Computational Thinking with Algorithms Problem Sheet (Java)

Question 1 (3 marks)

Consider the following method:

```
public static void mystery (int n) {
    System.out.print(n);
    if (n < 4) {
        mystery(n+1);
    }
    System.out.print(n);
}
```

What will the output of the call `mystery(1);` be?

Write an explanation of the reasoning behind your answer, using the aid of either a recursion trace diagram or a stack diagram. Include any code which you write for testing or explanation purposes as part of your answer.

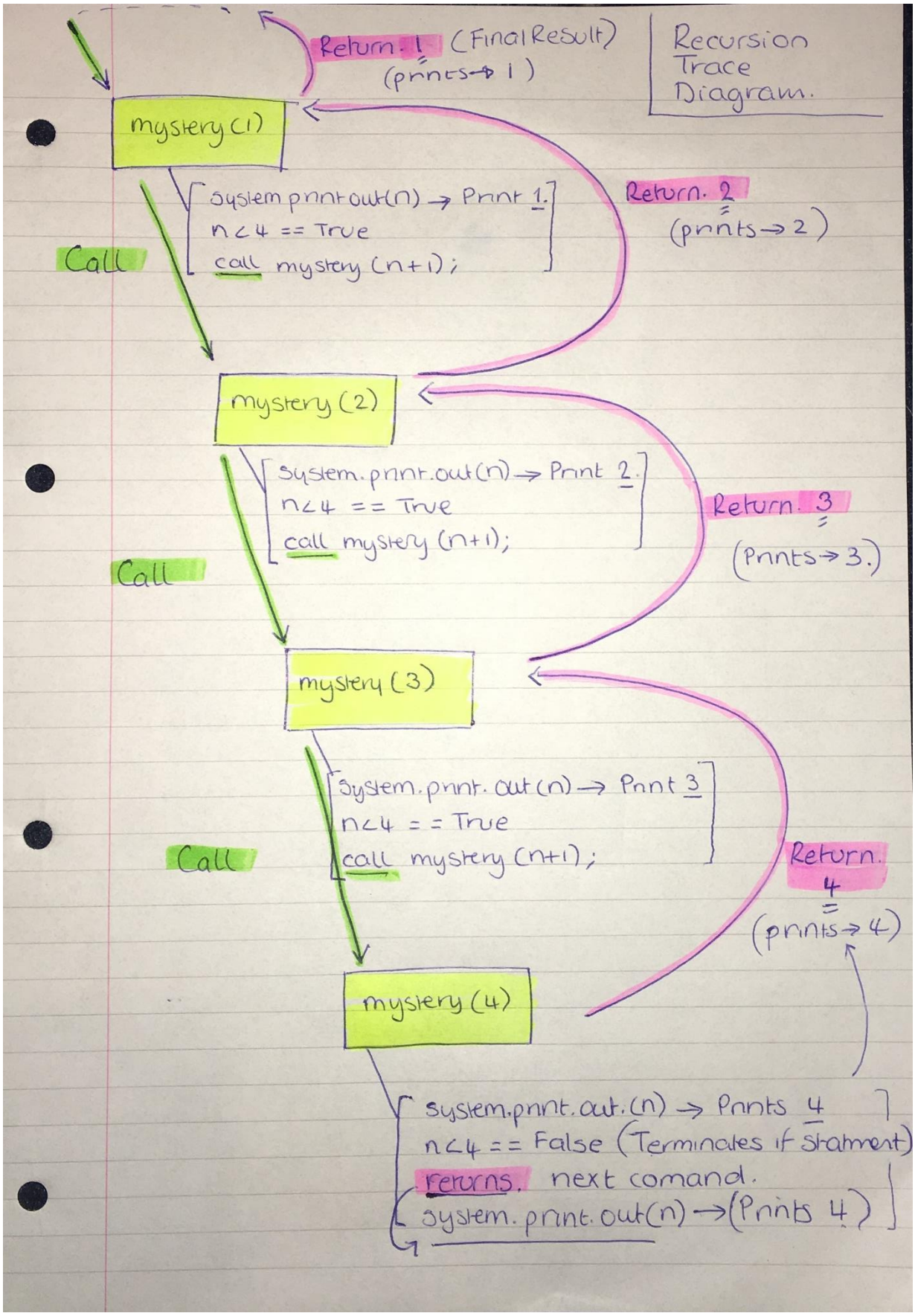
Output:

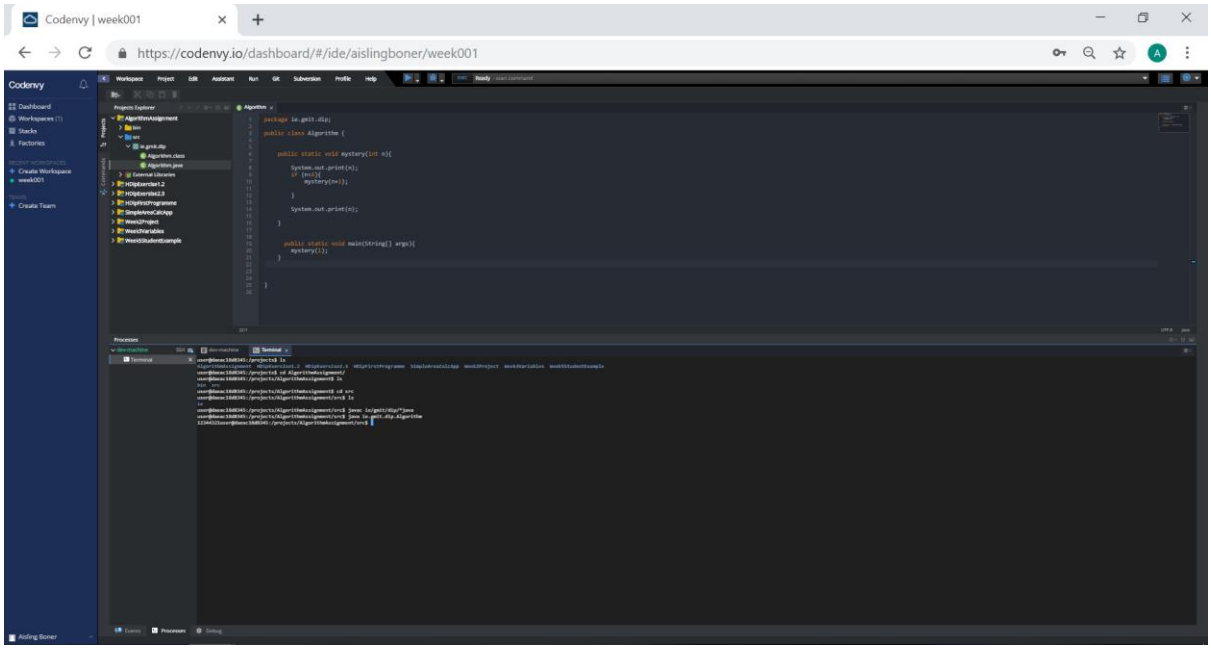
1
2
3
4
4
3
2
1

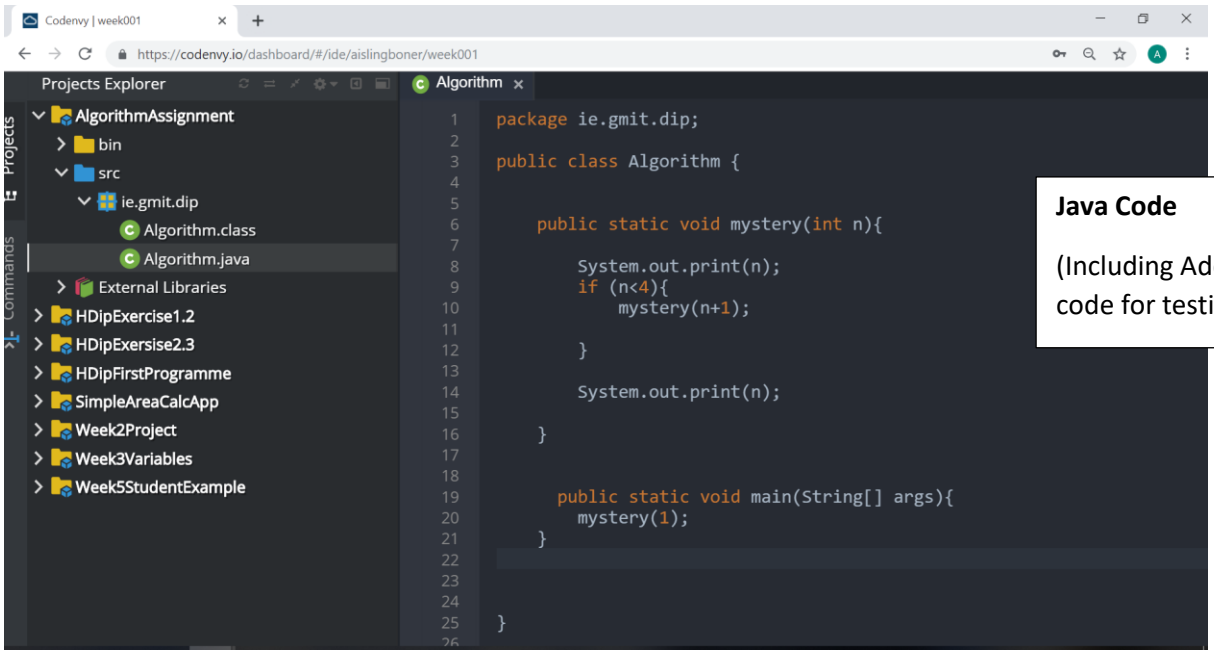
This Explained:

1. Firstly mystery (1); is inputted through the mystery method.
2. If n of mystery(n); satisfies the n<4 parameter of the recursive call (if statement), the function mystery(n+1); is invoked.
3. This results in the addition of n + 1 which can be clearly seen by the first print output of 1,2,3 & 4.
4. However once n=4 (mystery(4);), the output from the if statement can no longer be called as n is not less then 4 i.e. where n=4, n!<4 (4<4). This if statement return a false, terminating the original if statement and moving to the next command.
5. Below the recursive call exists a second print statement i.e. 'System.out.print(n)'.
6. This second print statement returns the recursive call, giving an output to the previous function it came from i.e. n=4.
7. Then the function returns to the previous call, mystery(3), it prints and returns to the next previous call i.e. mystery(2) and so on until it has returned to mystery(1) whereby the program exits.

This can be seen below in my explanation using a recusion trace drawing of the question:

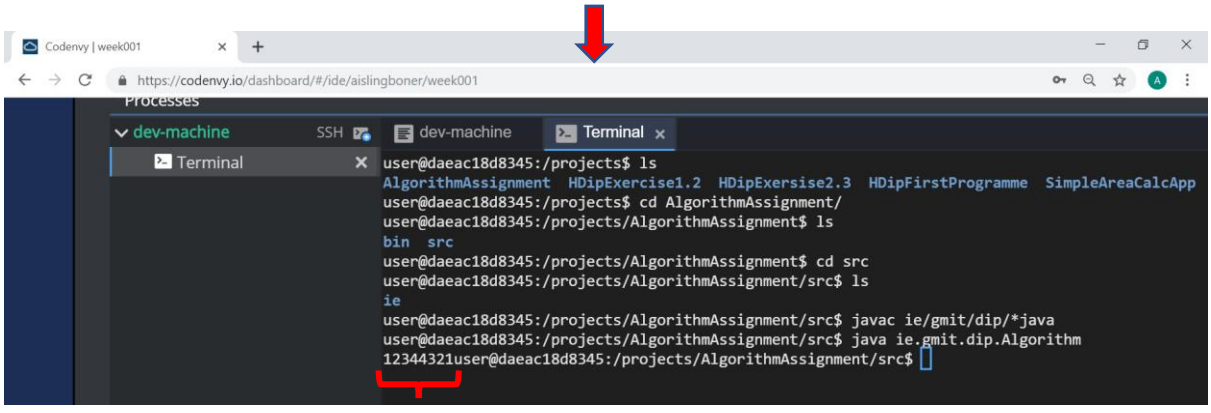






Java Code

(Including Additional code for testing):



Output