

UNIVERSIDADE DA BEIRA INTERIOR

Programação – LEI

1º Semestre

Exame Época Normal

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** **pertence** ao intervalo **[-50, 50[**
- b) o valor da variável do tipo inteiro **B** **é** um número **ímpar** e **pertence** ao conjunto **{10, ..., 99 }**

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

- c) a variável **B** recebe a **soma** dos **dois últimos algarismos** do valor da variável do tipo inteiro **A** (exemplo: $A = 23542 \Rightarrow B = 6$, pois $6 = 4 + 2$)
- d) a variável **P** recebe o valor **-5** se o valor da variável **N** for **par** e o valor **15** se **N** for **ímpar** (**P** e **N** são variáveis do tipo inteiro)

Supondo que **X = 5**, **Y = -2** e **Z = 2** (**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) >= (Z + X) % 3
 1 2 3 4 5

f) X + (2.5 - Y) * (25 / (X * Z))
 1 2 3 4 5

Sugestão de resposta (exemplo):

- ordem de cálculo: 5, 3, 2, 1, 4
- valor: 34 (e apresentar cálculos)

2.

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

- peça ao utilizador e **insira/leia** um número **inteiro N** com valor **entre 1 e 20** ($1 \leq N \leq 20$)
- peça ao utilizador e **insira/leia** um número **inteiro M** com valor **entre N e 50** ($N \leq M \leq 50$)
- **determine** a **soma** dos números **pares entre N e M**, inclusivé
- **mostre** o **resultado obtido** (soma dos números pares).

3.

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

- peça ao utilizador e **insira/leia** um número **inteiro N** com valor **entre 1 e 20** ($1 \leq N \leq 20$)
- peça ao utilizador e **insira/leia** um número **inteiro M** com valor **entre N e 50** ($N \leq M \leq 50$)
- **determine** a **soma** dos números **pares entre N e M**, inclusivé
- **mostre** o **resultado obtido** (soma dos números pares).

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** a **quantidade** (número inteiro) de números **positivos de X** e a **média aritmética** (número real) dos números **positivos de X**.

NOTA: caso o array **X** não contenha números positivos, considerar que o valor da média é 0 (zero).

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)

	...	
100500	110168	V
	...	
100700	110164	X
	...	
110160	90	
110164	800	
110168	700	
110172	600	
110176	50	
110180	40	
	...	
110500	100500	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) $X - 3$
- c) $\&V[2]$
- d) $*(X - 1)$
- e) $**W + 4$
- f) $W + 4$
- g) $*(*W + 3)$
- h) $*W - 2$
- i) $W[0]$
- j) $X[1]$

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: **um número real e um número inteiro**, por esta ordem. Implemente um **programa em C** que realize as seguintes acções, pela ordem indicada:

- 1º) **construa** um **array 1D X** com os **números reais** contidos no ficheiro de texto "**dados.txt**", usando **gestão dinâmica de memória**,
- 2º) **determine** o **maior** elemento e o **menor** elemento do array X, e
- 3º) **guarde** os valores do **maior** e do **menor** obtidos no passo 2, no ficheiro de texto "**saida.txt**".

7.

Implementar uma **função recursiva** em C que dados um **array 1D A** com **N** ($N \geq 1$) números **inteiros**, e um **número inteiro K** (com A, N e K parâmetros da função), **determine** quantos **elementos** do **array A** têm **valor** igual ao valor de **K**.

8.

Implementar um **subprograma** em C que dados um **array 1D A** com **N** números **inteiros** (A e $N > 0$ são parâmetros do subprograma), **determine e devolva o segundo maior elemento negativo do array A, caso exista** (se não existe, deve devolver um valor positivo).

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

Exemplo: $A = [2 \ -4 \ -9 \ 34 \ -12 \ 25 \ 76 \ -54] \Rightarrow$ segundo maior negativo = **-9** (maior = -4)