INSTITUTO UNIVERSITÁRIO DE LISBOA

Introdução à Programação
1º semestre - 2023/2024

Teste Intercalar

Número:						

Nome:	

Duração: 1h30 Cotação: 5 valores

Observações e Regras

- As respostas são para escrever no próprio enunciado sendo permitido entregar o teste escrito a lápis.
- É estritamente proibido manipular quaisquer aparelhos eletrónicos (telemóveis, tablets, etc). A violação desta regra implica a anulação do teste.
- Não serão respondidas questões após 15 minutos do início do teste.
- Só é permitido sair após 45 minutos do início do teste.
- Caso esteja a fazer o teste online deve fotografar e fazer upload no *moodle* no final.

Questão 1 (1,0 valor)

1.1. Desenvolva uma função que dado um caracter devolve verdadeiro caso esse caracter seja uma letra presente no alfabeto (tanto para maiúsculas como para minúsculas) e falso caso seja qualquer outro caracter.

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

static boolean isAlphabetLetter(char c) {
```

1.2. Desenvolva uma função que dado um vetor de caracteres devolve o índice do primeiro caracter que não faça parte do alfabeto. No caso de todos os caracteres pertencerem ao alfabeto deve devolver -1.

Use a função desenvolvida na alínea anterior, se não a fez assuma que está disponível.

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

(assuma um vector de entrada não nulo)

Questão 2 (0,75 valor)

Desenvolva uma função que dado um vector de caracteres, devolve a percentagem de caracteres do vector que não são letras do alfabeto (maiúsculas ou minúsculas). O resultado deve ser um número decimal contido no intervalo [0, 1].

Use a função desenvolvida na pergunta anterior, se não a fez assuma que está disponível.

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

(assuma um vector de entrada não nulo)

Questão 3 (1,5 valores)

3.1. Desenvolva uma função recursiva que devolve o número de dígitos de um número inteiro positivo.

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

(assuma que o valor dado é maior ou igual a zero)

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

3.2. Desenvolva um procedimento que dado um vetor de inteiros, substitui cada valor pelo número de dígitos do mesmo. Caso não tenha desenvolvido a alínea anterior, assuma que a função está disponível.

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

(assuma que os valores no vector dado são maiores ou iguais a zero)

Questão 4 (1,0 valores)

Desenvolva uma função que recebe dois vectores de inteiros e devolve um vector de inteiros. O vector devolvido é constituído pelos números do primeiro vector que se encontram nos índices definidos pelos valores do segundo vector.

Dado o vector {5, 2, 3, 7, 8, 3 ,1 ,10} e os índices {2, 4, 6} a função deve devolver um novo vector com os valores que se encontram nos índices 2, 4 e 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}
```

(assuma que os vectores não são nulos e que os índices indicados no segundo vector são índices válidos do primeiro vector)

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

Questão 5 (0,75 valor)

Desenvolva uma função que indica o tamanho da maior linha numa matriz irregular.

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

(assuma que a matriz não é nula e que nenhuma linha da matriz é nula)