

Robot Algorithms

Answer Key

Hook

1. [Computer Science Basics - Algorithms](#)

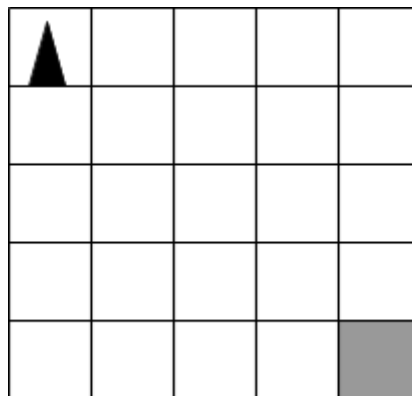
Watch this video and answer the question: What is an **algorithm**?

An algorithm is a set of steps used to complete a task

2. What does the AP® Exam robot look like? Watch [this video](#) and explain how the robot looks and how it works.

The robot on the AP test looks like a tall isosceles triangle. The tip of the triangle is the nose and that represents the front of the robot and the back of the robot is represented as the side that is opposite that tip.

Guided Problem-Solving Activities



1. If you want the robot to end up in the bottom right square facing up, write steps, using pseudocode to complete this task.

Use the space below to write your response:

There are many possible answers. Try:

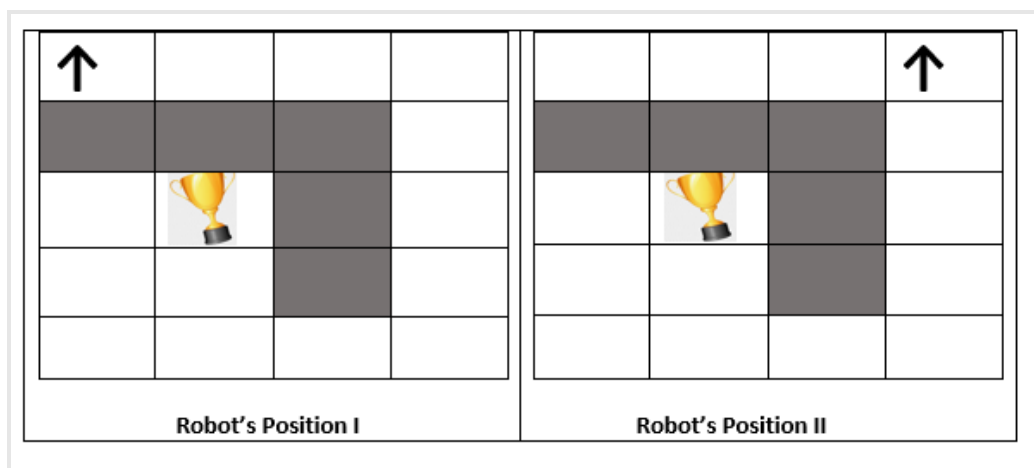
ROTATE RIGHT
MOVE FORWARD 4 TIMES
ROTATE RIGHT
MOVE FORWARD 4 TIMES
ROTATE RIGHT
ROTATE RIGHT

OR

REPEAT 2 TIMES
ROTATE RIGHT
REPEAT 4 TIMES
MOVE FORWARD
REPEAT 2 TIMES
ROTATE RIGHT

Consider the following program:

```
REPEAT UNTIL (foundGold())
{
  IF (CAN_MOVE(forward))
  {
    REPEAT UNTIL (NOT(CAN_MOVE(forward)))
    {
      MOVE_FORWARD()
    }
  }
  ELSE IF (CAN_MOVE(right))
  {
    ROTATE_RIGHT()
    REPEAT UNTIL (NOT(CAN_MOVE(forward)))
    {
      MOVE_FORWARD()
    }
  }
}
ELSE
{
  ROTATE_LEFT()
  REPEAT UNTIL (NOT(CAN_MOVE(forward)))
  {
    MOVE_FORWARD()
  }
}
}
```



2. In which of the following starting positions could the robot be to capture the gold?

- a. **Either Position I or Position II**
- b. Neither Position I nor Position II
- c. Position I only
- d. Position II only

Consider the following code:

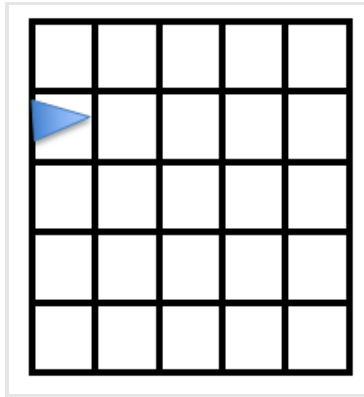
```
MOVE_FORWARD()
MOVE_FORWARD()
ROTATE_LEFT()
MOVE_FORWARD()
MOVE_FORWARD()
ROTATE_RIGHT()
MOVE_FORWARD()
ROTATE_LEFT()
```

<p>A</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 25%; height: 20px;"></td><td style="width: 25%; height: 20px;"></td><td style="width: 25%; height: 20px;"></td><td style="width: 25%; height: 20px; text-align: right;">→End</td></tr> <tr><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td></tr> <tr><td style="height: 20px; text-align: right;">→ Start</td><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td></tr> </table>				→End									→ Start								<p>B</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 25%; height: 20px;"></td><td style="width: 25%; height: 20px;"></td><td style="width: 25%; height: 20px; text-align: right;">← End</td><td style="width: 25%; height: 20px;"></td></tr> <tr><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td></tr> <tr><td style="height: 20px; text-align: right;">→ Start</td><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td></tr> </table>			← End										→ Start							
			→End																																						
→ Start																																									
		← End																																							
→ Start																																									
<p>C</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 25%; height: 20px;"></td><td style="width: 25%; height: 20px;"></td><td style="width: 25%; height: 20px;"></td><td style="width: 25%; height: 20px; text-align: right;">↓End</td></tr> <tr><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td></tr> <tr><td style="height: 20px; text-align: right;">↑ Start</td><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td></tr> </table>				↓End									↑ Start								<p>D</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 25%; height: 20px;"></td><td style="width: 25%; height: 20px;"></td><td style="width: 25%; height: 20px;"></td><td style="width: 25%; height: 20px;"></td></tr> <tr><td style="height: 20px; text-align: right;">↓ Start</td><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;"></td><td style="height: 20px;"></td><td style="height: 20px; text-align: right;">→End</td><td style="height: 20px;"></td></tr> </table>					↓ Start														→End	
			↓End																																						
↑ Start																																									
↓ Start																																									
		→End																																							

3. Which of the diagrams below match the code above?
 - a. A
 - b. B
 - c. C
 - d. D

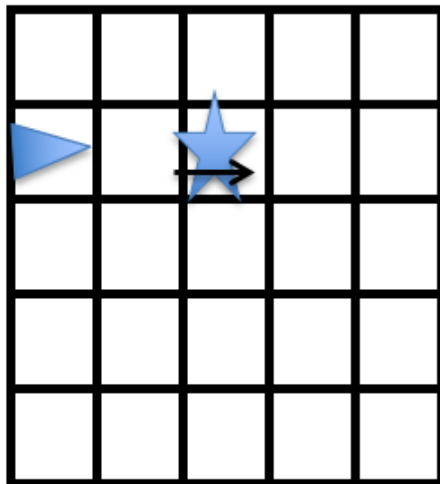
Remember Rotate does not move, just rotates 90 degrees.

4. Trace the code given and indicate where the robot will end on the grid. Indicate the final location on the grid with a star. Also indicate the direction the robot is facing with an arrow.

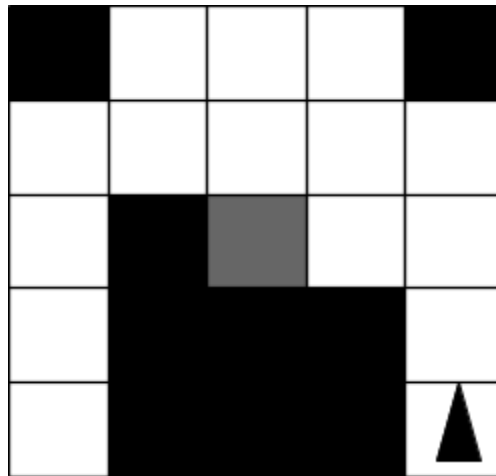


```
ROTATE_LEFT()
MOVE_FORWARD()
ROTATE_RIGHT()
MOVE_FORWARD()
ROTATE_RIGHT()
MOVE_FORWARD()
ROTATE_LEFT()
MOVE_FORWARD()
```

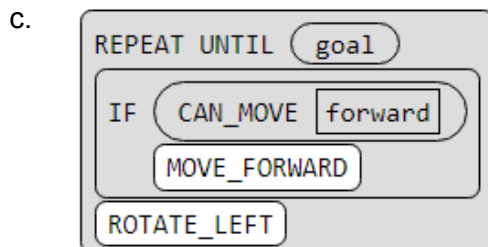
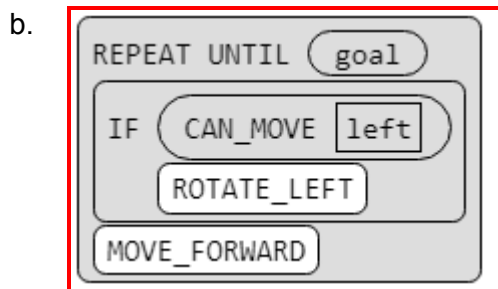
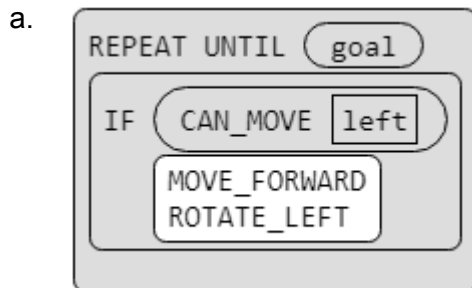
Draw your answer below:

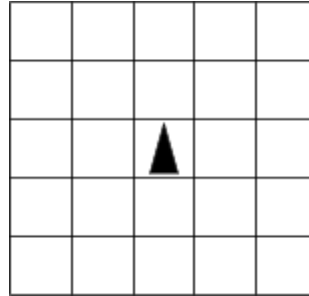


The robot cannot move into black boxes. The robot's goal is to reach the gray box. The code segment uses the procedure `goal()` which evaluates to `true` if the robot gets inside the gray box.



5. Which of the following code can be used to return `true`?

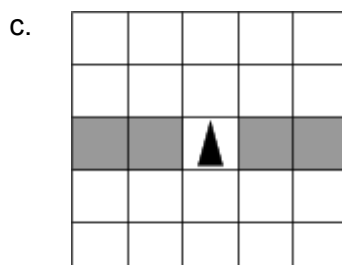
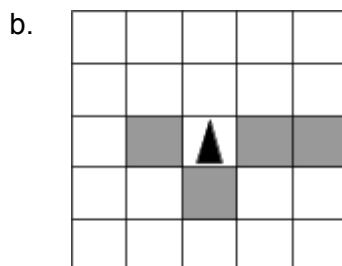
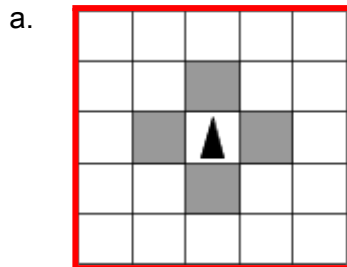




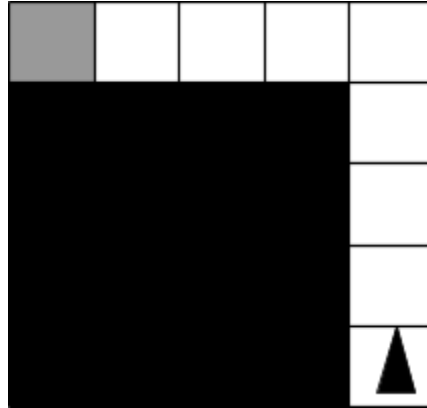
The following code segment is used to move the robot within the grid.

```
x ← RANDOM (1, 4)
REPEAT X TIMES
{
    ROTATE_RIGHT ( )
}
MOVE_FORWARD ( )
```

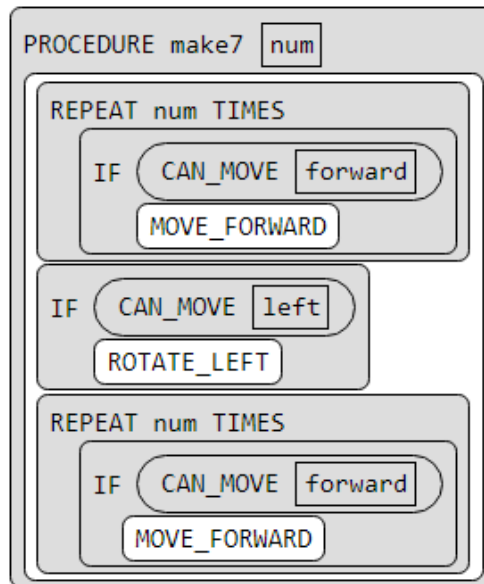
6. A gray box represents the final location of the robot after the code is executed. Which of the following represents all possible final locations?



The following question uses a robot in a grid of squares. The robot cannot move out of the grid or move into black squares. The gray box is where the robot wants to arrive. The robot can face any direction as long as it arrives at the gray square.



Read the procedure below.



7. Which of the following code segments will move the robot to the gray square?

- a. `make7 2`
- b. `make7 4`
- c. `make7 5`