AP Computer Science Principles **Name: _____**
Programming Question Tips

Recall that roughly 40 percent of the questions on the AP exam will be "programming" or "algorithm" questions. These will often fall into one of four major areas detailed below:

1: "Which algorithm/code segment achieves some result?"

Example 1:

In response to an urgent memo from the Ministry of Truth, Winston Smith must change all occurrences of "Eastasia" in a given text to "Eurasia," and all occurrences of "Eurasia" to "Eastasia." He will use the fact that the word "America" does not appear anywhere in the original text.

Which of the following algorithms can be used to accomplish Winston's goal?

A. First, change all occurrences of "Eastasia" to "Eurasia."
Then, change all occurrences of "Eurasia" to "Eastasia."

B. First, change all occurrences of "Eastasia" to "Eurasia."
Then, change all occurrences of "Eurasia" to "Eastasia."
Last, change all occurrences of "America" to "Eurasia."

C. First, change all occurrences of "Eastasia" to "America."
Then, change all occurrences of "Eurasia" to "Eastasia."
Last, change all occurrences of "America" to "Eurasia."

D. First, change all occurrences of "Eastasia" to "America."
Then, change all occurrences of "America" to "Eurasia."
Last, change all occurrences of "Eurasia" to "Eastasia."

How to solve it: Run through each algorithm and see which one works. Choice C is the only one that won't fill the entire text with "Eastasia":

- Choice A changes every original instance of "Eurasia" to "Eastasia"—but after changing every instance of "Eastasia" to "Eurasia," it changes each one *back* to "Eastasia."
- Choice B does the same since the first two steps are identical to choice A—the third step is irrelevant since "America" does not appear in the text.
- Choice D changes every instance of "Eastasia" to "Eurasia"—and then changes it back to "Eastasia," along with every original instance of "Eurasia."

Thus the answer is C.

Example 2:

There are 32 students standing in a classroom. Two different algorithms are given for finding the average height of the students.

Algorithm A

Step 1: All students stand.

Step 2: A randomly selected student writes his or her height on a card and is seated.

Step 3: A randomly selected standing student adds his or her height to the value on the card, records the new value on the card, and is seated. The previous value on the card is erased.

Step 4: Repeat step 3 until no students remain standing.

Step 5: The sum on the card is divided by 32. The result is given to the teacher.

Algorithm B

Step 1: All students stand.

Step 2: Each student is given a card. Each student writes his or her height on the card.

Step 3: Standing students form random pairs at the same time. Each pair adds the numbers written on their cards and writes the result on one student's card; the other student is seated. The previous value on the card is erased.

Step 4: Repeat step 3 until one student remains standing.

Step 5: The sum on the last student's card is divided by 32. The result is given to the teacher.

Which of the following statements is true?

A. Algorithm A calculates the correct average, but Algorithm B does not.

B. Algorithm B calculates the correct average, but Algorithm A does not.

C. Both Algorithm A and Algorithm B always calculate the correct average.

D. Neither Algorithm A nor Algorithm B calculates the correct average.

How to solve it: Run through both algorithms and see if each works.

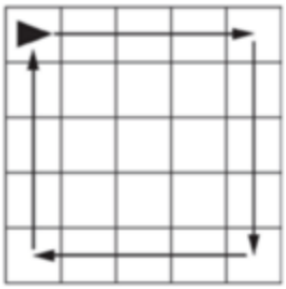
- Algorithm A works: You are finding the sum of all students' respective heights and dividing by the total number of students, which is how to find the average.
- Algorithm B is trickier—it may help to draw this one out. Though it may appear at first that some students' heights will be counted twice, this won't happen, since the same student can't partner with him- or herself. Thus Algorithm B also works.

Thus the answer is C.

You may be tempted to use sample numbers for a problem like this, but keep in mind you won't have a calculator on the exam. You have roughly 1 minute 36 seconds for each question—with 32 values and no calculator, it may be difficult to solve this question using sample numbers in that time.

Example 3:

The following question uses a robot in a grid of squares. The robot is represented as a triangle, which is initially in the top-left square of the grid and facing right.



The following program is intended to move the robot around the perimeter of the grid, as indicated in the drawing above.

```
Line 1: REPEAT 4 TIMES
Line 2: {
Line 3:      MOVE_FORWARD ()
Line 4:      ROTATE_RIGHT ()
Line 5:      MOVE_FORWARD ()
Line 6:      MOVE_FORWARD ()
Line 7:      MOVE_FORWARD ()
Line 8:      MOVE_FORWARD ()
Line 9:      ROTATE_RIGHT ()
Line 10: }
```

Which lines should be removed so the program will work as intended?

A. Line 3 and Line 4

B. Lines 5, 6, 7, 8, and 9

C. Line 8 and Line 9

D. Line 9

How to solve it: Remove the line(s) indicated by each choice and run the "program."

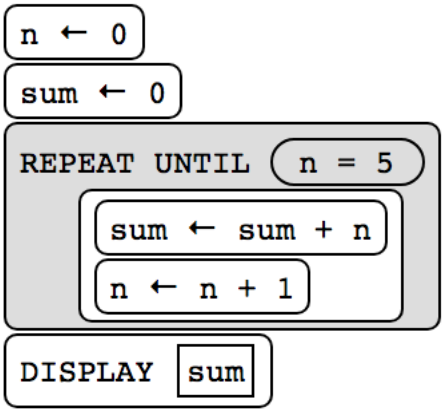
- If Lines 3 and 4 are removed, the robot will move forward four squares and turn right. As seen in the image, this is the desired result.
- If Lines 5, 6, 7, 8, and 9 are removed, the robot will move forward one square and turn.
- If Lines 8 and 9 are removed, the robot will move forward one square, turn right, move forward three squares, and then repeat (without turning right a second time).
- If Line 9 is removed, the robot will move forward one square, turn right, move forward four squares, and then repeat (without turning right a second time).

Thus the answer is A.

2: "What happens after this algorithm/code segment runs?"

Example 1:

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 10 is displayed.
C. The number 15 is displayed.
D. Nothing is displayed; the program results in an infinite loop.

How to solve it: Run through the program and see what happens.

- Start by setting `n` and `sum` to 0.
- Add the value of `n` to `sum`. The first time, the value of `n` is 0, so `sum` remains 0.
- Increment `n` by 1, so now the value of `n` is 1.
- The Boolean condition (`n = 5`) is still false, so the loop will run again. Add 1 to `sum` (which is now 1), and then increment `n` (so `n` is now 2).
- Repeat this three more times (while the value of `n` is 2, 3, and 4)—first add 2 to `sum` (making it 3), then 3 (making it 6), then 4 (making it 10).
- Increment `n`, so the value of `n` is now 5. Now the Boolean condition is true, so the loop stops.
- Display `sum`, which is 10.

Thus the answer is B.

Example 2:

A programmer has the following three procedures available:

```
PROCEDURE op1 (a, b)
{
     RETURN (a + b)
}

PROCEDURE op2 (c)
{
     RETURN (op1 (c, c))

PROCEDURE op3 (d)
{
     RETURN (d * d)
}
```

What will be displayed after the following code segment is run?

```
DISPLAY (op1 (op3 (4), 5))
```

A. 9
B. 21
C. 81
D. Nothing—the code segment results in an error.

How to solve it: Work from the inside out.

- You have `op3 (4)`—using the definition of `op3`, this returns 4 * 4, which is 16.
- Next, take `op1 (16, 5)`—using the definition of op1, this returns 16 + 5, which is 21.

Thus the answer is B.

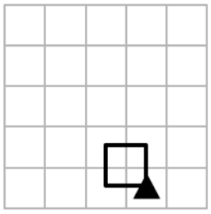
Example 3:

Consider the following code segment, which moves a robot in a grid of squares:

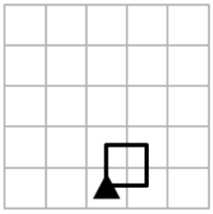
```
turn ← 0
REPEAT 4 TIMES
    MOVE_FORWARD
    IF turn = 0
        TURN_LEFT
        turn ← 1
    ELSE
        TURN_RIGHT
        turn ← 0
```

The ending position of the robot is shown in each diagram. The starting position is shown as a white triangle in cases where the robot starts and ends in different locations. Which diagram shows a possible output after the code segment is run?

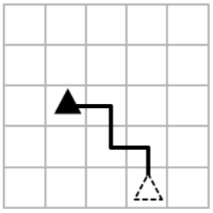
A.



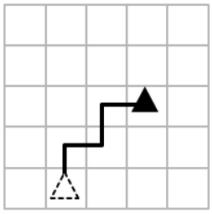
B.



C.



D.



How to solve it: Run through the code segment and see what happens. The following steps will be executed: move forward; turn left; move forward; turn right; move forward; turn left; move forward; turn right. If you draw out this sequence of moves, you will end up with something that most closely resembles the grid in choice C.

Thus the answer is C.

3: "Replace <MISSING CONDITION>"

Example 1:

Consider the following code segment:

```
IF (<MISSING CONDITION>)
{
     DISPLAY ("even")
}
```

This code segment is intended to display "even" if the positive number `num` is even. Which of the following can be used to replace `<MISSING CONDITION>` so that the code segment will work as intended?

A. `(num MOD 1) = 0`

B. `(num MOD 1) = 1`

C. `(num MOD 2) = 0`

D. `(num MOD 2) = 1`

How to solve it: Consider each of the four lines of code and determine which achieves the desired goal.

- Choice A: every whole number mod 1 = 0, so this won't tell you whether a given number is even or odd.
- Choice B: no whole number mod 1 = 1, since 1 is a factor of every whole number, so this won't tell you whether a given number is even or odd.
- Choice C: an even number mod 2 equals 0, since every even number is divisible by 2.
- Choice D: an odd number mod 2 equals 1, since no odd number is divisible by 2. This will tell you that a given number is *odd*.

Thus the answer is C.

Example 2:

According to polls, 90 percent of voters in Lee County, Alabama, are expected to write in Gus Malzahn in December's Senate special election. The code segment below is intended to calculate how many votes he can expect to receive if `n` people vote.

```
votes ← 0
REPEAT n TIMES
{
   IF (<MISSING CONDITION>)
   {
      votes ← votes + 1
   }
}
DISPLAY (sum)
```

Which of the following can be used to replace `<MISSING CONDITION>` so that the program will work as intended?

Select <u>two</u> answers.

A. `RANDOM (1, 10) = 9`

B. `RANDOM (1, 10) ≤ 9`

C. `RANDOM (1, 10) ≥ 1`

D. `NOT (RANDOM (1, 10) = 1))`

How to solve it: Find the choice(s) that cover the range of values stated in the question. In this case, you are looking for two conditions that include 90 percent of the possible values—so nine out of 10 values (this program generates a random number between 1 and 10).

- Choice A covers only one out of 10 possible values.
- Choice B covers nine out of 10 possible values. This is one answer.
- Choice C covers all ten possible values.
- Choice D covers nine out of 10 possible values (if it's not 1, there are nine other possibilities). This is the other answer.

Thus the answers are B and D.

Example 3:

Consider the following code segment:

```
n ← 0
REPEAT UNTIL <MISSING CONDITION>
{
     n ← n + 2
}
```

Which of the following replacements for `<MISSING CONDITION>` will result in an infinite loop?

A. `n = 5`

B. `n ≥ 5`

C. `n = 6`

D. `n ≥ 6`

How to solve it: Consider each of the four conditions and see which will never be achieved given the structure of the loop.

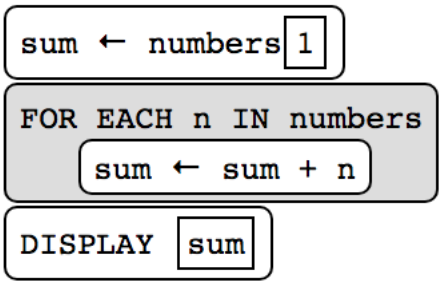
- `n` will always be even, since it starts at 0 and is always incremented by 2. Thus, the only condition that cannot be achieved is one that requires `n` to equal an odd number.
- Observe that condition A does exactly this: `n` will never equal 5, so this condition will never be met. Thus, this condition would result in an infinite loop.

Thus the answer is A.

4: "What does this program/algorithm do?"

Example 1:

Consider the following program, which uses a list of numbers called `numbers`:



For which of the following lists will this program correctly calculate the sum?

A. `[1, 2, 3, 4, 5]`
B. `[0, 2, 4, 6, 8]`
C. `[1, 3, 5, 7, 9]`
D. The program will never correctly calculate the sum of the items in `numbers`.

How to solve it: Carefully consider the code—the question all but tells you that something is wrong with it. Notice that instead of setting `sum` to 0, it sets it equal to the first element in the list…and then adds that number to `sum` anyway. Thus, the only time this program works will be when the first number in the list is 0, since this is the only way to have `sum` start at 0. Only choice B has a list that begins with 0.

Thus the answer is B.

Example 2:

Consider the following program, which is designed to find the minimum value in a list of integers named `numList`:

```
min ← −1
FOR EACH n IN numList
{
     IF (n < min)
     {
          min ← n
     }
}
DISPLAY (min)
```

Which of the following statements best describes the behavior of the program?

A. The program will always correctly calculate the minimum value in `numList`.

B. The program will never correctly calculate the minimum value in `numList`.

C. The program will correctly calculate the minimum value in `numList` only if it is negative.

D. The program will correctly calculate the minimum value in `numList` if it is not the first item.

How to solve it: Carefully consider the code, and remember the cardinal rule of finding the minimum or maximum value in an array. If you start by setting `min` equal to –1, and only change the value of `min` if it encounters a smaller value, what is the only time the value of `min` will change? That's right: if it finds another negative number. (If the minimum value in the array is –1, the final value of `min` will be –1, which is correct.)

Thus the answer is C.

Example 3:

Consider the procedure `mystery` below:

```
PROCEDURE mystery (a, b, c)
{
     IF (a ≥ b AND a ≥ c)
     {
          RETURN (a)
     }
     ELSE IF (b ≥ a AND b ≥ c)
     {
          RETURN (b)
     }
     ELSE
     {
          RETURN (c)
     }
}
```

Which of the following best describes the result of this procedure?

A. `mystery` returns the smallest of the three values `a`, `b`, and `c`.

B. `mystery` returns the largest of the three values `a`, `b`, and `c`.

C. `mystery` returns the average of the three values `a`, `b`, and `c`.

D. `mystery` always returns `c`.

How to solve it: Look at the code—this is so short it may help to assign actual values to `a`, `b`, and `c`. Suppose `a` = 5, `b` = 15, and `c` = 10.

- `a` is less than both `b` and `c`, so the first `if` condition is false. The procedure will not return `a`, which is the smallest.
- `b` is greater than both `a` and `c`, so the second `if` condition is true. The procedure will return `b`, which is the largest.
- Thus, `mystery` returns the largest of the three values.

Thus the answer is B.