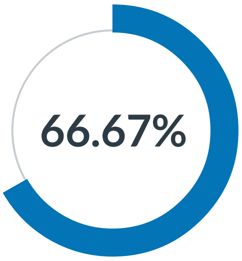
Results



4

Out of 6 points

This assessment has unlimited attempts. Take Now

Your Answers:

1 1 / 1 point

What is the output of the following recursive function when called with an argument of 5?

```
int recursive_func(int n) {
      if (n == 0)
            return 0;

      return recursive_func(n - 1) + pow(n, 2);
}
```

```
cout << recursive_func(5) << endl;
```

○ 5

✓ ● 55

○ 30

○ 14

2 0 / 1 point

What is the output of the following recursive function when called with an argument of 3?

```
int recursive_func(int n) {
      if (n == 0)
            return 0;
      if (n == 1)
            return 1;

      return recursive_func(n - 1) * recursive_func(n - 2);
}


cout << recursive_func(3) << endl;
```

○ 2

○ 5

○ 0

✗ ● 1

Correct Answer: **0**

3 1 / 1 point

What is the output of the following tail-recursive function when called with an argument of 5?

```
void print_numbers(int n) {
      if (n == 0)
            return;
      }

      cout << n << ” ”;
      print_numbers(n - 1);
}

print_numbers(5);
```

✓ ● 5 4 3 2 1

○ 5 4 3 2 1 0

○ 1 2 3 4 5

○ 0 1 2 3 4 5

4 1 / 1 point

You want to change the following recursive function to be tail recursive.

```
int recursive_func(int n) {
      if (n == 0)
            return 0;

      return recursive_func(n - 1) + pow(n, 2);
}
```

Which of the following is a possible tail recursive call?

○ acc = recursive_func(n - 1);
return acc + pow(n, 2);

○ return recursive_func(n - 1, acc) + pow(n, 2);

✓ ● return recursive_func(n - 1, acc + pow(n, 2));

○ return recursive_func(n - 1) + pow(n, 2);

5

1 / 1 point

Which of the following is true of tail recursion?

✓ ● Tail recursion uses less memory than non-tail recursion.

○ Tail recursion is easier to implement than non-tail recursion.

○ Tail recursion is faster than non-tail recursion.

○ There is no benefit to tail recursion, it is just an alternative way to write the code

6

0 / 1 point

You come across the following (incorrect) attempt at writing tail recursive code:

```
int sum(int n) {
    static int acc = 1;
    if (n==1)
        return acc;

    acc = acc + n;
    return sum(n - 1);
}
```

Which statement is true of this code?

✕ ● This code will never generate the correct answer.

Correct Answer: **This code will generate the correct result the first time it is called but will return the incorrect result in subsequent calls**

○ This code will always generate the correct result.

○ This code will generate the correct result the first time it is called but will return the incorrect result in subsequent calls

○ This code contains syntax errors that will prevent it from compiling