

TRABALHO 1 – VALOR: 15,0
data entrega: 10/11/2025

- 1) O trabalho será individual;
- 2) Criar um único “projeto” com o Menu e todas as resoluções das opções solicitadas no menu.
- 3) Você deverá enviar os arquivos (projeto *.cbp e main.c), COMPACTADOS em um **ÚNICO ARQUIVO**, identificando seu nome e trab1.
Deverá antes de compactar, apagar os arquivos/pastas “bin” e pasta “obj” de cada projeto.

Exemplo:

MENU DE OPÇÕES										
=====										
1 – ordenar vetor										
2 – quantidade números que aparecem mais de uma vez										
3 – uniao, interseção e diferença										
4 – multiplicação de vetores										
5 – vetor resultante										
6 – intercalação de vetores										
7 – controle de estoque										
8 – Finalizar programa (algoritmo)										
Escolha o número de uma das opções acima...:										

Opcão 1 – Receber 10 números em um vetor numérico com nome “V”. Logo após a entrada de dados, limpar a tela e mostrar esse vetor em sua ordem de entrada. Em seguida, processar a classificação desse vetor em ordem crescente, ou seja, do menor elemento para o maior elemento.

Exemplo, caso o usuário tenha digitado os números abaixo:

V	5	15	3	28	7	6	35	-8	26	19
	0	1	2	3	4	5	6	7	8	9

Vetor classificado:

V	-8	3	5	6	7	15	19	26	28	35
	0	1	2	3	4	5	6	7	8	9

Obs.: Separar os conteúdos em vídeo por vírgula e o ÚLTIMO com ponto final.

Obs.: Criar pelo menos 3 funções.

Opcão 2 – Receber 10 números em um vetor numérico, processar e mostrar os números que compõem esse vetor e quantidade de vezes que cada número aparece.

Exemplo:

V	5	15	5	28	5	15	25	28	26	15
	0	1	2	3	4	5	6	7	8	9

Ocorrência do número:

5	apareceu	3 vez(es)
15	apareceu	3 vez(es)
28	apareceu	2 vez(es)
25	apareceu	1 vez(es)
26	apareceu	1 vez(es)

Obs.:

- a) Os números que já foram apresentados **não irão** aparecer novamente;
- b) **Não usar** outro vetor para armazenar os já apresentados (ou repetidos);
- c) Separar os conteúdos em vídeo por vírgula e o último com ponto.
- d) Criar pelo menos 3 funções

Opcão 3 – Receber dois vetores “A” e “B” com 100 posições cada. Use a diretiva “#define” para definir o tamanho dos vetores.

- a) “UNIAO”, o qual será composto pela união de “A” com “B”, ou seja, todos elementos de “A” e todos elementos de “B”;
- b) “INTER”, o qual será composto pela interseção de “A” com “B”, ou seja, somente os elementos de “B” que aparecem no vetor “A”;
- c) “DIFER”, o qual será composto pela diferença de “A” com “B”, ou seja, todos elementos de “A” que não existem em “B” e todos elementos de “B” que não existem em “A”;

Exemplo:

A	5	15	25	35	40	45	50	55	60	65
	0	1	2	3	4	5	6	7	8	9

B	200	15	300	40	500	600	700	60	5	800
	0	1	2	3	4	5	6	7	8	9

UNIAO	<table border="1"> <tr><td>5</td><td>15</td><td>25</td><td>35</td><td>40</td><td>45</td><td>50</td><td>55</td><td>60</td><td>65</td><td>200</td><td>15</td><td>300</td><td>40</td><td>...</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td></tr> </table>	5	15	25	35	40	45	50	55	60	65	200	15	300	40	...	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
5	15	25	35	40	45	50	55	60	65	200	15	300	40	...																	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14																	
INTER	<table border="1"> <tr><td>15</td><td>40</td><td>60</td><td>5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td></td><td></td><td></td><td></td><td></td></tr> </table>	15	40	60	5												0	1	2	3	4	5	6	7	8	9					
15	40	60	5																												
0	1	2	3	4	5	6	7	8	9																						
DIFER	<table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td></td><td></td><td></td><td></td><td></td></tr> </table>																0	1	2	3	4	5	6	7	8	9					
0	1	2	3	4	5	6	7	8	9																						

Obs.:

- a) Criar uma ÚNICA função para mostrar os vetores. Esta função deverá receber o vetor e o tamanho do vetor. Separar os com vírgula e o último com ponto final.
 - b) Os vetores serão mostrados até as posições que foram utilizados;
 - c) Criar pelo menos 3 funções
-

Opcão 4 – Faça um programa que carregue dois vetores de 60 elementos numéricos cada um e mostre um vetor resultante da multiplicação desses dois vetores

Obs.: Criar as seguinte funções:

- a) Função para receber os dados;
 - b) Função para mostrar vetor
 - c) Função para criar o vetor resultante.
-

Opcão 5 – Faça um programa que carregue um vetor com 80 números inteiros, calcule e mostre dois vetores resultantes. O primeiro vetor resultante deve conter os números positivos. O segundo vetor resultante deve conter os números negativos. Cada vetor resultante vai ter *no máximo oitenta posições*, sendo que nem todas devem obrigatoriamente ser utilizadas

Obs.: Criar as seguinte funções:

- a) Função para receber os dados;
 - b) Função para mostrar vetor
-

Opcão 6 – Faça um programa que receba valores numéricos e inteiros para dois vetores numéricos de 5 elementos.

Pede-se:

- a) Exiba um vetor com o resultado da intercalação de ambos os vetores recebidos.

Ex:

Vetor A	1	2	5	7	9
Vetor B	2	4	6	7	10

Vetor intercalado	1	2	2	4	5	6	7	7	9	10
-------------------	---	---	---	---	---	---	---	---	---	----

- b) Exiba um vetor com o resultado da concatenação de ambos os vetores recebidos.

Ex:

Vetor A	1	2	5	7	9
Vetor B	2	4	6	7	10

Vetor concatenado	1	2	5	7	9	2	4	6	7	10
-------------------	---	---	---	---	---	---	---	---	---	----

Obs.: Criar as seguinte funções:

- a) Função para receber os dados;
 - b) Função para mostrar vetor
-

Opcão 7 – Faça um programa para controlar o estoque de mercadorias de uma empresa. Inicialmente o programa deverá ler dois vetores com cem posições cada, onde o primeiro corresponde ao código do produto e o segundo corresponde ao total desse produto em estoque. Logo após, o programa deverá ler um conjunto indeterminado de dados contendo o código de um cliente, o código do produto que este deseja comprar juntamente com a quantidade. Código do cliente igual a zero indica fim do programa. O programa deverá verificar:

- Se o código do produto solicitado existe. Se existir, tentar atender o pedido; caso contrário, exibir a mensagem *Código Inexistente*.
- Cada pedido feito por um cliente poderá ser atendido integralmente. Caso isso não seja possível, escrever a mensagem *"Não temos estoque suficiente desta mercadoria."* Se puder atendê-lo, escrever a mensagem *"Pedido atendido Obrigado e volte sempre"*.
- Efetuar a atualização do estoque somente se o pedido for atendido integralmente.
- No final do programa, escrever os códigos dos produtos com os seus respectivos estoques já atualizados.

Obs.: Criar pelo menos 03 funções: