

## Alocação dinâmica de memória

1) Faça um programa que: i) solicite ao usuário um valor  $n$ ; ii) crie, dinamicamente, um vetor de  $n$  elementos; e iii) passe esse vetor para uma função que preencherá o vetor. Depois, no programa principal, o vetor preenchido deverá ser impresso. Antes de finalizar o programa, libere a área de memória alocada.

2) No programa anterior, exiba os  $n$  elementos do vetor dentro da função que preenche o vetor. Após exibir o vetor, pergunte ao usuário quantos elementos a mais ele deseja adicionar ao vetor. Em seguida, use a função `realloc` para aumentar o tamanho do vetor. Preencha as novas posições do vetor com números aleatórios. Exiba o vetor novamente. No programa principal, exiba todos os elementos do vetor.

3) Implemente as seguintes funções:

- Função **criar\_vetor** que solicita ao usuário um valor  $n$  (tamanho do vetor) e cria dinamicamente um vetor de  $n$  elementos.
- Função **exibir\_vetor** que recebe um ponteiro para um vetor e imprime os  $n$  elementos desse vetor.
- Função **liberar\_vetor** que recebe um ponteiro para um vetor e libera esta área de memória.

As funções deverão ser chamadas pelo programa principal na seguinte ordem: `criar_vetor`, `exibir_vetor` e `liberar_vetor`.

4) Implemente um programa que preenche aleatoriamente dois vetores  $Q$  e  $R$  com 12 e 20 elementos inteiros (valores repetidos ou não) respectivamente. Implemente um procedimento que recebe os vetores  $Q$ ,  $R$  e um terceiro vetor  $W$ . O procedimento deverá preencher o vetor  $W$  com a interseção de  $Q$  e  $R$ . Além disso,  $W$  não poderá conter elementos repetidos. Exiba o conteúdo do vetor  $W$  no programa principal. Utilize a função `malloc` para os vetores  $Q$  e  $R$ . Utilize a função `realloc` toda vez que um elemento for inserido no vetor  $W$ . Não utilize variáveis globais.

5) Criar um tipo abstrato de dados (struct) que represente uma pessoa, contendo nome, data de nascimento e CPF. Crie uma variável que é um ponteiro para esta estrutura no programa principal. Crie a função **preencher** que receba este ponteiro e preencha os dados da estrutura. Crie a função **exibir** que receba o ponteiro para a estrutura e imprima todos os elementos. Chame as duas funções a partir do programa principal.