American University of Beirut
Department of Computer Science
CMPS 200 – Spring 2022
Deadline: Friday, April 8 @11:00 PM

Exercise 1. Random walk

You may know that the streets and avenues of Manhattan form a grid. A random walk through such a grid is a walk in which a random direction (N, E, S, or W) is chosen with equal probability at every intersection. For example, a random walk on a 5 x 11 grid starting at the center (2, 5) could visit grid points (2, 6), (2, 7), (2, 8), (2, 9), (2, 10), back to (2, 9) and then back to (2, 10) before leaving the grid.

Write function manhattan() that takes the number of rows and columns in the grid, simulates a random walk starting in the center of the grid, and computes the number of times each intersection has been visited by the random walk.

Your function should print the table line by line once the random walk moves outside the grid. As an example, the call manhattan(5, 11) could produce the following output:

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 2]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Exercise 2. GCD

The greatest common divisor (**gcd**) of two positive integers is the largest integer that divides evenly into both of them. For example, gcd(102, 68) = 34. We can efficiently compute the gcd using the following property, which holds for positive integers p and q:

- If p > q, the gcd of p and q is the same as the gcd of q and p % q.
- The gcd of (p, 0) is p.

Write a **recursive** function gcd(p, q) that returns the greatest common divisor and test the function on a few examples of your choice.

Exercise 3. String Repetition

Write a recursive method repeat that accepts a string s and an integer n as parameters and that returns a String consisting of n copies of s.
For example:
repeat("hello", 3) → "hellohellohello"
repeat("this is fun", 1) → "this is fun"
repeat("wow", 0) → ""

Exercise 4. Remove Repetition

Write a recursive method that takes a string as a parameter and that returns a new string obtained by replacing every sequence of repeated adjacent letters with just one of that letter. For example, the string "Boo" has three repeated adjacent letters ("oo", "kkkkk", and "ee"), so the call of dedup("bookkkkkeeper") should return "bokeper".

Exercise 5. Multiply Even

Write a method multiplyEvens that returns the product of the first n even integers.

For example:

```
multiplyEvens(1); 2 (2)
multiplyEvens(2); 8 (2 * 4)
multiplyEvens(3); 48 (2 * 4 * 6 = 48)
multiplyEvens(4); 384 (2 * 4 * 6 * 8 = 384)
```

Exercise 6. Different Ways

Write a **RECURSIVE** function **differentWays** that takes two integers ***n*** and ***x***, and returns the number of ways we can choose x different values out of n numbers.

Note that, choosing x elements out of n can be done:

- either by choosing all the x from the n elements excluding the last one
- or by choosing x-1 elements from the n elements excluding the last one and then adding the last element.

(You are not allowed to use the formula (using factorial) to do this calculation. You have to write your own code following the previous guidelines).

For example:

```
System.out.println(differentWays(10,  4)) → 210
System.out.println(differentWays(10,  3)) → 120
System.out.println(differentWays(10,  10)) → 1
System.out.println(differentWays(10,  1)) → 10
```