

UNIVERSIDADE DA BEIRA INTERIOR

Programação – LEI + LMA

1º Semestre

Frequência 2 (8 valores)

1:15 h + 10 min

04/01/2023

1. [1.00 val]

Considere as seguintes declarações de variáveis:

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

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

(esquema de um bloco de memória)

	...	
100500	110160	W
	...	
100700	110168	X
	...	
110160	50	
110164	100	
110168	150	
110172	600	
110176	700	
110180	800	
	...	
110500	100700	V
	...	

Usando os valores que constam no esquema de um bloco de memória que se encontra ao lado, indique os valores de cada uma das seguintes expressões:

- a) `&X`
- b) `X[1]`
- c) `W + 4`
- d) `W[4]`
- e) `*W + 4`
- f) `*(W + 2)`
- g) `*V + 4`
- h) `V + 4`

NOTA: apresente todos os cálculos efetuados

2. [1.75 val]

Implementar um **subprograma** em C que dado um número inteiro positivo **N** (parâmetro do subprograma), peça ao utilizador para inserir **N** números reais e **determine e devolva** como resultados a **soma** dos números **positivos** inseridos e a quantidade de números **positivos** inseridos.

3. [2.00 val]

Implementar um **subprograma** em C que dados um array (de 1 dimensão) **X** com **N** números reais (**X** e **N** são parâmetros do subprograma), **determine e devolva** o **menor** número **positivo de X**. Caso o array **X** não contenha números positivos, o subprograma deve devolver um número negativo.

4. [1.75 val]

Considere o tipo de dados seguinte (registo/estrutura):

```
typedef struct {  
    int numAluno;  
    int notaFinal;  
    int aprovado; // 1 = aprovado, 0 = reprovado  
} ALUNO;
```

Considere também o seguinte subprograma já implementado (incluído na biblioteca **Frequencia2.h**):

```
void lerArray1DALUNO (ALUNO A[], int *N)  
    // constrói um array A com "N" elementos do tipo ALUNO, mas apenas com os campos numAluno  
    e notaFinal de cada elemento com valores atribuídos (o campo aprovado não tem valor atribuído)  
void mostrarArray1DALUNO (ALUNO A[], int N)  
    // mostra no monitor os N elementos do tipo ALUNO do array A
```

Implemente um **programa em C** que realize as seguintes ações (pela ordem indicada):

- ler um valor inteiro TAM, em que $0 < \text{TAM} \leq 100$
- ler um array **A** de registos do tipo **ALUNO** com **TAM** elementos (usar subprograma dado)
- atribuir ao campo **aprovado** de cada um dos elementos do array **A** o valor **1** (se $\text{notaFinal} \geq 10$) ou **0** (se $\text{notaFinal} < 10$)
- mostrar todos os elementos do array A (usar subprograma dado)

5. [1.50 val]

Considere-se um ficheiro de texto de nome "entrada.txt", no qual cada uma das suas linhas contém 2 números: um inteiro e um real (por esta ordem).

Construa um **programa em C** que leia os números de cada uma das linhas do ficheiro "entrada.txt" e guarde-os no ficheiro de texto de nome "saida.txt", mas apenas se o **número inteiro** for **positivo** (> 0).

NOTAS:

- 1. Não pode usar arrays**
 - 2. Na abertura dos ficheiros não é necessário verificar se a houve algum erro na abertura**
-
-