

Lista de Exercícios - Vetores

ATENÇÃO!!!!

É PROIBIDA a declaração automática (estática) dos vetores com VARIÁVEL na DIMENSÃO.
Apenas utilize constantes ou macros (constantes nomeadas).

```

1
2 int vet[5]; OK!!!
3
4 #define TAM 5
5 int vet[TAM] OK!!!
6
7 int tam, vet[tam]; PROIBIDO!!!!

```

1. Implemente um programa em C que preenche os elementos de um vetor de forma a permitir apenas a inserção de valores distintos. Caso o usuário tente inserir um valor já existente, o programa deve alertar a duplicidade e solicitar um novo valor.
2. Implemente um programa em C que preenche com valores aleatórios um vetor de N posições, onde N é uma **constante** pré-definida. Dada uma posição do vetor (informada pelo usuário), o programa deve remover o elemento desta posição, “puxando” todos os elementos das posições posteriores uma posição para “trás”. Por exemplo, se o vetor contém os números 1, 3, 6, 2 e 10 e o usuário digita 2, o programa deve remover o elemento na posição número 2 do vetor - o 6 (lembre-se que a contagem inicia em 0) - ficando com os números 1, 3, 2 e 10. Após a remoção, o programa deve mostrar o “novo” vetor na tela. Atenção, a solução deve declarar apenas o vetor original (não utilize mais de um vetor para resolver o problema).
3. Implemente um programa em C que gere dois vetores contendo N números inteiros aleatórios cada um. Após a geração dos vetores, o programa deve imprimir a soma dos elementos dos vetores da seguinte forma: o primeiro elemento do primeiro vetor + o último elemento do segundo vetor, o segundo elemento do primeiro vetor + o penúltimo elemento do segundo vetor, e assim por diante, até o último elemento do primeiro vetor + o primeiro elemento do segundo vetor.

Vetores e Funções

4. Faça uma função que receba como parâmetro um valor inteiro representando um limite superior e preencha um vetor de N posições com valores aleatórios de 0 até esse limite. Atenção: use essa função para preencher vetores nos exercícios seguintes.

Protótipo: `void preencheVetor(int v[], int n, int lim);`

5. Modifique a função anterior para que preencha um vetor de N posições com valores aleatórios (agora esses valores estão limitados a um intervalo `[min,max]`, onde `min` e `max` são parâmetros recebidos por valor pela função).

Protótipo: `void preencheVetor(int v[], int n, int min, int max);`

6. Faça duas funções: uma que imprima o conteúdo de um vetor em ordem direta (do primeiro ao último elemento) e outra que imprima em ordem inversa (do último elemento até o primeiro).
7. Faça uma função que, dados como entrada um vetor e a sua dimensão, retorne a média dos elementos pares.
8. Modifique a função do item anterior de forma que ela também retorne, **por referência**, o maior e o menor elemento do vetor.
9. Faça uma função com protótipo:

`void intercalaVetores(int v1[], int v2[], int v3[], int n);`

que receba como parâmetros de entrada dois vetores de inteiros (e sua dimensão) e preencha um terceiro vetor de tal forma que este possua nas posições de índice par os valores do primeiro e nas posições de índice ímpar os valores do segundo vetor (em outras palavras: intercalar os dois vetores iniciais. Obs - Todos os elementos dos vetores originais devem estar presentes no vetor intercalado conforme a figura abaixo).

v1	1	2	5	4
v2	2	3	7	3

v3	1	2	2	3	5	7	4	3
----	---	---	---	---	---	---	---	---

transcrição da figura acima:

exemplo do primeiro vetor com 4 elementos: 1 2 5 4

exemplo do segundo vetor com 4 elementos: 2 3 7 3

resultado do vetor que intercala o primeiro e segundo: 1 2 2 3 5 7 4 3

Extras

Tente refazer os exercícios 1 a 3 utilizando funções.