

# UNIVERSIDADE DA BEIRA INTERIOR

Programação – LEI

1º Semestre

Frequência 2 (8 valores)

Resolução

19/12/2023

## 1. [1.25 val]

Considere as seguintes declarações de variáveis:

```
int **V, *W, *X;
```

e que `sizeof(int) = 4` e `sizeof(int *) = 8`.

(esquema de bloco de memória)

	...	
100500	110172	<b>W</b>
	...	
100800	110160	<b>X</b>
	...	
110160	50	
110164	10	
110168	15	
110172	60	
110176	70	
110180	80	
	...	
110500	100800	<b>V</b>
	...	

Considerando os valores que constam no esquema de um bloco de memória que se encontra ao lado, determine o valor de cada uma das seguintes expressões:

a)  $X + 2 = 110160 + 2 \times \text{sizeof}(\text{int}) = 110160 + 2 \times 4 = \mathbf{110168}$

b)  $V[0] = \mathbf{110160}$

c)  $W + 4 = 110172 + 4 \times \text{sizeof}(\text{int}) = 110172 + 4 \times 4 = \mathbf{110188}$

d)  $(*V)[2] = \mathbf{15}$

e)  $V + 2 = 100800 + 2 \times \text{sizeof}(\text{int}*) = 100800 + 2 \times 8 = \mathbf{100816}$

f)  $*V + 3 = 110160 + 3 \times \text{sizeof}(\text{int}) = 110160 + 3 \times 4 = \mathbf{110172}$

g)  $**V + 4 = 50 + 4 = \mathbf{54}$

h)  $*W + 4 = 60 + 4 = \mathbf{64}$

i)  $*(W - 2) = *(110172 - 2 \times \text{sizeof}(\text{int})) = *(110172 - 2 \times 4) = *(110164) = \mathbf{10}$

j)  $*(X + 3) = *(110160 + 3 \times \text{sizeof}(\text{int})) = *(110160 + 3 \times 4) = *(110172) = \mathbf{60}$

**NOTA:** apresente todos os cálculos efetuados

## 2. [1.75 val]

Implemente um **subprograma** em C que dado um número inteiro positivo **N** ( $N > 0$  é parâmetro do subprograma), **devolva** a quantidade de divisores de N. Exemplo: divisores de 10 = 4 ( $\{1, 2, 5, 10\}$ ).

**int subprograma1 (int N)**

```
{
    int k, cont;
    cont = 0;
    for (k = 1; k <= N; k = k + 1)
        if (N % k == 0)
            cont = cont + 1;
    return cont;
}
```

**void subprograma2 (int N, int \*cont)**

```
{
    int k;
    *cont = 0;
    for (k = 1; k <= N; k = k + 1)
        if (N % k == 0)
            *cont = *cont + 1;
}
```

---

---

### 3. [2.50 val]

Considere o ficheiro de texto "Inteiros.txt", apenas com números inteiros. Implemente um **programa em C** que realize as seguintes ações (pela ordem indicada):

- 1) construir um array de 1 dimensão apenas com os números positivos não nulos ( $> 0$ ) do ficheiro "Inteiros.txt", usando gestão de memória dinâmica, e
- 2) guarde os números pares do array no ficheiro de texto "Pares.txt" e os números ímpares do array no ficheiro "Impares.txt".

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int *A, tamA, num, k;
    FILE *f, *g;

    // 1)
    f = fopen("Inteiros.txt", "r");
    tamA = 0;
    A = (int*) malloc(tamA * sizeof(int));
    while (fscanf (f, "%d", &num) == 1)
    {
        if (num > 0) {
            tamA = tamA + 1;
            A = (int*) realloc(A, tamA * sizeof(int));
            A[tamA-1] = num;
        }
    }
    fclose(f);

    // 2)
    f = fopen("Pares.txt", "w");
    g = fopen("Impares.txt", "w");
    for (k = 0; k < tamA; k = k + 1)
    {
        if (A[k] % 2 == 0)
            fprintf(f, "%d\n", A[k]);
        else
            fprintf(g, "%d\n", A[k]);
    }
    fclose(f);
    fclose(g);
}
```

---

---

#### 4. [2.50 val]

Implemente um **subprograma** em C que dado um array de 1 dimensão **X** com **N** ( $N > 0$ ) números inteiros positivos ( $X$  e  $N$  são parâmetros do subprograma), **determine e devolva** como resultados o **menor** número e o **segundo menor** número do array **X** (se todos os números de  $X$  são iguais, então não há segundo menor número e deve devolver -1 para indicar este facto => segundo menor = -1).

**NOTA: o array só pode ser percorrido uma única vez.**

```
int subprograma (int *X, int N, int *segMenor)
{
    int k, menor;
    menor = X[0];
    *segMenor = -1;
    for (k = 1; k < N; k = k + 1)
    {
        if (A[k] < menor)
        {
            *segMenor = menor;
            menor = A[k];
        }
        else // A[k] >= menor
            if (A[k] > menor && (*segMenor == -1 || A[k] < *segMenor) )
                *segMenor = A[k];
    }
    return menor;
}
```