INSTITUTO UNIVERSITÁRIO DE LISBOA

Introdução à Programação
1st semester - 2023/2024

Midterm Test

Number:						

Name:	

Duration: 1h30 Grading: 5 val.

Rules and Observations

- The answers are to be written in this document, the test can be written in pencil.
- It is strictly prohibited to manipulate any electronic devices (cell phones, tablets, etc.). Violation of this rule will result in cancellation of the test.
- No questions will be answered after 15 minutes from the start of the test.
- It is only permitted to leave the classroom 45 minutes after the start of the test.
- If you are taking the test online, you must take a photo and upload it to Moodle at the end.

Question 1 (1,0 val.)

1.1. Develop a function that, for a given a character, returns true if that character is a letter in the alphabet (both uppercase and lowercase) and false if it is any other character.

```
isAlphabetLetter('A') -> true     isAlphabetLetter('w') -> true
isAlphabetLetter('2') -> false    isAlphabetLetter('*') -> false
```

```
static boolean isAlphabetLetter(char c) {
```

1.2. Develop a function that, for a given array of characters, returns the index of the first character that is not part of the alphabet. If all characters belong to the alphabet, it should return -1.

Use the function developed in the previous paragraph, if you haven't done so, please assume it is available.

```
indexOfFirstNonAlphabetLetter({'a', '1', 'c'}) -> 1
indexOfFirstNonAlphabetLetter({'?', 'B', '#'}) -> 0
indexOfFirstNonAlphabetLetter({'g', 'F', 'K'}) -> -1
```

(assume a non-zero input array)

Question 2 (0,75 val.)

Develop a function that, given an array of characters, returns the percentage of characters in the array that are not letters of the alphabet (upper or lower case). The result must be a decimal number contained in the range [0, 1].

Use the function developed in the previous question, if you didn't do it, assume it is available.

```
nonLetterPercentage({'s', '1', '~'}) -> 0.66
nonLetterPercentage({‘G’, ‘J’, 'R', 'r', '%'}) -> 0.2
```

(assume a non-zero input array)

Question 3 (1,5 val.)

3.1. Develop a recursive function that returns the number of digits in a given positive integer.

```
numDigits(2) -> 1
numDigits(2023) -> 4
```

(assume the given value is greater than or equal to zero)

```
static int numDigits (int n){
```

3.2. Develop a procedure that, for a given array of integers, replaces each value with the number of digits of each value. If you have not developed the previous paragraph, assume that the function is available.

```
countDigits({2023, 82, 9, 123}) -> {4, 2, 1, 3}
countDigits({133, 0, 43}) -> {3, 1, 2}
```

(assume the values in the given vector are greater than or equal to zero)

Question 4 (1,0 val.)

Develop a function that receives two arrays of integers and returns an integer array. The returned array is made up of the numbers from the first vector that are in the indexes defined by the values of the second vector.

Given the array {5, 2, 3, 7, 8, 3,1,10} and the indexes {2, 4, 6} the function must return a new array with the values found in indices 2, 4 and 6 : {3, 8, 1}.

```
getVector({5, 2, 3, 7, 8, 3, 1, 10}, {2, 4, 6}) –> {3, 8, 1}
getVector({2, 10, 3, 0, -18 }, {0, 2, 4}) –> {2, 3, -18}
```

(assume the arrays are not null and that the indexes indicated in the second vector are valid indices of the first vector)

```
static int[] getVector (int[] v, int[] indexes){
```

Question 5 (0,75 val.)

Develop a function that indicates the size of the largest row in an irregular matrix.

```
maxRowSize({{5}, {2, 3, 7}, {8, 3}, {1, 10}, {6}}) -> 3
maxRowSize({{0, 2}, {5, 77, 58, -12}}) -> 4
```

(assume that the matrix is not null and that no row in the matrix is null)