL * SEND C)
-->
REPL((GOAL * SEND C)  ;  (GOAL *S* SEND C));
INS((GOAL * PUSH BUTTON))  )
(P3 (GOAL * PUSH BUTTON)
-->
DEL((GOAL * PUSH BUTTON));
INS((ELM A));
INS((GOAL * SUBIT))  )
(P4  (GOAL * SUBIT)(ELM A)
-->
DEL((GOAL * SUBIT));
INS((QS 1));
DEL((ELM A));
INS((GOAL * COUNT))  )
(P5  (GOAL * COUNT)(QS 1)(Y <X> Z)
-->
REPL((GOAL * COUNT)  ;  (GOAL + COUNT));
DEL((QS 1));
SAY(<X>);
REPL((Y <X> Z)  ;  (SAID <X>));
INS((GOAL * MARK))  )
(P6 (GOAL * MARK)(SAID 1)
-->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y 2 Z))   )
(P6A  (GOAL * MARK)(SAID 2)
-->
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y 3 Z))   )
(P6B  (GOAL * MARK)(SAID 3)
```

```
==>
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y 4 Z))    )
(P6C  (GOAL * MARK)(SAID 4)
==>
REPL((GOAL * MARK)  ;  (GOAL + MARK));
INS((Y 5 Z))    )
(P7 (GOAL + MARK)(GOAL + COUNT)(GOAL *S* SEND C)
(SAID C)
==>
DEL((GOAL *S* SEND C));
DEL((GOAL + MARK));
DEL((GOAL + COUNT));
DEL((SAID C));
REPL((GOAL *S* ATTEND)  ;  (GOAL * ATTEND))  )
(P8  (GOAL + MARK)(GOAL + COUNT)(GOAL *S* SEND C)(SAID D)
==>
REPL((GOAL *S* SEND C)  ;  (GOAL * SEND C));
DEL((GOAL + MARK));
DEL((GOAL + COUNT));
DEL((SAID D))    )
(P9 (GOAL * ATTEND)(MANY BOTTOM BEFORE)
==>
REPL((GOAL * ATTEND)  ;  (GOAL *S* ATTEND));
DEL((MANY BOTTOM BEFORE));
INS((GOAL * RECALL BOTTOM))    )
(P10  (GOAL * RECALL BOTTOM)(BOTTOM B)
==>
SAY((BOTTOM B));
DEL((GOAL * RECALL BOTTOM));
REPL((GOAL *S* ATTEND)  ;  (GOAL * ATTEND))  )
(P11 (GOAL * ATTEND)(MANY TUBE)
==>
REPL((GOAL * ATTEND)  ;  (GOAL *S* ATTEND));
DEL((MANY TUBE));
INS((GOAL * RECALL  TUBE))    )
(P12  (GOAL * RECALL TUBE)(SEND C)
==>
DEL((GOAL * RECALL TUBE));
REPL((GOAL *S* ATTEND)  ;  (GOAL * ATTEND));
SAY((SEND C))    )
(P13 (GOAL * ATTEND)(MANY BOTTOM NOW)
==>
REPL((GOAL * ATTEND)  ;  (GOAL *S* ATTEND));
DEL((MANY BOTTOM NOW));
INS((GOAL * ADD))    )
(P14 (GOAL * ADD)(BOTTOM B)(ABS (GOAL * LIST B))
==>
REPL((GOAL * ADD)  ;  (GOAL *S* ADD));
INS((GOAL * LIST B))  )
(P15 (GOAL * LIST ZERO)(GOAL *S* ADD)
==>
REPL((GOAL *S* ADD)  ;  (GOAL * ADD));
INS((2 TWO Z));
INS((3 THREE Z))    )
(P15A  (GOAL * LIST TWO)(GOAL *S* ADD)
==>
REPL((GOAL *S* ADD)  ;  (GOAL * ADD));
INS((3 FIVE Z));
INS((4 SIX Z))    )
(P15B (GOAL * LIST SIX)(GOAL *S* ADD)
==>
REPL((GOAL *S* ADD)  ;  (GOAL * ADD));
INS((2 EIGHT Z));
INS((3 NINE Z))    )
(P15C  (GOAL * LIST NINE)(GOAL *S* ADD)
```

```
==>
REPL((GOAL *S* ADD)  ;  (GOAL * ADD));
INS((0 NINE Z));
INS((1 TEN Z))     )
(P16   (GOAL * ADD)(SEND C)(C <X> Z)(BOTTOM B)
==>
SAY(<X>);
DEL((GOAL * ADD));
DEL((GOAL * LIST B));
INS((GOAL * UPDATE))    )
(P24A (GOAL * UPDATE)(SEND C)(C <X> Z)(BOTTOM B)
==>
REPL((BOTTOM B)  ;  (BOTTOM <X>));
DEL((GOAL * UPDATE));
INS((GOAL * PURGE));
DEL((C <X> Z))     )
(P24 (GOAL * PURGE)(Y <X> Z)
==>
DEL((Y <X> Z))    )
(P25  (GOAL * PURGE)(C <P> Z)
==>
DEL((C <P> Z))    )
(P26 (GOAL * PURGE)(BOTTOM 12)
==>
STOP)
(P27 (GOAL * PURGE)
==>
DEL((GOAL * PURGE));
REPL((GOAL *S* ATTEND)  ;  (GOAL * ATTEND))    )
(PA9 (GOAL * ATTEND)(MANY TOP BEFORE)
==>
REPL((GOAL * ATTEND)  ;  (GOAL *S* ATTEND));
DEL((MANY TOP BEFORE));
INS((GOAL * RECALL TOP))    )
(PA10  (GOAL * RECALL TOP)(TOP A)
==>
SAY((TOP A));
DEL((GOAL * RECALL TOP));
REPL((GOAL *S* ATTEND)  ;  (GOAL * ATTEND))  )
(PA13 (GOAL * ATTEND)(MANY TOP NOW)
==>
REPL((GOAL * ATTEND)  ;  (GOAL *S* ATTEND));
DEL((MANY TOP NOW));
INS((GOAL * SUBTRACT))      )
(PA14 (GOAL * SUBTRACT)(TOP A)(ABS (GOAL * LIST A))
==>
REPL((GOAL * SUBTRACT)  ;  (GOAL *S* SUBTRACT));
INS((GOAL * LIST A))  )
(PA15 (GOAL * LIST THREE)(GOAL *S* SUBTRACT)
==>
REPL((GOAL *S* SUBTRACT)  ;  (GOAL * SUBTRACT));
INS((1 TWO Z));
INS((2 ONE Z))    )
(PA15A  (GOAL * LIST SIX)(GOAL *S* SUBTRACT)
==>
REPL((GOAL *S* SUBTRACT)  ;  (GOAL * SUBTRACT));
INS((2 FOUR Z));
INS((3 THREE Z))    )
(PA15B (GOAL * LIST TEN)(GOAL *S* SUBTRACT)
==>
REPL((GOAL *S* SUBTRACT)  ;  (GOAL * SUBTRACT));
INS((3 SEVEN Z));
INS((4 SIX Z))    )
(PA15C  (GOAL * LIST TWELVE)(GOAL *S* SUBTRACT)
==>
REPL((GOAL *S* SUBTRACT)  ;  (GOAL * SUBTRACT));
```

```
INS((1 ELEVEN Z));
INS((2 TEN Z))     )
(PA16    (GOAL * SUBTRACT)(SEND C)(C <X> Z)(TOP A)
-==->
SAY(<X>);
DEL((GOAL * SUBTRACT));
DEL((GOAL * LIST A));
INS((GOAL * PURGE))    )
(PA24A (GOAL * PURGE)(SEND C)(C <X> Z)(TOP A)
-==->
REPL((TOP A)  ;  (TOP <X>));
REPL((SEND C)  ;  (OLD SEND C));
DEL((C <X> Z))     )
) .
(PDLISTS:
(N-CYC-NV1
P9 PA9 P11 P13 PA13
P000 P00 P0
P1 P2 P3 P4 P5 P6
P6A P6B P6C
P7 P8 P10 PA10 P12
P13 P14 P16 P15
P15A P15B P15C
   PA14 PA16 PA15
PA15A PA15B PA15C
P24A
PA24A
P24 P25 P26 P27
)
)
(DEF: STM((GOAL * ATTEND)(TOP TWELVE)(BOTTOM ZERO)(SEND 2)))
***END
*
```

```
RUN
UNDEFINED FUNCTION
BACKTRACE
?-*EVAL

*(RUN: N-CYC-NV1 STM 15 500)



RUN BEGINS :


((GOAL * ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK
UNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(GO)


STM: ((GO) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK
UNK JUNK JUNK JUNK JUNK JUNK)
P00


STM: ((GOAL ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JU
K JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y 1 Z) (GOAL * SEND 2) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 2) (Y 1 Z) (GOAL *S* ATTEN
) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 2) (Y 1 Z) (GOAL *S* ATT
ND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (QS 1) (GOAL *S* SEND 2) (Y 1 Z) (GOAL *S* ATTE
D) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P5

******** 1


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 2) (GOAL *S
 ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK
)
P6
```

```
STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 2) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUN
K)
P8


STM: ((GOAL * SEND 2) (Y 2 Z) (GOAL *S* ATTEND) (TOP TWELVE) (BOTTOM
ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 2) (Y 2 Z) (GOAL *S* ATTEND
) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 2) (Y 2 Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (QS 1) (GOAL *S* SEND 2) (Y 2 Z) (GOAL *S* ATTEN
D) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK)
P5

******** 2


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 2) (GOAL *S* SEND 2) (GOAL *S*
 ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK
)
P6A


STM: ((Y 3 Z) (GOAL MARK) (SAID 2) (GOAL COUNT) (GOAL *S* SEND 2) (GO
AL *S* ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUN
K)
P7


STM: ((Y 3 Z) (GOAL * ATTEND) (TOP TWELVE) (BOTTOM ZERO) (SEND 2) JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM BEFORE)


STM: ((MANY BOTTOM BEFORE) (GOAL * ATTEND) (Y 3 Z) (TOP TWELVE) (BOTT
OM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9


STM: ((GOAL * RECALL BOTTOM) (GOAL *S* ATTEND) (Y 3 Z) (TOP TWELVE) (
BOTTOM ZERO) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10
```

```
********  (BOTTOM ZERO)

STM: ((BOTTOM ZERO) (GOAL * ATTEND) (Y 3 Z) (TOP TWELVE) (SEND 2) JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)


STM: ((MANY TUBE) (GOAL * ATTEND) (BOTTOM ZERO) (Y 3 Z) (TOP TWELVE)
(SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (BOTTOM ZERO) (Y 3 Z) (T
OP TWELVE) (SEND 2) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

********  (SEND 2)


STM: ((SEND 2) (GOAL * ATTEND) (BOTTOM ZERO) (Y 3 Z) (TOP TWELVE) JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TOP BEFORE)


STM: ((MANY TOP BEFORE) (GOAL * ATTEND) (SEND 2) (BOTTOM ZERO) (Y 3 Z
) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA9


STM: ((GOAL * RECALL TOP) (GOAL *S* ATTEND) (SEND 2) (BOTTOM ZERO) (Y
 3 Z) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA10

********  (TOP TWELVE)


STM: ((TOP TWELVE) (GOAL * ATTEND) (SEND 2) (BOTTOM ZERO) (Y 3 Z) JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM NOW)


STM: ((MANY BOTTOM NOW) (GOAL * ATTEND) (TOP TWELVE) (SEND 2) (BOTTOM
 ZERO) (Y 3 Z) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((GOAL * ADD) (GOAL *S* ATTEND) (TOP TWELVE) (SEND 2) (BOTTOM ZE
```

```
RG) (Y 3 Z) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P14


STM: ((GOAL * LIST ZERO) (GOAL *S* ADD) (BOTTOM ZERO) (GOAL *S* ATTEN
D) (TOP TWELVE) (SEND 2) (Y 3 Z) JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((3 THREE Z) (2 TWO Z) (GOAL * LIST ZERO) (GOAL * ADD) (BOTTOM Z
ERO) (GOAL *S* ATTEND) (TOP TWELVE) (SEND 2) (Y 3 Z) JUNK JUNK JUNK J
UNK)
P16

********  TWO


STM: ((GOAL * UPDATE) (SEND 2) (2 TWO Z) (BOTTOM ZERO) (3 THREE Z) (G
OAL *S* ATTEND) (TOP TWELVE) (Y 3 Z) JUNK JUNK JUNK JUNK JUNK)
P24A


STM: ((GOAL * PURGE) (SEND 2) (BOTTOM TWO) (3 THREE Z) (GOAL *S* ATTE
ND) (TOP TWELVE) (Y 3 Z) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (SEND 2) (BOTTOM TWO) (3 THREE Z) (GOAL *S* ATTE
ND) (TOP TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (SEND 2) (BOTTOM TWO) (GOAL *S* ATTEND) (TOP TWE
LVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((SEND 2) (BOTTOM TWO) (GOAL * ATTEND) (TOP TWELVE) JUNK JUNK JU
NK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TOP NOW)


STM: ((MANY TOP NOW) (GOAL * ATTEND) (SEND 2) (BOTTOM TWO) (TOP TWELV
E) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA13


STM: ((GOAL * SUBTRACT) (GOAL *S* ATTEND) (SEND 2) (BOTTOM TWO) (TOP
TWELVE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA14


STM: ((GOAL * LIST TWELVE) (GOAL *S* SUBTRACT) (TOP TWELVE) (GOAL *S*
```

ATTEND) (SEND 2) (BOTTOM TWO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA15C

112

STM: ((2 TEN Z) (1 ELEVEN Z) (GOAL * LIST TWELVE) (GOAL * SUBTRACT) (
TOP TWELVE) (GOAL *S* ATTEND) (SEND 2) (BOTTOM TWO) JUNK JUNK JUNK JU
NK JUNK)
PA16

******** TEN

STM: ((GOAL * PURGE) (SEND 2) (2 TEN Z) (TOP TWELVE) (1 ELEVEN Z) (GO
AL *S* ATTEND) (BOTTOM TWO) JUNK JUNK JUNK JUNK JUNK JUNK)
PA24A

STM: ((GOAL * PURGE) (OLD SEND 2) (TOP TEN) (1 ELEVEN Z) (GOAL *S* AT
TEND) (BOTTOM TWO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25

STM: ((GOAL * PURGE) (OLD SEND 2) (TOP TEN) (GOAL *S* ATTEND) (BOTTOM
TWO) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27

STM: ((OLD SEND 2) (TOP TEN) (GOAL * ATTEND) (BOTTOM TWO) JUNK JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*(SEND 4)

STM: ((SEND 4) (GOAL ATTEND) (TOP TEN) (BOTTOM TWO) JUNK JUNK JUNK JU
NK JUNK JUNK JUNK JUNK JUNK)
P1

STM: ((Y 1 Z) (GOAL * SEND 4) (GOAL *S* ATTEND) (TOP TEN) (BOTTOM TWO
) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2

STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 1 Z) (GOAL *S* ATTEND
) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3

STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4

STM: ((GOAL * COUNT) (QS 1) (GOAL *S* SEND 4) (Y 1 Z) (GOAL *S* ATTEN

```
D) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

********  1


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P6


STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK)
P8


STM: ((GOAL * SEND 4) (Y 2 Z) (GOAL *S* ATTEND) (TOP TEN) (BOTTOM TWO
) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 2 Z) (GOAL *S* ATTEND
) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y 2 Z) (GOAL *S* ATTE
ND) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (QS 1) (GOAL *S* SEND 4) (Y 2 Z) (GOAL *S* ATTEN
D) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

********  2


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 2) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P6A


STM: ((Y 3 Z) (GOAL MARK) (SAID 2) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK)
P8


STM: ((GOAL * SEND 4) (Y 3 Z) (GOAL *S* ATTEND) (TOP TEN) (BOTTOM TWO
) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 3 Z) (GOAL *S* ATTEND
) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3
```

```
STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y 3 Z) (GOAL *S* ATTE
ND) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (QS 1) (GOAL *S* SEND 4) (Y 3 Z) (GOAL *S* ATTEN
D) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

********  3


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 3) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P6B


STM: ((Y 4 Z) (GOAL MARK) (SAID 3) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK)
P8


STM: ((GOAL * SEND 4) (Y 4 Z) (GOAL *S* ATTEND) (TOP TEN) (BOTTOM TWO
) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 4) (Y 4 Z) (GOAL *S* ATTEND
) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 4) (Y 4 Z) (GOAL *S* ATTE
ND) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (QS 1) (GOAL *S* SEND 4) (Y 4 Z) (GOAL *S* ATTEN
D) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P5

********  4


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 4) (GOAL *S* SEND 4) (GOAL *S*
 ATTEND) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK)
P6C


STM: ((Y 5 Z) (GOAL MARK) (SAID 4) (GOAL COUNT) (GOAL *S* SEND 4) (GO
AL *S* ATTEND) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JUNK JUNK JUNK)
P7
```

```
STM: ((Y 5 Z) (GOAL * ATTEND) (TOP TEN) (BOTTOM TWO) (SEND 4) JUNK JU
NK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM BEFORE)



STM: ((MANY BOTTOM BEFORE) (GOAL * ATTEND) (Y 5 Z) (TOP TEN) (BOTTOM
TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9


STM: ((GOAL * RECALL BOTTOM) (GOAL *S* ATTEND) (Y 5 Z) (TOP TEN) (BOT
TOM TWO) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

********  (BOTTOM TWO)


STM: ((BOTTOM TWO) (GOAL * ATTEND) (Y 5 Z) (TOP TEN) (SEND 4) JUNK JU
NK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TOP BEFORE)



STM: ((MANY TOP BEFORE) (GOAL * ATTEND) (BOTTOM TWO) (Y 5 Z) (TOP TEN
) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA9


STM: ((GOAL * RECALL TOP) (GOAL *S* ATTEND) (BOTTOM TWO) (Y 5 Z) (TOP
 TEN) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA10

********  (TOP TEN)


STM: ((TOP TEN) (GOAL * ATTEND) (BOTTOM TWO) (Y 5 Z) (SEND 4) JUNK JU
NK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)



STM: ((MANY TUBE) (GOAL * ATTEND) (TOP TEN) (BOTTOM TWO) (Y 5 Z) (SEN
D 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (TOP TEN) (BOTTOM TWO) (
Y 5 Z) (SEND 4) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12
```

```
********  (SEND 4)

STM: ((SEND 4) (GOAL * ATTEND) (TOP TEN) (BOTTOM TWO) (Y 5 Z) JUNK JU
NK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM BEFORE)

STM: ((MANY BOTTOM NOW) (GOAL * ATTEND) (SEND 4) (TOP TEN) (BOTTOM TW
O) (Y 5 Z) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13

STM: ((GOAL * ADD) (GOAL *S* ATTEND) (SEND 4) (TOP TEN) (BOTTOM TWO)
(Y 5 Z) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P14

STM: ((GOAL * LIST TWO) (GOAL *S* ADD) (BOTTOM TWO) (GOAL *S* ATTEND)
 (SEND 4) (TOP TEN) (Y 5 Z) JUNK JUNK JUNK JUNK JUNK JUNK)
P15A

STM: ((4 SIX Z) (3 FIVE Z) (GOAL * LIST TWO) (GOAL * ADD) (BOTTOM TWO
) (GOAL *S* ATTEND) (SEND 4) (TOP TEN) (Y 5 Z) JUNK JUNK JUNK JUNK)
P16

********  SIX

STM: ((GOAL * UPDATE) (SEND 4) (4 SIX Z) (BOTTOM TWO) (3 FIVE Z) (GOA
L *S* ATTEND) (TOP TEN) (Y 5 Z) JUNK JUNK JUNK JUNK JUNK)
P24A

STM: ((GOAL * PURGE) (SEND 4) (BOTTOM SIX) (3 FIVE Z) (GOAL *S* ATTEN
D) (TOP TEN) (Y 5 Z) JUNK JUNK JUNK JUNK JUNK JUNK)
P24

STM: ((GOAL * PURGE) (SEND 4) (BOTTOM SIX) (3 FIVE Z) (GOAL *S* ATTEN
D) (TOP TEN) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25

STM: ((GOAL * PURGE) (SEND 4) (BOTTOM SIX) (GOAL *S* ATTEND) (TOP TEN
) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27

STM: ((SEND 4) (BOTTOM SIX) (GOAL * ATTEND) (TOP TEN) JUNK JUNK JUNK
JUNK JUNK JUNK JUNK JUNK JUNK)
P0
```

```
INPUT TO STM:
*(MANY TOP NOW)


STM: ((MANY TOP NOW) (GOAL * ATTEND) (SEND 4) (BOTTOM SIX) (TOP TEN)
JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA13


STM: ((GOAL * SUBTRACT) (GOAL *S* ATTEND) (SEND 4) (BOTTOM SIX) (TOP
TEN) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA14


STM: ((GOAL * LIST TEN) (GOAL *S* SUBTRACT) (TOP TEN) (GOAL *S* ATTEN
D) (SEND 4) (BOTTOM SIX) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA15B


STM: ((4 SIX Z) (3 SEVEN Z) (GOAL * LIST TEN) (GOAL * SUBTRACT) (TOP
TEN) (GOAL *S* ATTEND) (SEND 4) (BOTTOM SIX) JUNK JUNK JUNK JUNK JUNK
)
PA16

******** SIX


STM: ((GOAL * PURGE) (SEND 4) (4 SIX Z) (TOP TEN) (3 SEVEN Z) (GOAL *
S* ATTEND) (BOTTOM SIX) JUNK JUNK JUNK JUNK JUNK JUNK)
PA24A


STM: ((GOAL * PURGE) (OLD SEND 4) (TOP SIX) (3 SEVEN Z) (GOAL *S* ATT
END) (BOTTOM SIX) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (OLD SEND 4) (TOP SIX) (GOAL *S* ATTEND) (BOTTOM
 SIX) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((OLD SEND 4) (TOP SIX) (GOAL * ATTEND) (BOTTOM SIX) JUNK JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*(SEND 3)


STM: ((SEND 3) (GOAL ATTEND) (TOP SIX) (BOTTOM SIX) JUNK JUNK JUNK JU
NK JUNK JUNK JUNK JUNK JUNK)
P1
```

```
STM: ((Y 1 Z) (GOAL * SEND 3) (GOAL *S* ATTEND) (TOP SIX) (BOTTOM SIX
) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2
```



```
STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 3) (Y 1 Z) (GOAL *S* ATTEND
) (TOP SIX) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 3) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP SIX) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (QS 1) (GOAL *S* SEND 3) (Y 1 Z) (GOAL *S* ATTEN
D) (TOP SIX) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P5

******** 1


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 3) (GOAL *S*
 ATTEND) (TOP SIX) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P6


STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 3) (GO
AL *S* ATTEND) (TOP SIX) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK)
P8


STM: ((GOAL * SEND 3) (Y 2 Z) (GOAL *S* ATTEND) (TOP SIX) (BOTTOM SIX
) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 3) (Y 2 Z) (GOAL *S* ATTEND
) (TOP SIX) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 3) (Y 2 Z) (GOAL *S* ATTE
ND) (TOP SIX) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (QS 1) (GOAL *S* SEND 3) (Y 2 Z) (GOAL *S* ATTEN
D) (TOP SIX) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P5

******** 2


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 2) (GOAL *S* SEND 3) (GOAL *S*
 ATTEND) (TOP SIX) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
```

```
P6A

STM: ((Y 3 Z) (GOAL MARK) (SAID 2) (GOAL COUNT) (GOAL *S* SEND 3) (GO
AL *S* ATTEND) (TOP SIX) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK)
P8

STM: ((GOAL * SEND 3) (Y 3 Z) (GOAL *S* ATTEND) (TOP SIX) (BOTTOM SIX
) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2

STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 3) (Y 3 Z) (GOAL *S* ATTEND
) (TOP SIX) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK)
P3

STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 3) (Y 3 Z) (GOAL *S* ATTE
ND) (TOP SIX) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P4

STM: ((GOAL * COUNT) (OS 1) (GOAL *S* SEND 3) (Y 3 Z) (GOAL *S* ATTEN
D) (TOP SIX) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P5

******** 3

STM: ((GOAL * MARK) (GOAL COUNT) (SAID 3) (GOAL *S* SEND 3) (GOAL *S*
 ATTEND) (TOP SIX) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK)
P6B

STM: ((Y 4 Z) (GOAL MARK) (SAID 3) (GOAL COUNT) (GOAL *S* SEND 3) (GO
AL *S* ATTEND) (TOP SIX) (BOTTOM SIX) (SEND 3) JUNK JUNK JUNK JUNK)
P7

STM: ((Y 4 Z) (GOAL * ATTEND) (TOP SIX) (BOTTOM SIX) (SEND 3) JUNK JU
NK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM BEFORE)

STM: ((MANY BOTTOM BEFORE) (GOAL * ATTEND) (Y 4 Z) (TOP SIX) (BOTTOM
SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9

STM: ((GOAL * RECALL BOTTOM) (GOAL *S* ATTEND) (Y 4 Z) (TOP SIX) (BOT
TOM SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10
```

```
********  (BOTTOM SIX)


STM: ((BOTTOM SIX) (GOAL * ATTEND) (Y 4 Z) (TOP SIX) (SEND 3) JUNK JU
NK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TOP BEFORE)



STM: ((MANY TOP BEFORE) (GOAL * ATTEND) (BOTTOM SIX) (Y 4 Z) (TOP SIX
) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA9


STM: ((GOAL * RECALL TOP) (GOAL *S* ATTEND) (BOTTOM SIX) (Y 4 Z) (TOP
 SIX) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA10

********  (TOP SIX)


STM: ((TOP SIX) (GOAL * ATTEND) (BOTTOM SIX) (Y 4 Z) (SEND 3) JUNK JU
NK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)



STM: ((MANY TUBE) (GOAL * ATTEND) (TOP SIX) (BOTTOM SIX) (Y 4 Z) (SEN
D 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (TOP SIX) (BOTTOM SIX) (
Y 4 Z) (SEND 3) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

********  (SEND 3)


STM: ((SEND 3) (GOAL * ATTEND) (TOP SIX) (BOTTOM SIX) (Y 4 Z) JUNK JU
NK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM NOW)



STM: ((MANY BOTTOM NOW) (GOAL * ATTEND) (SEND 3) (TOP SIX) (BOTTOM SI
X) (Y 4 Z) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13
```

```
STM: ((GOAL * ADD) (GOAL *S* ATTEND) (SEND 3) (TOP SIX) (BOTTOM SIX)
(Y 4 Z) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P14


STM: ((GOAL * LIST SIX) (GOAL *S* ADD) (BOTTOM SIX) (GOAL *S* ATTEND)
 (SEND 3) (TOP SIX) (Y 4 Z) JUNK JUNK JUNK JUNK JUNK JUNK)
P15B


STM: ((3 NINE Z) (2 EIGHT Z) (GOAL * LIST SIX) (GOAL * ADD) (BOTTOM S
IX) (GOAL *S* ATTEND) (SEND 3) (TOP SIX) (Y 4 Z) JUNK JUNK JUNK JUNK)
P16

********  NINE


STM: ((GOAL * UPDATE) (SEND 3) (3 NINE Z) (BOTTOM SIX) (2 EIGHT Z) (G
OAL *S* ATTEND) (TOP SIX) (Y 4 Z) JUNK JUNK JUNK JUNK JUNK)
P24A


STM: ((GOAL * PURGE) (SEND 3) (BOTTOM NINE) (2 EIGHT Z) (GOAL *S* ATT
END) (TOP SIX) (Y 4 Z) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (SEND 3) (BOTTOM NINE) (2 EIGHT Z) (GOAL *S* ATT
END) (TOP SIX) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (SEND 3) (BOTTOM NINE) (GOAL *S* ATTEND) (TOP SI
X) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((SEND 3) (BOTTOM NINE) (GOAL * ATTEND) (TOP SIX) JUNK JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK)
PQ
INPUT TO STM:
*(MANY TOP NOW)


STM: ((MANY TOP NOW) (GOAL * ATTEND) (SEND 3) (BOTTOM NINE) (TOP SIX)
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA13


STM: ((GOAL * SUBTRACT) (GOAL *S* ATTEND) (SEND 3) (BOTTOM NINE) (TOP
 SIX) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA14


STM: ((GOAL * LIST SIX) (GOAL *S* SUBTRACT) (TOP SIX) (GOAL *S* ATTEN
```

```
D) (SEND 3) (BOTTOM NINE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA15A


STM: ((3 THREE Z) (2 FOUR Z) (GOAL * LIST SIX) (GOAL * SUBTRACT) (TOP
 SIX) (GOAL *S* ATTEND) (SEND 3) (BOTTOM NINE) JUNK JUNK JUNK JUNK JU
NK)
PA16

********  THREE


STM: ((GOAL * PURGE) (SEND 3) (3 THREE Z) (TOP SIX) (2 FOUR Z) (GOAL
*S* ATTEND) (BOTTOM NINE) JUNK JUNK JUNK JUNK JUNK JUNK)
PA24A


STM: ((GOAL * PURGE) (OLD SEND 3) (TOP THREE) (2 FOUR Z) (GOAL *S* AT
TEND) (BOTTOM NINE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (OLD SEND 3) (TOP THREE) (GOAL *S* ATTEND) (BOTT
OM NINE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((OLD SEND 3) (TOP THREE) (GOAL * ATTEND) (BOTTOM NINE) JUNK JUN
K JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*(SEND 1)


STM: ((SEND 1) (GOAL ATTEND) (TOP THREE) (BOTTOM NINE) JUNK JUNK JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK)
P1


STM: ((Y 1 Z) (GOAL * SEND 1) (GOAL *S* ATTEND) (TOP THREE) (BOTTOM N
INE) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P2


STM: ((GOAL * PUSH BUTTON) (GOAL *S* SEND 1) (Y 1 Z) (GOAL *S* ATTEND
) (TOP THREE) (BOTTOM NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK)
P3


STM: ((GOAL * SUBIT) (ELM A) (GOAL *S* SEND 1) (Y 1 Z) (GOAL *S* ATTE
ND) (TOP THREE) (BOTTOM NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK)
P4


STM: ((GOAL * COUNT) (RS 1) (GOAL *S* SEND 1) (Y 1 Z) (GOAL *S* ATTEN
```

```
D) (TOP THREE) (BOTTOM NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK)
P5

********  1


STM: ((GOAL * MARK) (GOAL COUNT) (SAID 1) (GOAL *S* SEND 1) (GOAL *S*
 ATTEND) (TOP THREE) (BOTTOM NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK)
P6


STM: ((Y 2 Z) (GOAL MARK) (SAID 1) (GOAL COUNT) (GOAL *S* SEND 1) (GO
AL *S* ATTEND) (TOP THREE) (BOTTOM NINE) (SEND 1) JUNK JUNK JUNK JUNK
)
P7


STM: ((Y 2 Z) (GOAL * ATTEND) (TOP THREE) (BOTTOM NINE) (SEND 1) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM BEFORE)


STM: ((MANY BOTTOM BEFORE) (GOAL * ATTEND) (Y 2 Z) (TOP THREE) (BOTTO
M NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9


STM: ((GOAL * RECALL BOTTOM) (GOAL *S* ATTEND) (Y 2 Z) (TOP THREE) (B
OTTOM NINE) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

********  (BOTTOM NINE)


STM: ((BOTTOM NINE) (GOAL * ATTEND) (Y 2 Z) (TOP THREE) (SEND 1) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TOP BEFORE)


STM: ((MANY TOP BEFORE) (GOAL * ATTEND) (BOTTOM NINE) (Y 2 Z) (TOP TH
REE) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P9


STM: ((GOAL * RECALL TOP) (GOAL *S* ATTEND) (BOTTOM NINE) (Y 2 Z) (TO
P THREE) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P10

********  (TOP THREE)
```

```
STM: ((TOP THREE) (GOAL * ATTEND) (BOTTOM NINE) (Y 2 Z) (SEND 1) JUN
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY TUBE)


STM: ((MANY TUBE) (GOAL * ATTEND) (TOP THREE) (BOTTOM NINE) (Y 2 Z) (
SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P11


STM: ((GOAL * RECALL TUBE) (GOAL *S* ATTEND) (TOP THREE) (BOTTOM NINE
) (Y 2 Z) (SEND 1) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P12

********  (SEND 1)


STM: ((SEND 1) (GOAL * ATTEND) (TOP THREE) (BOTTOM NINE) (Y 2 Z) JUNK
 JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:
*(MANY BOTTOM NOW)


STM: ((MANY BOTTOM NOW) (GOAL * ATTEND) (SEND 1) (TOP THREE) (BOTTOM
NINE) (Y 2 Z) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P13


STM: ((GOAL * ADD) (GOAL *S* ATTEND) (SEND 1) (TOP THREE) (BOTTOM NIN
E) (Y 2 Z) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P14


STM: ((GOAL * LIST NINE) (GOAL *S* ADD) (BOTTOM NINE) (GOAL *S* ATTEN
D) (SEND 1) (TOP THREE) (Y 2 Z) JUNK JUNK JUNK JUNK JUNK JUNK)
P15C


STM: ((1 TEN Z) (0 NINE Z) (GOAL * LIST NINE) (GOAL * ADD) (BOTTOM NI
NE) (GOAL *S* ATTEND) (SEND 1) (TOP THREE) (Y 2 Z) JUNK JUNK JUNK JUN
K)
P16

********  TEN


STM: ((GOAL * UPDATE) (SEND 1) (1 TEN Z) (BOTTOM NINE) (0 NINE Z) (GO
AL *S* ATTEND) (TOP THREE) (Y 2 Z) JUNK JUNK JUNK JUNK JUNK)
P24A


STM: ((GOAL * PURGE) (SEND 1) (BOTTOM TEN) (0 NINE Z) (GOAL *S* ATTEN
```

```
D) (TOP THREE) (Y 2 Z) JUNK JUNK JUNK JUNK JUNK JUNK)
P24


STM: ((GOAL * PURGE) (SEND 1) (BOTTOM TEN) (0 NINE Z) (GOAL *S* ATTEN
D) (TOP THREE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (SEND 1) (BOTTOM TEN) (GOAL *S* ATTEND) (TOP THR
EE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27


STM: ((SEND 1) (BOTTOM TEN) (GOAL * ATTEND) (TOP THREE) JUNK JUNK JUN
K JUNK JUNK JUNK JUNK JUNK JUNK)
P0
INPUT TO STM:"
*(MANY TOP NOW)


STM: ((MANY TOP NOW) (GOAL * ATTEND) (SEND 1) (BOTTOM TEN) (TOP THREE
) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA13


STM: ((GOAL * SUBTRACT) (GOAL *S* ATTEND) (SEND 1) (BOTTOM TEN) (TOP
THREE) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA14


STM: ((GOAL * LIST THREE) (GOAL *S* SUBTRACT) (TOP THREE) (GOAL *S* A
TTEND) (SEND 1) (BOTTOM TEN) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
PA15


STM: ((2 ONE Z) (1 TWO Z) (GOAL * LIST THREE) (GOAL * SUBTRACT) (TOP
THREE) (GOAL *S* ATTEND) (SEND 1) (BOTTOM TEN) JUNK JUNK JUNK JUNK JU
NK)
PA16

******** TWO


STM: ((GOAL * PURGE) (SEND 1) (1 TWO Z) (TOP THREE) (2 ONE Z) (GOAL *
S* ATTEND) (BOTTOM TEN) JUNK JUNK JUNK JUNK JUNK JUNK)
PA24A


STM: ((GOAL * PURGE) (OLD SEND 1) (TOP TWO) (2 ONE Z) (GOAL *S* ATTEN
D) (BOTTOM TEN) JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P25


STM: ((GOAL * PURGE) (OLD SEND 1) (TOP TWO) (GOAL *S* ATTEND) (BOTTOM
```

```
 TEN) JUNK JUNK JUNK JUNK JUNK JUNK JUNK JUNK)
P27

STM: ((OLD SEND 1) (TOP TWO) (GOAL * ATTEND) (BOTTOM TEN) JUNK JUNK J
UNK JUNK JUNK JUNK JUNK JUNK JUNK)
P000
INPUT TO STM:
*
*()

STM: (NIL (GOAL ATTEND) (TOP TWO) (BOTTOM TEN) JUNK JUNK JUNK JUNK JU
NK JUNK JUNK JUNK JUNK)

END OF RUN

NUMBER OF PRODUCTIONS FIRED : 256

NUMBER OF OPERATIONS EXECUTED : 733

NUMBER OF PRODUCTIONS TRIED : 6120

NUMBER OF SINGLE-ELEMENT COMPARISONS : 115536

SIMULATED TIME : 0

COMPUTER TIME: 503505 MSECS

NUMBER OF CONS-OPERATIONS : 1121342

*
*
```