



Exercícios sobre Alocação Dinâmica

1. Codifique um programa em C que:
 - (a) Aloque dinamicamente um vetor formado por N números inteiros, sendo N um valor informado pelo usuário;
 - (b) Peça ao usuário para digitar os N números no espaço alocado;
 - (c) Imprima na tela os N números digitados;
 - (d) Libere a memória alocada.
2. Faça um programa em C que calcule a soma de duas matrizes $M \times N$ de números reais (double).
 - (a) A implementação deste programa deve considerar as dimensões fornecida pelo usuário (Dica: represente a matriz através de variáveis do tipo double **, usando alocação dinâmica de memória).
 - (b) Não esqueça de criar uma função para liberar a área de memória alocada pela matriz.
3. Faça um programa em C que contenha uma ou mais estruturas para representar os dados de um funcionário de uma empresa. O tipo Funcionário deve possuir os seguintes campos:
 - Nome do funcionário
 - Data de nascimento
 - Endereço completo
 - Salário

O programa deve alocar dinamicamente um vetor de N posições, sendo cada posição ocupada por um funcionário da empresa. Lembrando que N deve ser um número fornecido pelo usuário do programa. Em seguida, preencha os dados do funcionário e imprima os nomes dos funcionários que possuem o maior salário.

4. Escreva um programa que lê um vetor de inteiros (número de elementos definido pelo usuário), remove valores consecutivos repetidos e mostra o vetor atualizado. O programa deve alocar memória dinamicamente para utilizar o vetor, e o vetor deve ter ao final da operação sempre o tamanho exato dos elementos presentes nele. Permita ao usuário informar mais conjuntos de números, e defina uma condição para o usuário parar a criação de novos vetores. O programa deve, ao final de cada operação, mostrar o seu tamanho final.

5. Escreva um programa que aloque dinamicamente uma matriz (de inteiros) de dimensões definidas pelo usuário, e preencha a matriz com valores informados pelo usuário. O programa deve então construir um outro vetor que contenha exatamente os números que aparecem na matriz. Por exemplo, caso a matriz seja $[[1,3,5],[2,3,1],[1,1,6]]$, o vetor resultante será $[1,3,5,2,6]$. O vetor não precisa estar ordenado ao final da execução, e seu tamanho deve coincidir com o necessário para o resultado.
6. Faça um programa que pergunte ao usuário quantos valores ele deseja armazenar em um vetor de double, depois aloque dinamicamente o espaço de memória de acordo com o especificado pelo usuário. Esse vetor deve ter um tamanho maior ou igual a 10 elementos. Use este vetor dinâmico como um vetor comum, atribuindo aos 10 primeiros elementos do vetor valores aleatórios (usando a função rand) entre 0 e 100. Exiba na tela os valores armazenados nos 10 primeiros elementos do vetor.
7. Faça um programa que leia números do teclado e os armazene em um vetor alocado dinamicamente. O usuário digitará uma sequência de números, sem limite de quantidade. Os números serão digitados um a um e, se o usuário digitar o número 0 (ZERO), o programa termina. Antes de terminar, imprima todos os valores armazenados no vetor. Os dados devem ser armazenados na memória deste modo:
 - Inicie com um vetor de tamanho 5 alocado dinamicamente;
 - Caso o vetor alocado esteja cheio, aloque um novo vetor do tamanho do vetor anterior adicionado espaço para mais 5 valores (tamanho $N+5$, onde N inicia com 5);

Observações:

- Todas as questões devem ser feitas de três maneiras:
 - usando `malloc` e
 - utilizando `calloc` e
 - Ao terminar o programa solicitado na questão, solicite ao usuário “uma nova” quantidade de itens e, utilizando o `realloc`, acerte a memória conforme a solicitação.
- Antes de terminar o programa, desaloque a memória (`free`).