

**American University of Armenia, CSE**  
**CS121 Data Structures A, C**  
**Fall 2021**

**Homework Assignment 1**

Due Date: Monday, September 13 by 23:59 electronically on Moodle

*Please solve the programming tasks either in Java or C++, following good coding practices (details are on Moodle).*

**1. (9 points)** For the code below:

```
1 for (int i = 0; i < a.length; i++) {
2     if (a[i] < 0)
3         continue;
4     if (a[i] != 0)
5         a[i] = a[i] * (-1);
6 }
```

- (a) Give the total number of element comparisons (i.e. involving an element of the array) as a function of the length of the array (not Big-O notation).
- (b) Can the algorithm be improved in terms of the number of comparisons? If so, show how and give the total number of element comparisons in the best and worst case analysis for your improved version of the algorithm.
- (c) Give the Big-O estimate of the running time of the original code and the improved version from (b).

**Briefly justify your answers for all three parts.**

**2. (12 points)** Order the following functions by asymptotic growth rate.

$48n + 5 \log n$	$2n^5$	$3n + 2n \log n$
$4^n$	$10n$	$1 + \log n$
$-8n + 2n^3$	$2100 \times 2^{110}$	$2n^3$
$\left(\frac{9}{4}\right)^n$	$60 \log n + 10n$	$17n^3 + n^4$

- 3. (19 points)** Write a **recursive** program that inputs a natural number  $n$  and an array  $A$  of  $n$  decimal digits, determines and outputs the number of all length-2 or length-3 subarrays of  $A$  that form a palindrome. *Inefficient solutions will receive partial credit.*
- 4. (20 points)** Write a **recursive** program that, given a natural number  $n$ , determines and outputs all the possible strings of length  $n$ , s.t. they can contain only the letters **a**, **b**, **c**, **d**, **e**, and consecutive letters cannot be both vowels or both consonants. *Inefficient solutions will receive partial credit.*

5. (20 points) Write a **recursive** program that inputs a natural number  $n$  and prints an isosceles triangle of asterisks (the symbol `*`) with height  $n$ . Note that your program is not allowed to use arrays, strings or loops; it should directly generate the output in a recursive manner.

For example, when  $n = 5$ , your program should print:

```

      *
     * *
    *  *
   *   *
  *    *
*****

```

6. (20 points) Consider a rectangular maze of size  $n \times m$  and a player is trying to find the exit in it. An available position is denoted with '`.`', a wall with '`X`', the exit position with '`E`'. The player can move up, down, left or right one position at every step but cannot leave the boundaries of the maze.

Write a **recursive** program that determines if it is possible to exit the maze from the given initial position of the player. The input consists of the maze dimensions  $n$  and  $m$  and the coordinates of the initial position of the player, followed by  $n$  lines of  $m$  characters each, describing the maze. The output is '`Yes`' or '`No`' to indicate whether the exit is reachable or not.

sample input	sample output
4 5 0 3 .X..X ..X.. .X..X X...E	Yes