UNIVERSIDADE DA BEIRA INTERIOR

Programação – LEI **1º Semestre**

Exame Época Normal (16 val) **2h + 15 min** **2023/2024**

1. [1.50 val]

Escreva uma **expressão lógica em linguagem C** para a seguinte condição:

a) o valor da variável do tipo inteiro **N** é um número ímpar, mas não está entre 100 e 500

b) a soma dos valores das variáveis do tipo inteiro **N** e **M** é nula (zero), mas o valor de **N** é positivo

c) o valor da variável do tipo inteiro **N** é múltiplo de 5, mas não é divisor de 20

Escreva uma **instrução de atribuição em linguagem C** para cada uma das seguintes acções:

d) a variável do tipo inteiro **A** é decrementada em 10 valores

e) a variável do tipo inteiro **ARRED** recebe o valor arredondado da variável do tipo real **X**

f) a variável do tipo inteiro **P** recebe o valor **10** se **N** for **par** ou **5** se **N** for **ímpar**

g) a variável do tipo inteiro **SOMA** recebe o valor da soma do algarismo das **centenas** da variável **A** com o algarismo das **unidades** da variável **B** (A e B são do tipo inteiro)

2. [2.50 val]

Usando um fluxograma, construa um algoritmo que realize as seguintes ações (pela ordem indicada):

1. peça ao utilizador que insira (leia) dois números inteiros positivos M e N, em que M < N
2. determine a quantidade de divisores de N entre M e N, incluindo eles próprios
3. mostre o resultado obtido (quantidade) com uma mensagem adequada.

3. [2.50 val]

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

1. peça ao utilizador que insira e leia dois números inteiros positivos não nulos **M** e **N**, em que M < N (M, N > 0 e M < N)
2. insira e leia **N** números inteiros positivos não nulos (> 0) e determine a **quantidade** destes números que são **divisores** de **M** e a **quantidade** destes números que são **múltiplos** de **M**
3. mostre os resultados obtidos no passo anterior (quantidade de divisores e quantidade de múltiplos) com uma mensagem adequada.

4. [2.25 val]

Implemente um **subprograma em C** que dado um array de 1 dimensão **X** com **N** números inteiros (X e N > 0 são parâmetros do subprograma), **devolva** como resultados a **quantidade** de elementos de **X** que são positivos (> 0) e a **média** desses elementos (dos positivos).

5. [1.50 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	**V**
	...	
100800	110180	**W**
	...	
110160	10	
110164	20	
110168	30	
110172	40	
110176	50	
110180	60	
	...	
110500	100800	**X**
	...	

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 (**apresentar todos os cálculos efetuados**):

a) X + 2

b) V[0]

c) W + 4

d) (*X)[0]

e) V + 2

f) *V + 3

g) **X + 2

h) X[0] + 2

i) *(W - 3)

j) *(*X - 4)

NOTA: se o valor da expressão não existir ou for desconhecido, então responder "**Indefinido**"

6. [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) **construa** um **array de 1 dimensão** com todos os números do ficheiro "Inteiros.txt", usando gestão de **memória dinâmica**,
2) **determine** a **quantidade** de números positivos (> 0) e a **quantidade** de números negativos (< 0) do array construído em 1),
3) **guarde** no ficheiro de texto "Saida.txt" os resultados obtidos no passo anterior.

NOTA: não é preciso verificar se há **ERRO** na **abertura dos ficheiros**, nem na **alocação de memória**

7. [3.25 val]

Implemente um **subprograma em C** que dado um **array** de 1 dimensão **X** com **N** (N > 0) números inteiros (X e N são parâmetros do subprograma), **remova** do array todos os elementos repetidos e mantendo a ordem pela qual se encontram no array, tendo em conta que o array foi construído usando memória dinâmica.

NOTA: não pode usar outros arrays nem ficheiros, e deve percorrer o array o número mínimo de vezes.

Exemplo:

A = [3 -5 3 8 3 4 8 6 -2 -5] ==> A = [3 -5 8 4 6 -2]