



Lista de Exercícios sobre Alocação Dinâmica

- 1) Faça um programa que leia um valor n e crie dinamicamente um vetor de n elementos e passe esse vetor para uma função que vai ler os elementos desse vetor. 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) Faça uma função que receba um valor n e crie dinamicamente um vetor de n elementos e retorne um ponteiro. Crie uma função que receba um ponteiro para um vetor e um valor n e imprima os n elementos desse vetor. Construa também uma função que receba um ponteiro para um vetor e libere esta área de memória. Ao final, crie uma função principal que leia um valor n e chame a função criada acima. Depois, a função principal deve ler os n elementos desse vetor. Então, a função principal deve chamar a função de impressão dos n elementos do vetor criado e, finalmente, liberar a memória alocada através da função criada para liberação.
- 3) Construa um programa (`main`) que aloque em tempo de execução (dinamicamente) uma matriz de ordem $m \times n$ (linha por coluna), usando $1+m$ chamadas a função `malloc`. Agora, aproveite este programa para construir uma função que recebendo os parâmetros m e n aloque uma matriz de ordem $m \times n$ e retorne um ponteiro para esta matriz alocada. Crie ainda uma função para liberar a área de memória alocada pela matriz. Finalmente, crie um novo programa (`main`) que teste/use as duas funções criadas acima.
- 4) Construa um programa (`main`) que aloque em tempo de execução (dinamicamente) uma matriz de ordem $m \times n$ (linha por coluna), usando 2 chamadas a função `malloc`. Agora, aproveite este programa para construir uma função que recebendo os parâmetros m e n aloque uma matriz de ordem $m \times n$ e retorne um ponteiro para esta matriz alocada. Crie ainda uma função para liberar a área de memória alocada pela matriz. Finalmente, crie um novo programa (`main`) que teste/use as duas funções criadas acima.
- 5) Criar um tipo abstrato de dados que represente uma pessoa, contendo nome, data de nascimento e CPF. Crie uma variável que é um ponteiro para este TAD (no programa principal). Depois crie uma função que receba este ponteiro e preencha os dados da estrutura e também uma função que receba este ponteiro e imprima os dados da estrutura. Finalmente, faça a chamada a esta função na função principal.
- 6) Idem a questão acima, mas construa as funções usando referências (&).