

Aula15 - Exercícios: TADs e listas sequenciais

1. Escreva um programa que lê um valor n , cria dinamicamente um vetor de n elementos, e passa esse vetor para uma função que vai ler os seus elementos. Depois, no programa principal, o vetor preenchido deve ser impresso. Além disso, antes de finalizar o programa, deve-se liberar a área de memória alocada.

2. Escreva um programa que solicita ao usuário a quantidade de alunos de uma turma e aloca um vetor de notas (números reais). Depois de ler as notas de todos os alunos, seu programa deverá imprimir a média aritmética da turma. Obs: note que não deve ocorrer desperdício de memória, e após a utilização da memória, ela deve ser liberada.

3. Escreva um programa que calcula a soma de duas matrizes $M \times N$ de números reais (double). A implementação deste programa deve considerar as dimensões fornecida pelo usuário. Obs: represente a matriz através de variáveis do tipo `double**`, usando alocação dinâmica de memória. Obs: lembre-se que diversas chamadas à função de alocação de memória devem ser efetuadas para poder construir uma matriz.

4. **(Desafio)** Escreva um programa que monte uma lista de números inteiros positivos. Para cada novo número inteiro digitado pelo usuário no teclado, o programa deve alocar uma nova variável no *heap* e inserí-la no final da lista composta pelos números já digitados. Quando o usuário digitar um número menor ou igual a 0 (zero), o programa deve imprimir todos os números digitados, na ordem em que eles foram alimentados pelo usuário. Utilizar o tipo de dados struct Item definido abaixo:

```
Struct Item {  
    int valor;  
    struct Item *proximo;  
};
```

5. Modificar o exercício 4 para imprimir os números digitados na ordem inversa daquela em que eles foram informados pelo usuário.

6. Modificar o exercício 4 para montar a lista na ordem crescente dos números informados pelo usuário.

7. **(Desafio)** Os motoristas estão preocupados com a quilometragem obtida por seus automóveis. Um motorista fez um controle de vários tanques de gasolina gastos, anotando os quilômetros de rodagem obtidos pelo tanque. Desenvolva um programa em C++ que recebe como entrada os quilômetros rodados e os litros para cada tanque. O programa deve calcular e exibir os quilômetros por litro obtidos para cada tanque. Depois de processar todas as informações fornecidas, o programa deve calcular e imprimir os quilômetros por litro obtidos por todos os tanques combinados. Você deve utilizar uma estrutura para representar as informações sobre cada automóvel. Lembre-se de alocar/liberar a memória necessária.