## Packet #3: 3.6-3.10

**1.** A certain game keeps track of the maximum and minimum scores obtained so far. If `num` represents the most recent score obtained, which of the following algorithms correctly updates the values of the maximum and the minimum?

(A) If `num` is greater than the minimum, set the minimum equal to `num`. Otherwise, if `num` is greater than the maximum, set the maximum equal to `num`.

(B) If `num` is less than the minimum, set the minimum equal to `num`. Otherwise, if `num` is greater than the maximum, set the maximum equal to `num`. ✓

(C) If `num` is less than the minimum, set the minimum equal to `num`. Otherwise, if `num` is less than the maximum, set the maximum equal to `num`.

(D) If `num` is greater than the minimum, set the minimum equal to `num`. Otherwise, if `num` is less than the maximum, set the maximum equal to `num`.

**2.** In a certain game, the integer variable `bonus` is assigned a value based on the value of the integer variable `score`.

- If `score` is greater than `100`, `bonus` is assigned a value that is `10` times `score`.
- If `score` is between `50` and `100` inclusive, `bonus` is assigned the value of `score`.
- If `score` is less than `50`, `bonus` is assigned a value of `0`.

Which of the following code segments assigns `bonus` correctly for all possible integer values of `score`?

Select two answers.

## Packet #3: 3.6-3.10

(A) ✓

```
IF(score > 100)
{
   bonus score * 10
}
ELSE
{
   IF(score ≥ 50)
   {
     bonus score
   }
   ELSE
   {
     bonus 0
   }
}
```

(B)

```
IF(score ≥ 50)
{
   IF(score > 100)
   {
     bonus score * 10
   }
   ELSE
   {
     bonus 0
   }
}
ELSE
{
   bonus score
}
```

## Packet #3: 3.6-3.10

```
IF(score < 50)
{
   bonus 0
}
ELSE
{
   IF(score ≥ 50)
   {
     bonus score
   }
   ELSE
   {
     bonus score * 10
   }
}
```

(C)

```
IF(score < 50)
{
   bonus 0
}
ELSE
{
   IF(score > 100)
   {
     bonus score * 10
   }
   ELSE
   {
     bonus score
   }
}
```

(D) ✓

## Packet #3: 3.6-3.10

**3.** The following algorithm is intended to determine the average height, in centimeters, of a group of people in a room. Each person has a card, a pencil, and an eraser. Step 2 of the algorithm is missing.

Step 1: All people stand up.

Step 2: (missing step)

Step 3: Each standing person finds another standing person and they form a pair. If a person cannot find an unpaired standing person, that person remains standing and waits until the next opportunity to form pairs.

Step 4: In each pair, one person hands their card to the other person and sits down.

Step 5: At this point, the standing person in each pair is holding two cards. The standing person in each pair replaces the top number on their card with the sum of the top numbers on the two cards and replaces the bottom number on their card with the sum of the bottom numbers on the two cards. The sitting partner's card is discarded.

Step 6: Repeat steps 3–5 until there is only one person standing.

Step 7: The last person standing divides the top number by the bottom number to determine the average height.

Which of the following can be used as step 2 so that the algorithm works as intended?

(A) Step 2: Each person writes their height, in centimeters, at the top of the card and writes the number 1 at the bottom of the card. ✓

(B) Step 2: Each person writes their height, in centimeters, at the top of the card and writes the number 2 at the bottom of the card.

(C) Step 2: Each person writes the number 1 at the top of the card and writes their height, in centimeters, at the bottom of the card.

(D) Step 2: Each person writes the number 2 at the top of the card and writes their height, in centimeters, at the bottom of the card.

## Packet #3: 3.6-3.10

**4.** In the following code segment, assume that `x` and `y` have been assigned integer values.

```
sum ← 0
REPEAT x TIMES
{
   REPEAT y TIMES
   {
     sum sum + 1
   }
}
```

At the end of which of the following code segments is the value of `sum` the same as the value of `sum` at the end of the preceding code segment?

Select two answers.

## Packet #3: 3.6-3.10

(A)
```
sum ← 0
z ← x + y
REPEAT z TIMES
{
   sum sum + 1
}
```

(B) ✓
```
sum ← 0
z ← x * y
REPEAT z TIMES
{
   sum sum + 1
}
```

(C)
```
sum ← 0
REPEAT x TIMES
{
   sum sum + 1
}
REPEAT y TIMES
{
   sum sum + 1
}
```

(D) ✓
```
sum ← 0
REPEAT y TIMES
{
   REPEAT x TIMES
   {
     sum sum + 1
   }
}
```

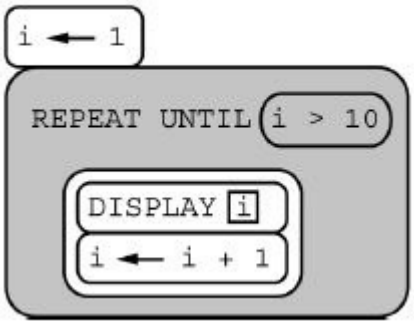
## Packet #3: 3.6-3.10

**5.**

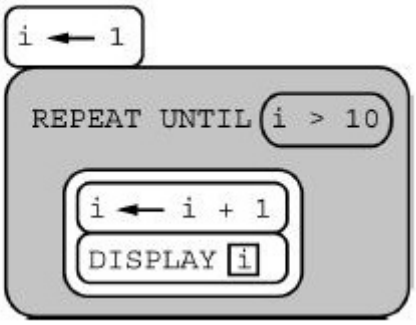
**Directions: The question or incomplete statement below is followed by four suggested answers or completions. Select the one that is best in each case.**

Consider the two programs below.

Program A:




Program B:




Which of the following best compares the values displayed by programs A and B?

(A) Program A and program B display identical values.

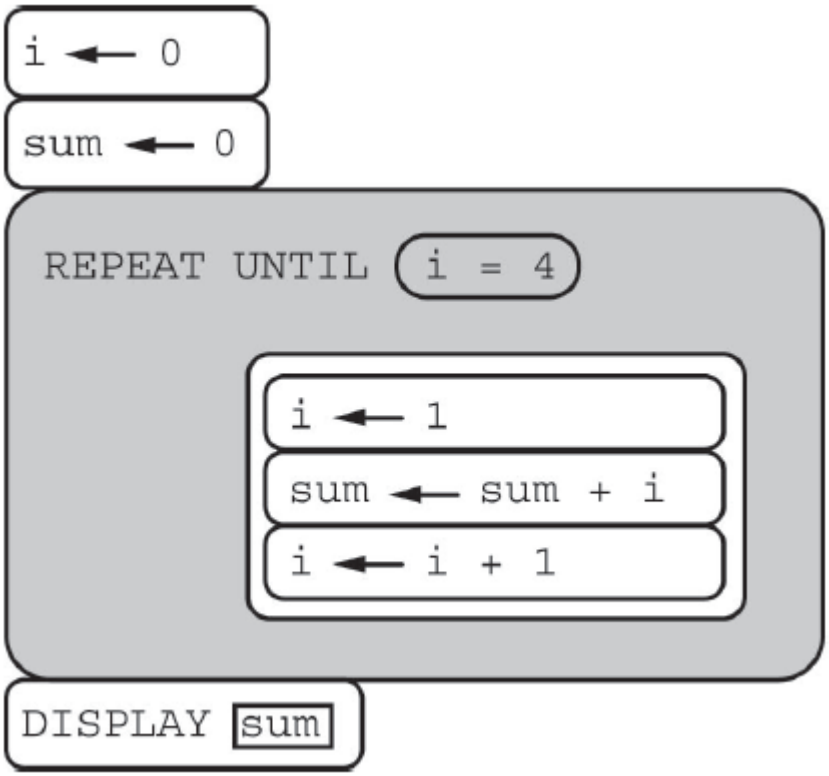
(B) Program A and program B display the same values in different orders.

(C) Program A and program B display the same number of values, but the values differ. ✓

(D) Program A and program B display a different number of values.

## Packet #3: 3.6-3.10

**6.** Consider the following program code.




Which of the following best describes the result of running the program code?

(A) The number *0* is displayed.

(B) The number *6* is displayed.

(C) The number *10* is displayed.

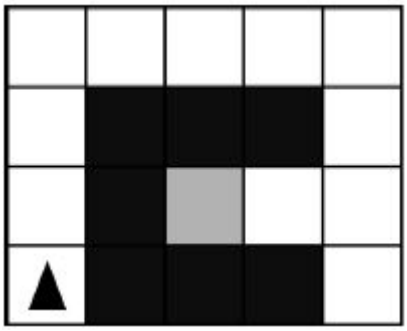
(D) Nothing is displayed; the program results in an infinite loop. ✓

## Packet #3: 3.6-3.10

**7.**

**Directions: The question or incomplete statement below is followed by four suggested answers or completions. Select the one that is best in each case.**

The grid below contains a robot represented as a triangle, initially facing up. The robot can move into a white or gray square but cannot move into a black region.



The code segment below uses the procedure `GoalReached`, which evaluates to `true` if the robot is in the gray square and evaluates to `false` otherwise.

```
REPEAT UNTIL (GoalReached ())
{
   <MISSING CODE>
}
```

Which of the following replacements for `<MISSING CODE>` can be used to move the robot to the gray square?

## Packet #3: 3.6-3.10

(A) ✓

```
IF (CAN_MOVE (right))
{
   ROTATE_RIGHT ()
}
MOVE_FORWARD ()
```

(B)

```
IF (CAN_MOVE (right))
{
   ROTATE_RIGHT ()
   MOVE_FORWARD ()
}
```

(C)

```
IF (CAN_MOVE (forward))
{
   MOVE_FORWARD ()
}
ROTATE_RIGHT ()
```

(D)

```
IF (CAN_MOVE (forward))
{
   MOVE_FORWARD ()
   ROTATE_RIGHT ()
}
```

## Packet #3: 3.6-3.10

**8.** Consider the following code segment, where `exam` and `presentation` are integer variables and `grade` is a string variable.

```
IF((exam > 90) AND (presentation > 80))
{
   grade "A"
}
IF((exam > 80) OR (presentation > 75))
{
   grade "B"
}
ELSE
{
   IF((exam > 70) OR (presentation > 60))
   {
     grade "C"
   }
   ELSE
   {
     IF(exam > 60)
     {
grade ← "D"
     }
     ELSE
     {
grade ← "F"
     }
   }
}
```

Under which of the following conditions will the value `"C"` be assigned to the variable `grade`?

(A) When the value of `exam` is `70` and the value of `presentation` is `50`

(B) When the value of `exam` is `70` and the value of `presentation` is `80`

(C) When the value of `exam` is `80` and the value of `presentation` is `60` ✓

(D) When the value of `exam` is `80` and the value of `presentation` is `80`

**9.** A list of numbers has `n` elements, indexed from `1` to `n`. The following algorithm is intended to display the number of elements in the list that have a value greater than `100`. The algorithm uses the variables `count` and `position`. Steps 3 and 4 are missing.

Step 1: Set `count` to `0` and `position` to `1`.

Step 2: If the value of the element at index `position` is greater than `100`, increase the value of `count` by `1`.

Step 3: (missing step)

Step 4: (missing step)

Step 5: Display the value of `count`.

Which of the following could be used to replace steps 3 and 4 so that the algorithm works as intended?

## Packet #3: 3.6-3.10

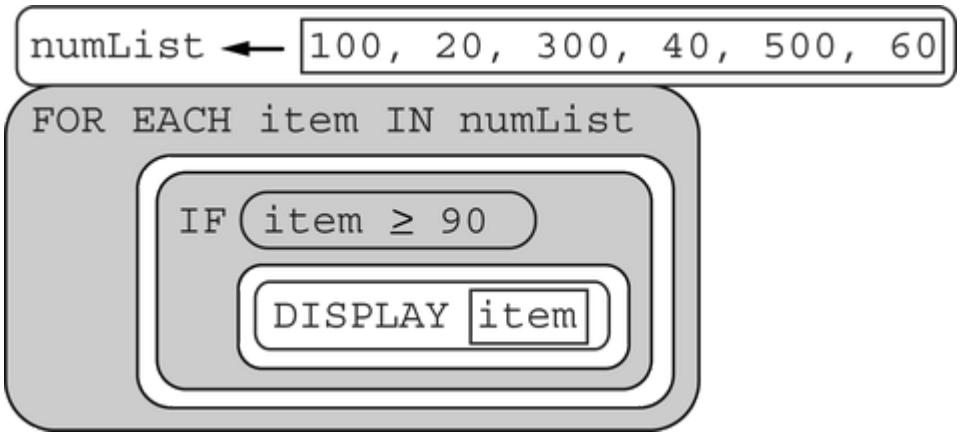
(A) Step 3:
Increase the value of `position` by `1`.
Step 4:
Repeat steps 2 and 3 until the value of `count` is greater than `100`.

(B) Step 3:
Increase the value of `position` by `1`.
Step 4:
Repeat steps 2 and 3 until the value of `position` is greater than `n`. ✓

(C) Step 3:
Repeat step 2 until the value of `count` is greater than `100`.
Step 4:
Increase the value of `position` by `1`.

(D) Step 3:
Repeat step 2 until the value of `position` is greater than `n`.
Step 4: Increase the value of `count` by `1`.

**10.** Consider the following code segment.




What is displayed as a result of executing the code segment?

## Packet #3: 3.6-3.10

(A) 1 3 5

(B) 5 3 1

(C) 100 300 500 ✓

(D) 500 300 100

**11.** In a science experiment, result X is expected to occur 25% of the time and result Y is expected to occur the remaining 75% of the time. The following code segment is intended to simulate the experiment if there are 100 trials.

```
Line 1:  xCount ← 0
Line 2:  yCount ← 0
Line 3:  REPEAT 100 TIMES
Line 4:  {
Line 5:     IF(RANDOM(1, 4) = 1)
Line 6:     {
Line 7:        xCount ← xCount + 1
Line 8:     }
Line 9:     IF(RANDOM(1, 4) > 1)
Line 10:    {
Line 11:       yCount ← yCount + 1
Line 12:    }
Line 13: }
Line 14: DISPLAY("Result X occurred")
Line 15: DISPLAY(xCount)
Line 16: DISPLAY("times and result Y occurred")
Line 17: DISPLAY(yCount)
Line 18: DISPLAY("times.")
```

A programmer runs the code segment, and the following message is displayed.

```
Result X occurred 24 times and result Y occurred 70 times.
```

The result shows that 94 trials were counted, rather than the intended 100 trials. Which of the following changes to the code segment will ensure a correct simulation of the experiment?

## Packet #3: 3.6-3.10

A Replacing line 9 with `IF(RANDOM(1, 4) ≥ 2)`

B Replacing line 9 with `ELSE` ✓

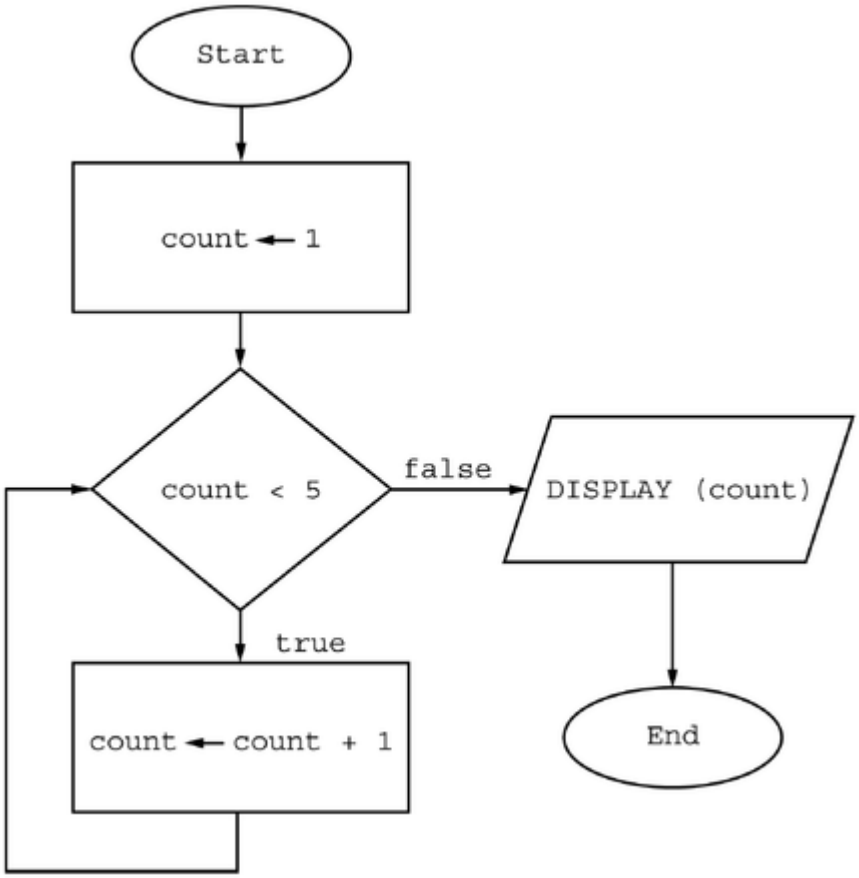
C Interchanging lines 5 and 9

D Interchanging lines 7 and 11

## Packet #3: 3.6-3.10

**12.** A flowchart is a way to visually represent an algorithm. The flowchart below uses the following building blocks.

| Block | Explanation |
|---|---|
| Oval | The start or end of the algorithm |
| Rectangle | One or more processing steps, such as a statement that assigns a value to a variable |
| Diamond | A conditional or decision step, where execution proceeds to the side labeled `true` if the condition is true and to the side labeled `false` otherwise |
| Parallelogram | Displays a message |




What is displayed as a result of executing the algorithm in the flowchart?

## Packet #3: 3.6-3.10

(A) 5 ✓

(B) 15

(C) 1 2 3 4

(D) 1 2 3 4 5

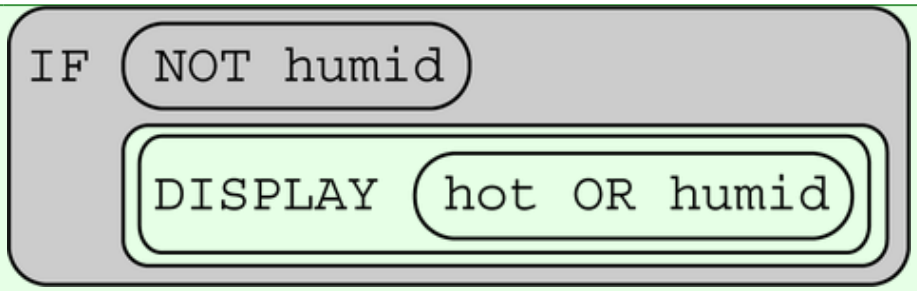
**13.** Assume that the Boolean variable `hot` is assigned the value `true` and the Boolean variable `humid` is assigned the value `false`. Which of the following will display the value `true`?

Select two answers.

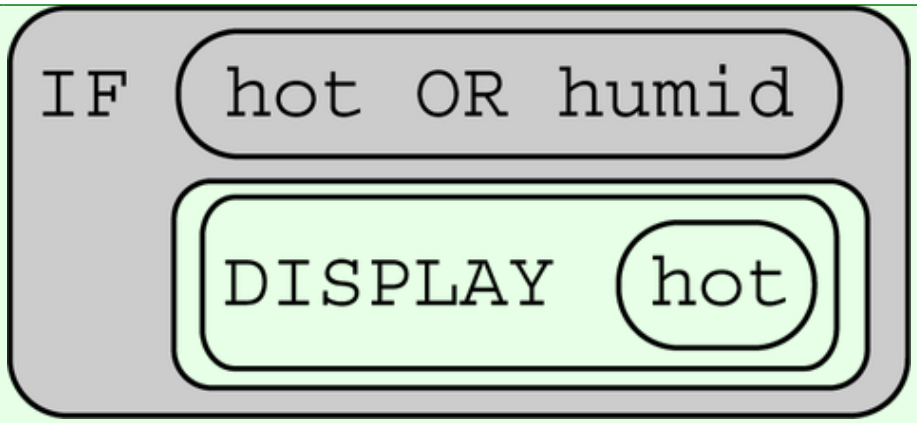
## Packet #3: 3.6-3.10

A

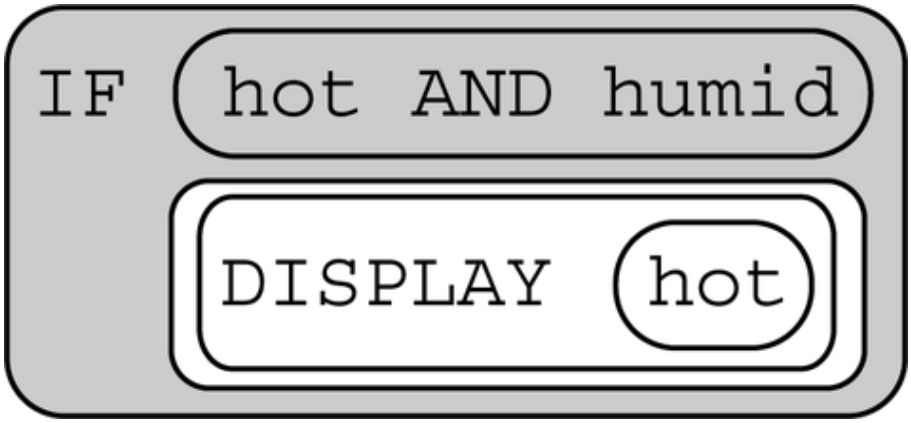

B ✓




C ✓




D



## Packet #3: 3.6-3.10

**14.** Shoppers at a mall were asked whether they preferred wearing gloves or mittens in cold weather. Shoppers' preferences were stored in the list `voteList` as strings, with the string `"Gloves"` representing a preference for gloves and the string `"Mittens"` representing a preference for mittens.

The following code segment is intended to traverse the list and display the number of shoppers who chose gloves and the number of shoppers who chose mittens.

```
numGlovesVotes ← 0
numMittensVotes ← 0
<MISSING CODE>
{
   IF(vote = "Gloves")
   {
     numGlovesVotes numGlovesVotes + 1
   }
   ELSE
   {
     numMittensVotes numMittensVotes + 1
   }
}
DISPLAY(numGlovesVotes)
DISPLAY(" shoppers chose gloves and")
DISPLAY(numMittensVotes)
DISPLAY(" shoppers chose mittens.")
```

Which of the following should replace `<MISSING CODE>` so that the code segment works as intended?

(A) `IF(vote ≤ LENGTH(voteList))`

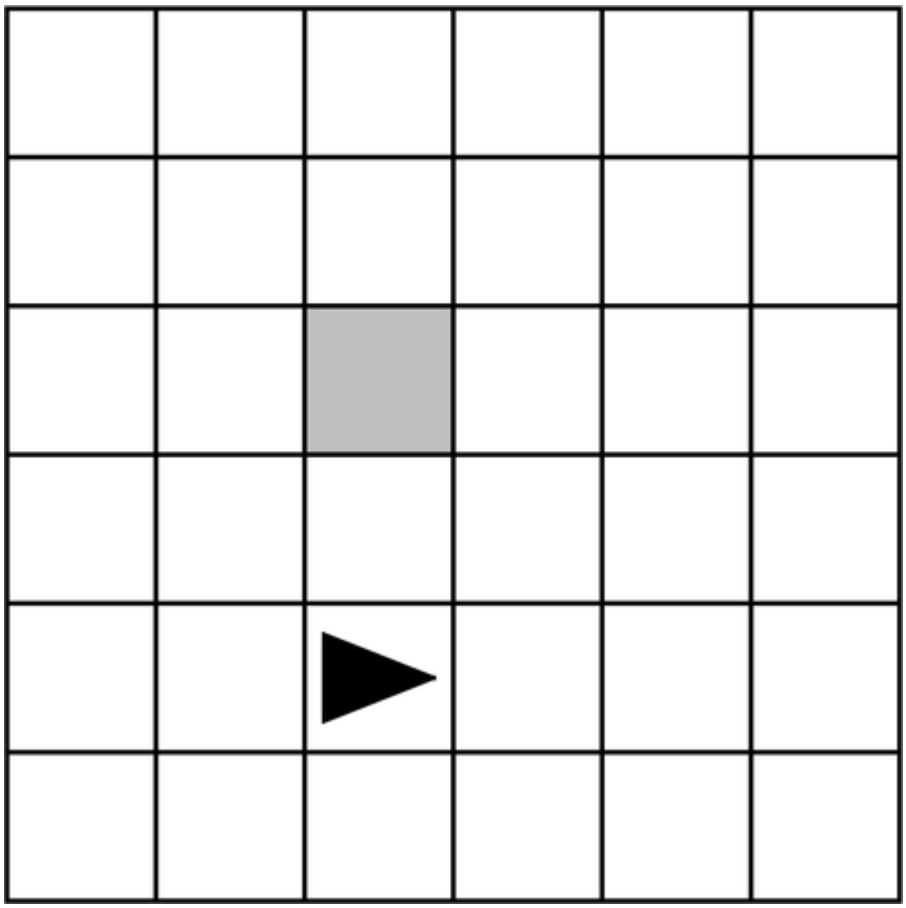
(B) `FOR EACH vote IN voteList` ✓

(C) `REPEAT LENGTH(voteList) TIMES`

(D) `REPEAT UNTIL(vote > LENGTH(voteList))`

## Packet #3: 3.6-3.10

**15.** The following grid contains a robot represented as a triangle, which is initially facing right.



The following code segment is intended to move the robot to the gray square.

```
<MISSING STATEMENT>
{
   REPEAT 4 TIMES
   {
     MOVE_FORWARD()
     ROTATE_RIGHT()
   }
   ROTATE_LEFT()
   MOVE_FORWARD()
   ROTATE_RIGHT()
}
```

Which of the following can be used as a replacement for `<MISSING STATEMENT>` so that the code segment works as intended?

## Packet #3: 3.6-3.10

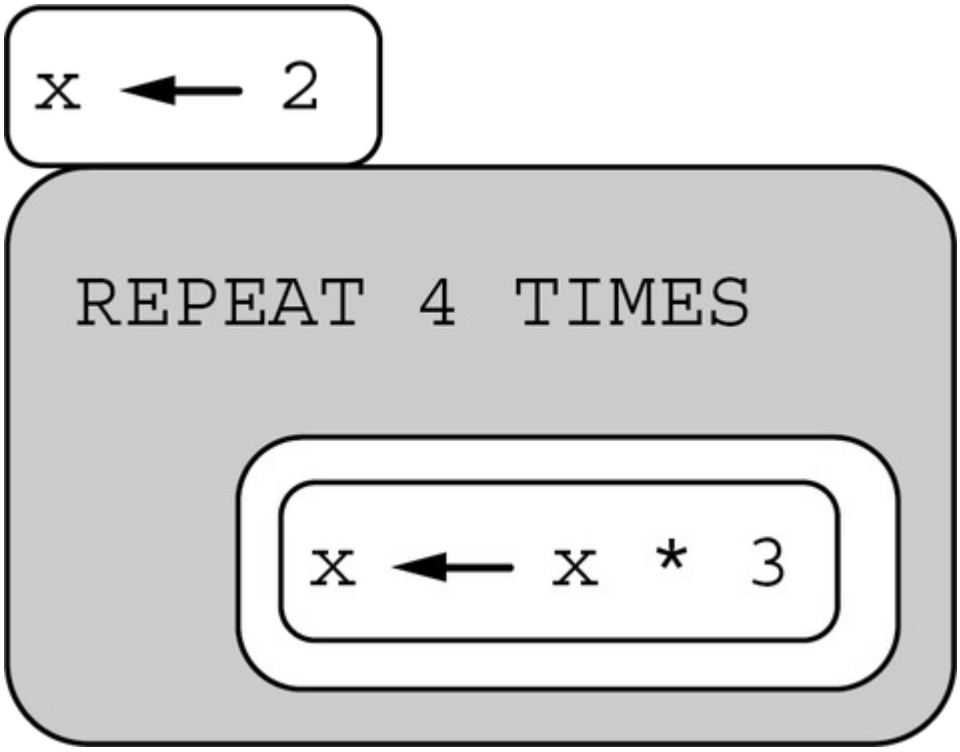
A `REPEAT 1 TIMES`

B `REPEAT 2 TIMES` ✓

C `REPEAT 3 TIMES`

D `REPEAT 4 TIMES`

**16.** Consider the following program.




Which of the following expressions represents the value stored in the variable `x` as a result of executing the program?

A `2 * 3 * 3 * 3`

B `2 * 4 * 4 * 4`

C `2 * 3 * 3 * 3 * 3` ✓

D `2 * 4 * 4 * 4 * 4`

**17.** In the following statement, `val1`, `val2`, and `result` are Boolean variables.

```
result ← val1 AND NOT val2
```

Which of the following code segments produce the same result as the statement above for all possible values of `val1` and `val2`?

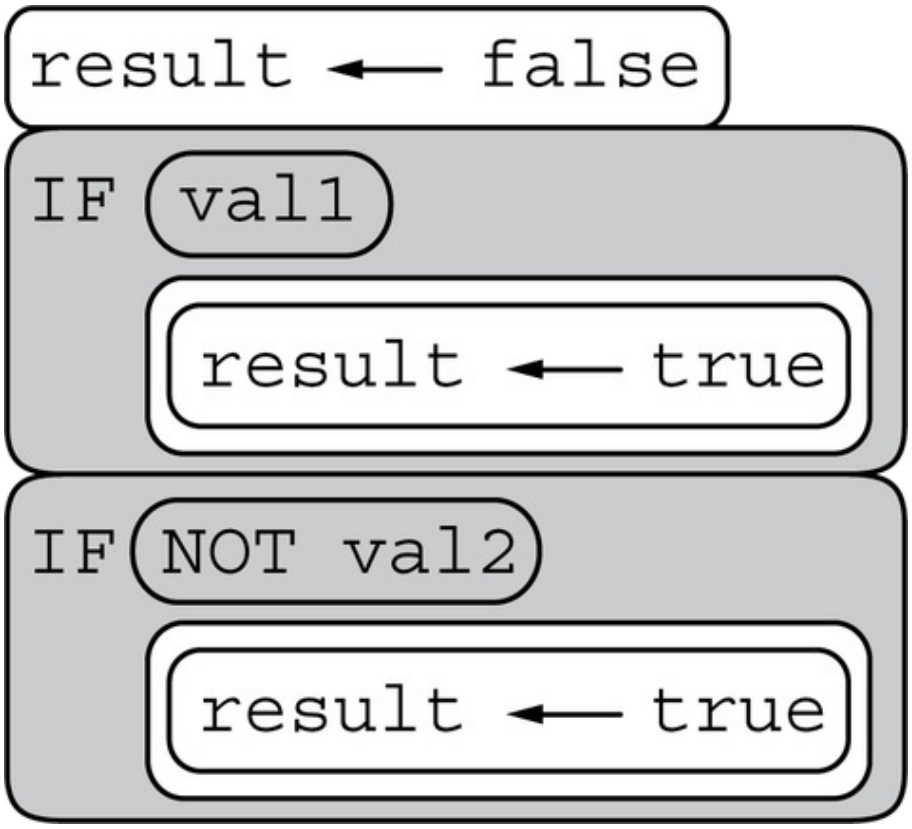
Select two answers.

A ✓

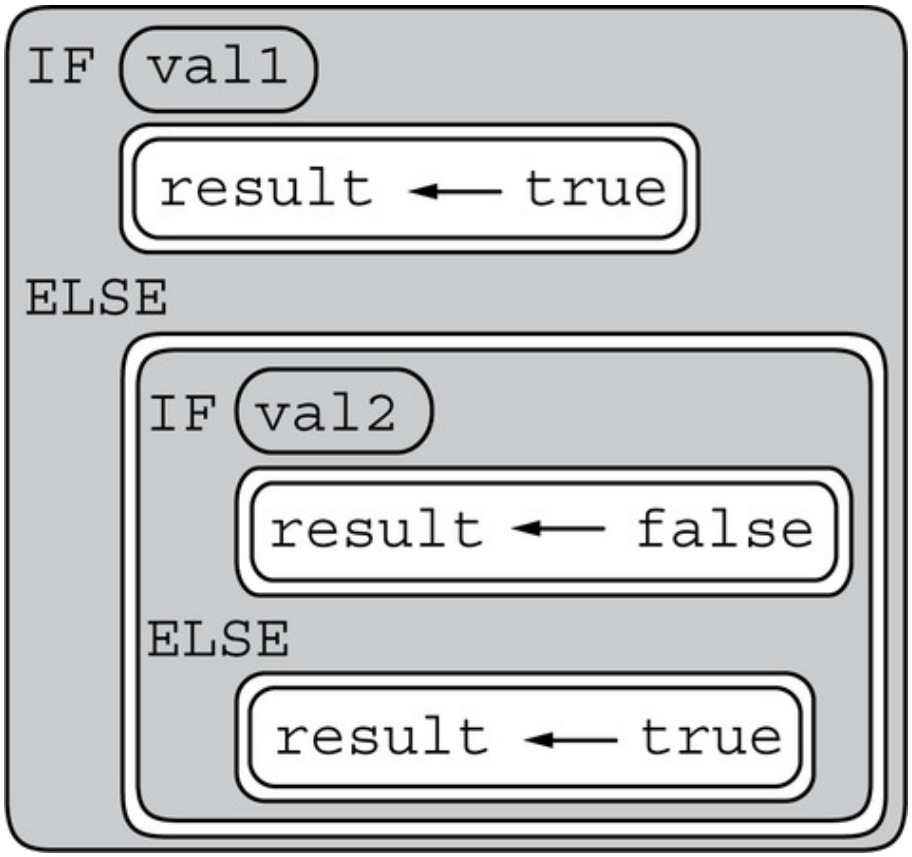
```
result ← false
IF val1
    IF NOT val2
        result ← true
```

**Packet #3: 3.6-3.10**

B

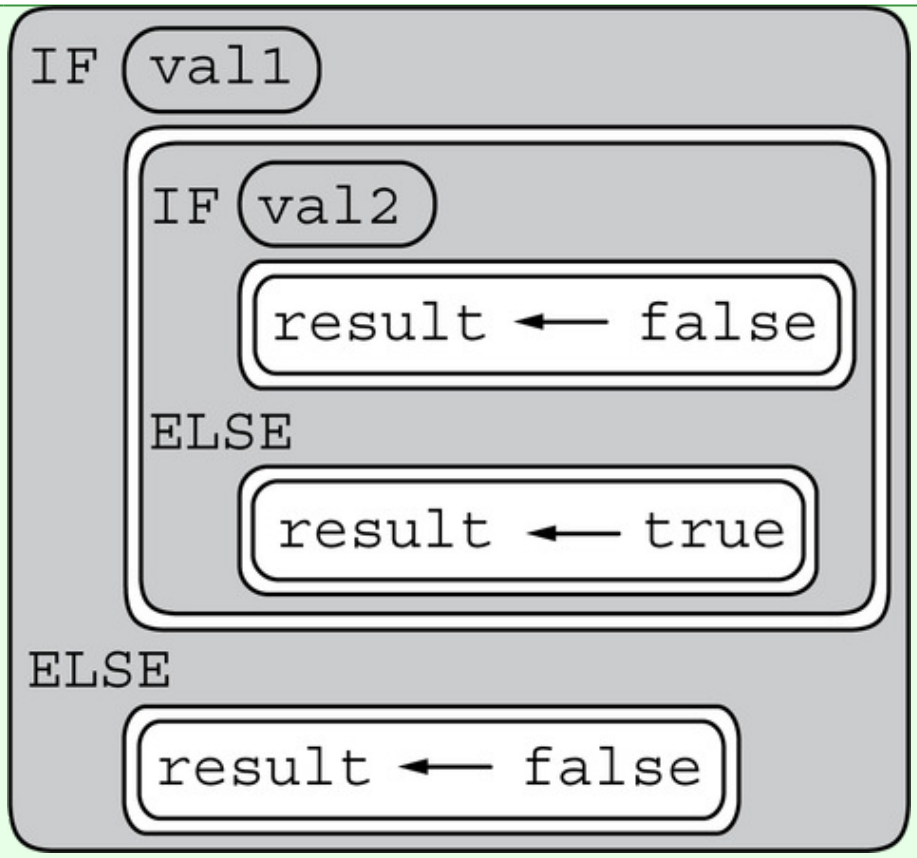



C



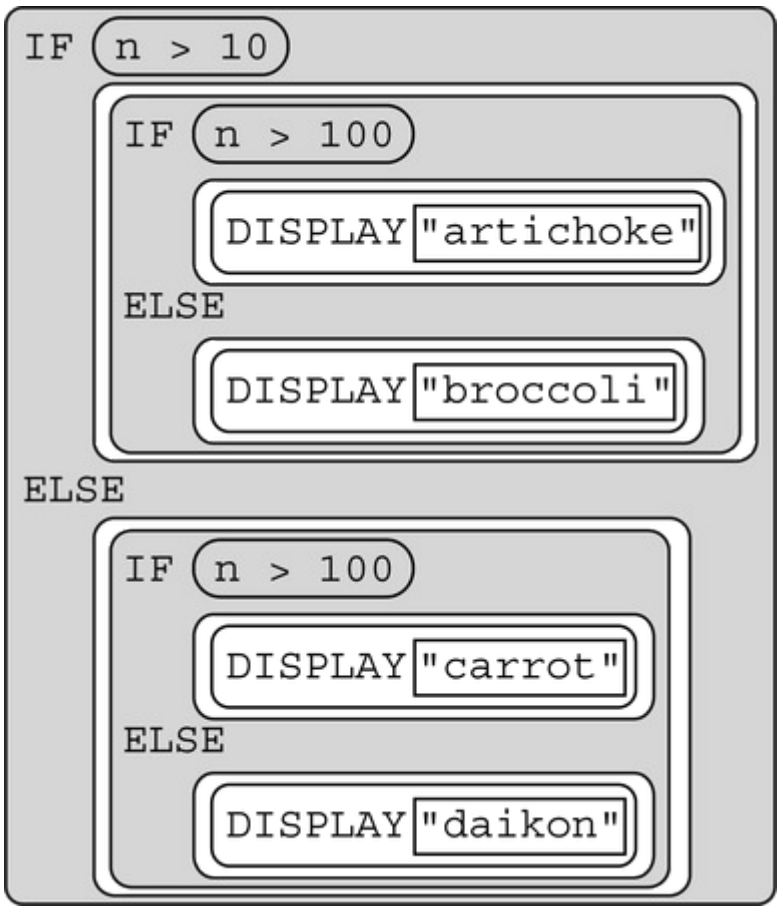
## Packet #3: 3.6-3.10

D



## Packet #3: 3.6-3.10

**18.** In the following code segment, assume that the variable `n` has been initialized with an integer value.




Which of the following is NOT a possible value displayed by the program?

A `"artichoke"`

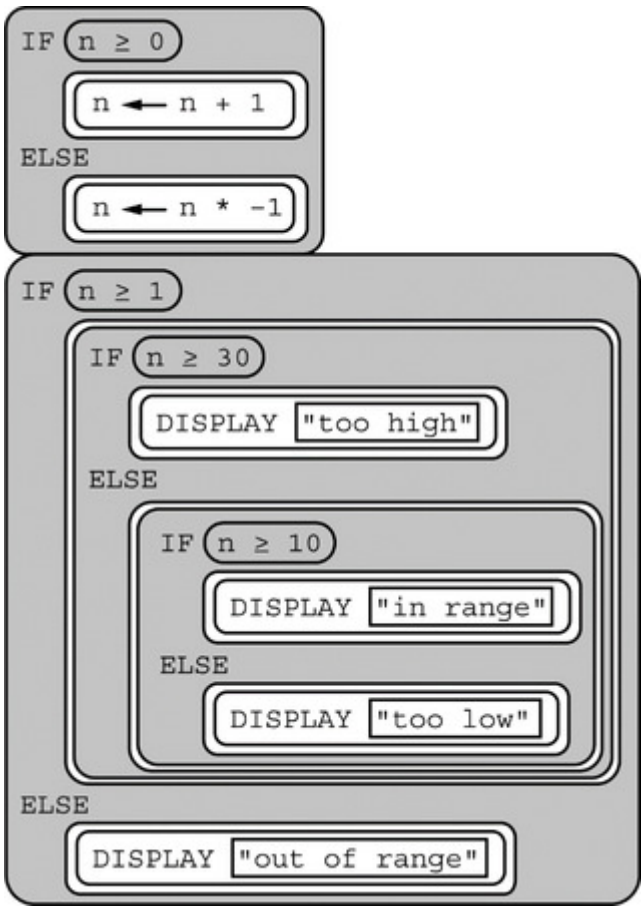
B `"broccoli"`

C `"carrot"` ✓

D `"daikon"`

## Packet #3: 3.6-3.10

**19.** In the following program, assume that the variable `n` has been initialized with an integer value.




Which of the following is NOT a possible value displayed by the program?

(A) too high

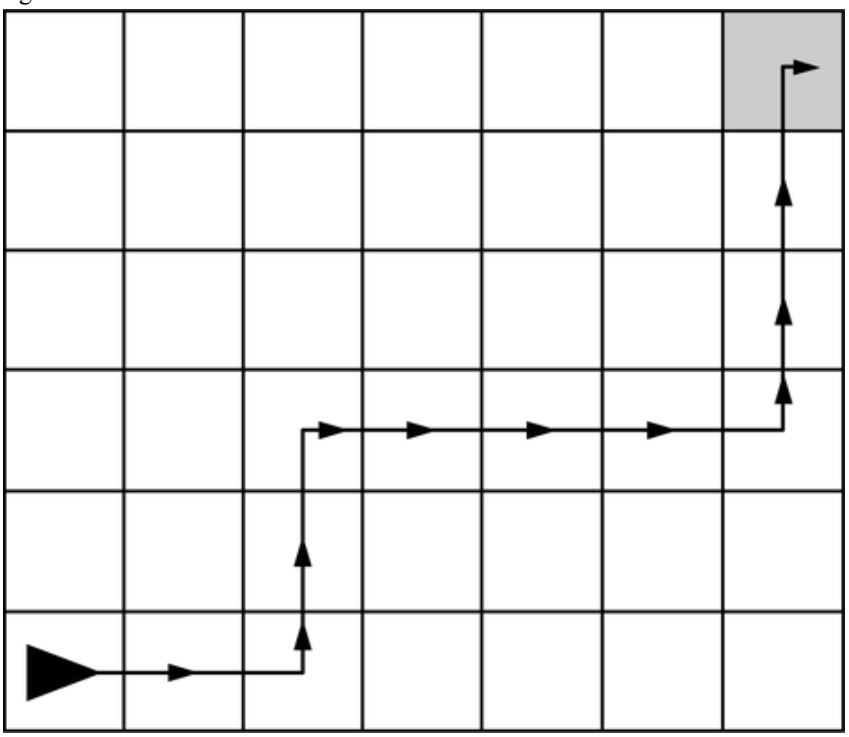
(B) in range

(C) too low

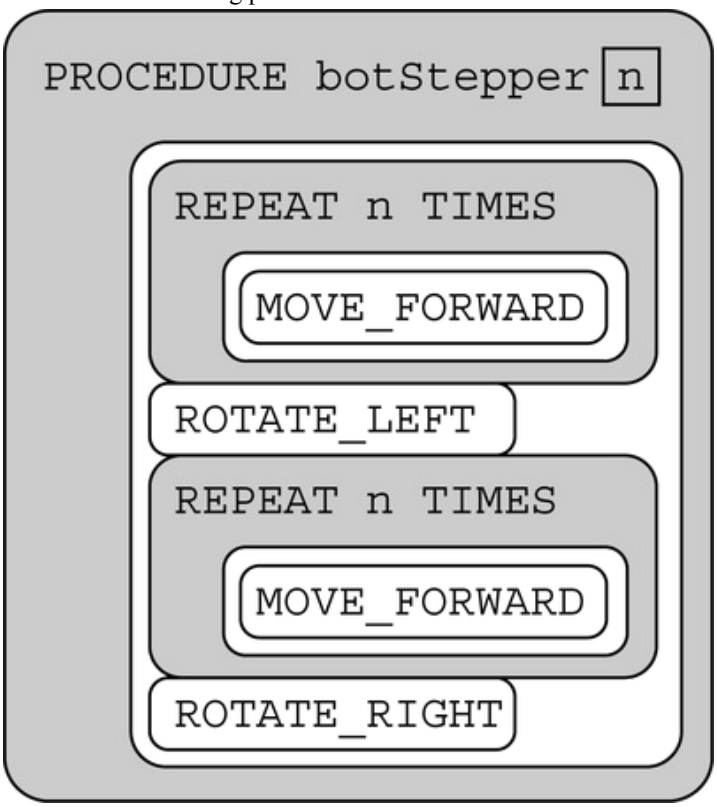
(D) out of range ✓

## Packet #3: 3.6-3.10

**20.** The following question uses a robot in a grid of squares. The robot is represented by a triangle, which is initially facing right.



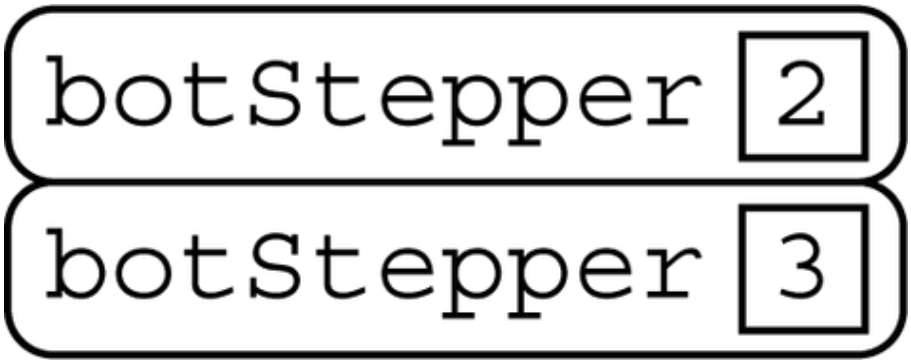
Consider the following procedure.



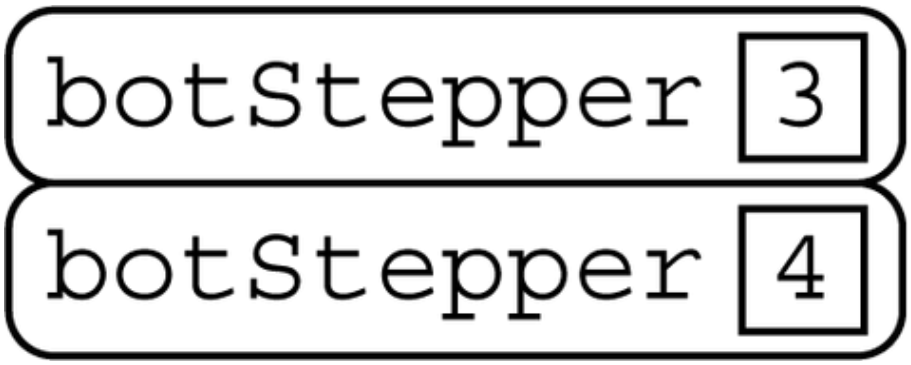

Which of the following code segments will move the robot to the gray square along the path indicated by the arrows?

## Packet #3: 3.6-3.10

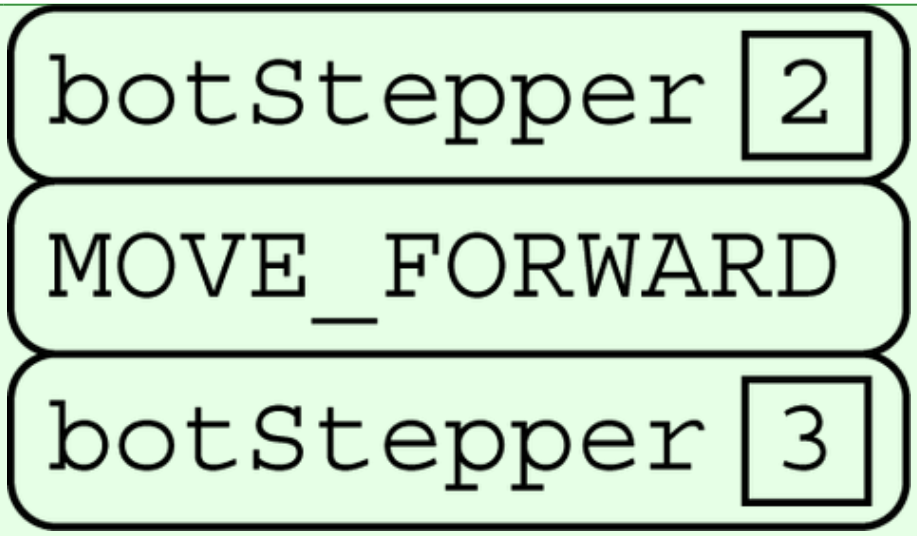
A




B




C ✓



**Packet #3: 3.6-3.10**

D