

Duração: 2:15 h

1.

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

(a) o valor da variável **X** deve ser **positivo** e **não deve pertencer** ao intervalo] -50, 100 [

(b) 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 ações:

(c) a variável **B** recebe o valor do número formado pelos algarismos das centenas e das dezenas (por esta ordem) do valor da variável **A** (A e B são do tipo inteiro e $A > 0$)

(Ex: $A = 23547 \Rightarrow B = 54$)

(d) a variável **P** recebe o valor **-5** se o valor de **N** for **par** e o valor **3** se o valor de **N** for **ímpar** (P e N são variáveis do tipo inteiro e $N > 0$)

Supondo que **X = 9**, **Y = 6** 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) $X > 9 \ \&\& \ (Y < 6 \ || \ Z \leq 2)$
 1 2 3 4 5

(f) $20 / (Y + Z) * (1.2 + 13 \% X)$
 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 **programa** em C que traduza o **fluxograma** dado na última página do enunciado (faça uma correspondência direta).

3.

Implementar uma **função** em C que dados **dois** números **inteiros** positivos não nulos (> 0) **M** e **N** (M e N parâmetros da função), **determine e devolva** a **soma** de todos os **divisores de 5** que existem entre o **menor dos dois valores** de entrada e **MxN**.

Exemplo: $M = 8$ e $N = 12$ ($M \times N = 8 \times 12 = 96$); soma = 945 ($10 + 15 + 20 + \dots + 95$).

4.

Implementar uma **função** em C que dado um array 1D **A** com **N** números inteiros e um número inteiro **K** ($N > 0$, A e $K > 0$ são parâmetros da função), **determine e devolva** a **quantidade** de números do array A que são **divisores de K** e a **quantidade** de números do array A que são **múltiplos de K**.

Notas:

1) **30** é **múltiplo** de **15** e **5** é **divisor** de **15**

2) na resolução deste exercício não pode usar outros arrays.

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

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) $X + 2$
- 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 não existir resposta, indicar com **ERRO**

6.

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

- 1) **construa** dois arrays 1D **X** e **Y** com os números positivos (**X**) e os números negativos (**Y**) do ficheiro "dados.txt", usando gestão de memória dinâmica,
- 2) **determine** a **soma** de números positivos e a **soma** de números negativos (< 0) usando os arrays **X** e **Y** construídos em 1),
- 3) **guarde** no ficheiro de texto "resultados.txt" os resultados obtidos no passo anterior (as somas).

NOTA: não é necessário verificar se há ERRO na abertura dos ficheiros e na alocação de memória

7.

Implemente uma **função recursiva** em C que dados um array 1D **A** com **N** números inteiros não nulos ($N > 0$ e **A** são parâmetros da função), **devolva** o número negativo do array **A** com menor índice. Caso no array **A** não existam números negativos, a função deve devolver **0**.

Exemplo: $A = [6 \text{ } -5 \text{ } 3 \text{ } 8 \text{ } -9 \text{ } 7]$ ($N = 6$) $\Rightarrow -5$ ($A[1] = -5$)

8.

Implemente uma **função** em C que dado um array 1D **A** com **N** números inteiros positivos ($N > 0$ e **A** são parâmetros da função), **determine e devolva** o **segundo menor** número do array **A** (se todos os números de **A** são iguais, então considera-se como segundo menor o valor **-1**).

NOTA: o array **A** só pode ser percorrido uma única vez e não pode usar outros arrays nem ficheiros.

Exemplo:

$A = [3 \text{ } 5 \text{ } 3 \text{ } 8 \text{ } 3 \text{ } 4 \text{ } 8 \text{ } 6 \text{ } 2 \text{ } 5] \Rightarrow$ segundo menor número: 3 (o menor é o 2)

