

ASSIGNMENT 07: RECURSION

Task 01: For this recursive code

```
public static int fun1( int x, int y) {  
    if (x == 0)  
        return y;  
    else  
        return fun1(x - 1, x + y);  
}
```

a) Post results for the following values of x and y

b) Draw recursive tree

1) (0,0)

a) 0, 0

2) (-10,0)

a) *Stack Overflow: Runs infinitely since x can never reach 0*

3) (0,-7)

a) 0, -7

b) 0, -7 → fun1(0, -7)

4) (1,5)

a) 0, 6

b) 1, 5 → fun1(1, 5)

fun1(0, 6)

5) (10,5)

a) 0, 60

b) 10, 5 →

fun1(10, 5)

fun1(9, 15)

fun1(8, 24)

fun1(7, 32)

fun1(6, 39)

fun1(5, 45)

fun1(4, 50)

fun1(3, 54)

fun1(2, 57)

fun1(1, 59)

fun1(0, 60)

c) What does the recursive code do?

- *It checks if X has reached 0 in order to break the loop*
- *Otherwise, for each pass it will pause the execution to spawn a new stack which will pass the next call as $x = x-1$, $y = x+y$ (note that x is not decremented when added to y)*
- *Once it reaches $x = 0$ it will resume execution of each stack from the end towards the front*
- *Since the executed statement is a *return* it had nothing else to do, so the stack is closed immediately after its child has finished executing*

Task 02:

Write a recursive function to find the first lower case in a string, it should return the index of the first lower case letter in an input string or -1 if there wasn't any.

GitHub Link: [RecursionTest.java](#)

Task 03:

Create a profile on HackerRank & Solve 2 challenges: Completed

- A) [Profile](#)
- B) [Arrays – DS](#)
- C) [Dynamic Arrays](#)

Task 04:

Implement and measure the performance of 4 sorting algorithms:

A) [SortingAlgorithms.java](#)

B) Measured performance of sorting algorithms (in milliseconds)

Array Length	Bubble Sort	Insertion Sort	Selection Sort	Merge Sort
500	14ms	3ms	1ms	0ms
1,500	11ms	6ms	8ms	0ms
30,000	3545ms	2539ms	2174ms	7ms
50,000	10396ms	7658ms	6163ms	13ms
100,000	44895ms	27073ms	28586ms	21ms