

UNIVERSIDADE DA BEIRA INTERIOR

Programação – LEI

1º Semestre

Exame Época Recurso

2 h + 15 min

2024-2025

1.

Escreva uma **expressão lógica** em linguagem C para as seguintes condições:

- a) o valor da variável do tipo real **A não pertence** ao intervalo **[-50, 50[**
- b) o valor da variável do tipo inteiro **B é positivo e não pertence** ao conjunto **{ 100, ..., 999 }**

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

- c) a variável **B** recebe o **valor** do número composto pelos **três últimos algarismos** do valor (número) da variável do tipo inteiro **A** (exemplo: $A = 23542 \Rightarrow B = 542$)
- d) a variável **P** recebe o valor **5** se o valor da variável **N** for **par** e o valor **9** se **N** for **ímpar** (P e N são variáveis do tipo inteiro)

Supondo que **X = 12**, **Y = 2** e **Z = 6** (X, Y e Z são variáveis do tipo inteiro), indique a ordem de cálculo dos operadores e determine o valor de cada uma das seguintes expressões. Apresente todos os cálculos.

- e) $(Y + 5 * Z) < X / (Z \% 4)$
1 2 3 4 5
- f) $(3 + 2.5) * Y - (24 \% (X + Z))$
1 2 3 4 5

Sugestão de resposta (exemplo):

- ordem de cálculo: 5, 3, 2, 1, 4
- valor: 34.0 (e apresentar cálculos)

2.

Construa um algoritmo (**sem usar arrays**), através de um fluxograma, que:

- **peça** ao utilizador e **insira/leia** dois números **reais, X e Y**, com valores no intervalo **[15.5, 60.5]**
- **determine** os valores de **min** = menor(X, Y) e **max** = maior(X, Y)
- **peça** ao utilizador e **insira/leia 20** números **reais** e **determine** a **soma** dos números inseridos que **pertencem** ao intervalo **[min, max]**
- **mostre** o **resultado obtido** no passo anterior (**soma**).

3.

Construa um **programa em C (sem usar arrays)** que:

- **peça** ao utilizador e **insira/leia** dois números **reais, X e Y**, com valores no intervalo **[15.5, 60.5]**
- **determine** os valores de **min** = menor(X, Y) e **max** = maior(X, Y)
- **peça** ao utilizador e **insira/leia 20** números **reais** e **determine** a **soma** dos números inseridos que **pertencem** ao intervalo **[min, max]**
- **mostre** o **resultado obtido** no passo anterior (**soma**).

4.

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 como resultados** o **produto** entre todos os números **positivos não nulos (> 0) de X** e a **soma** dos números **negativos de X**.

NOTA: não pode usar outros arrays nem estruturas.

5.

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)

	...	
110040	110252	X
	...	
110120	110256	V
	...	
110244	90	
110248	80	
110252	70	
110256	60	
110260	50	
110264	40	
	...	
110500	110120	w
	...	

Usando os valores contidos no esquema de um bloco de memória dado ao lado, indique, justificando, os valores de cada uma das seguintes expressões:

- a) $*(V - 2)$
- b) $V + 5$
- c) $\&V[1]$
- d) $*(X - 2)$
- e) $**W + 4$
- f) $W + 4$
- g) $*(*W + 2)$
- h) $*W - 2$
- i) $W[0]$
- j) $X[2]$

NOTA: se não existir resposta, indicar com **ERRO**

6.

Considere um ficheiro de texto de nome "**dados.txt**", em que **cada linha** deste ficheiro contém dois números **inteiros**. Implemente um **programa em C** que realize as seguintes acções, pela ordem indicada:

- 1º) **construa** um **array 1D** com os **números** contidos no ficheiro "**dados.txt**", usando **gestão dinâmica de memória**, da seguinte forma: **para cada linha** do ficheiro, **leia os dois números e acrescente ao array os dois números** (se os dois números são diferentes entre si) ou **acrescente apenas um** deles (se os dois números são iguais),
- 2º) **guarde** os elementos do **array** que são **positivos** no ficheiro de texto "**saida.txt**".

7.

Implementar uma **função recursiva** em C que dados um **array 1D X** com **N** números **reais** (com X e N são parâmetros da função), **determine a soma** dos números **positivos** do **array X**.

8.

Implementar um **subprograma** em C que dados um **array 1D A** com **N** números **inteiros** (A e N > 0 são parâmetros do subprograma), **remova** do array **A** todos os **números que se repetem**, ficando o array **sem repetidos** (mantendo ou não a ordem inicial entre do números).

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

Exemplo: $A = [2 \ -4 \ 9 \ -4 \ 2 \ 5 \ 9 \ -4 \ 7] \Rightarrow A = [2 \ -4 \ 9 \ \blacksquare \ 2 \ 5 \ 9 \ \blacksquare \ 7]$