CodeHS

More AP CSP Robot

Accompanying Material

Exam reference sheet: https://apcentral.collegeboard.org/pdf/ap-csp-student-task-directions.pdf?course=ap-computer-science-principles (pages 19-24)

Discussion

Since AP CSP is language agnostic, meaning it does not designate any particular programming language, it's important to understand the format and meaning of the questions on the exam. The exam reference sheet contains two programming formats: text based and block based.

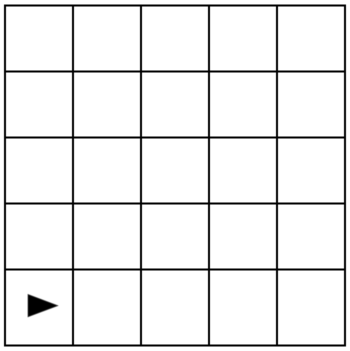
Knowing how to write code in a language agnostic manner is a great skill, too! Oftentimes potential employers during interviews want to see an algorithm, not the actual code, and these kinds of formats can be used for that. Additionally, diving into syntax without thinking through algorithms first can get us into trouble, so using this kind of coding can help with computational thinking.

This handout gives more practice on the robot-on-a-grid type problems using the coding instructions from the exam reference sheet.

Exercise

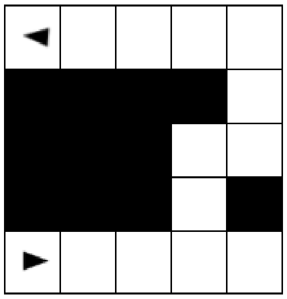
Using the AP CSP exam reference sheet, answer the following questions.

The following questions use a robot in a grid of squares. The robot is represented as a triangle, which is initially in the bottom left square of the grid and facing right when the gridworld opens (as shown below). Some problems will set up the grid world so that the robot starts in a different position, is facing a different direction and/or has squares that are blacked out as barriers to the robot's path.



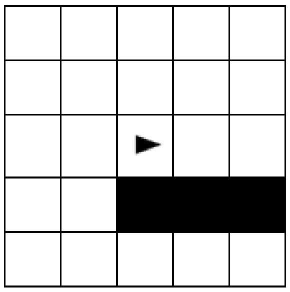
1. Consider the following maze for the robot. Write the code that will get the robot to its final position and direction. You should not repeat any code; use loops and functions instead.

Program this Maze



2. Consider the following code segment meant to guide the robot through the maze below. Why does it not work as intended for the starting grid world?

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



3. Consider the following code segment. What are the possible outcomes of the code regardless of the robot's initial position and direction?

```
REPEAT UNTIL (CAN_MOVE(forward))
{
      ROTATE_LEFT()
}
```