

## Alocação dinâmica de memória

Vimos:

- 1) Funções de alocação;
- 2) Funções de redimensionamento;
- 3) Funções para liberar memória;

//testar se realocar sem memória suficiente pra ver o que dá e o que a função retorna;

A função malloc/calloc retorna NULL quando a alocação não é bem sucedida;

==> Tratar o caso em que a alocação é mal-sucedida;

```
int *v = (int *) malloc (10*sizeof(int));

if (v==NULL){
    printf("A alocação de memória não foi possível.\n");
    exit(1);
}
else{
    return 0;
}
```

///pesquisar tabela de tipos de erros à serem retornados

//realizar operações com v

==>Podemos implementar uma função para alocar e verificar o sucesso da alocação.

```
void * malloc (size_t nbytes){
    void *ptr = malloc(nbytes);
    if(ptr==NULL){
        printf("A alocação de memória não foi possível \n.");
        exit(1);
    }
    return ptr;
}
```

## Exercícios

1 – Faça um programa que leia  $n$  inteiros (definidos pelo usuário) armazenando-os em uma memória alocada dinamicamente. Em seguida, mostre quantos dos números são pares e quantos são ímpares.

2 – Faça um programa que receba do usuário o tamanho de uma string e chame uma função para alocar dinamicamente essa string. Em seguida, o programa deverá informar o conteúdo dessa string sem suas vogais.

3 – Faça um programa que :

a) Cria uma matriz de distâncias entre  $n$  cidades diferentes;

b) peça para o usuário entrar com as distâncias entre as cidades; exiba na tela a matriz de distâncias criadas;

c) exiba na tela a matriz de distâncias criada;

d) quando o usuário digitar o número de duas cidades, o programa deverá imprimir a distância entre elas.

4 – Escreva um programa que aloque dinamicamente uma matriz (de inteiros) de dimensões definidas pelo usuário. Em seguida, implemente uma função que receba um valor esteja na matriz e 0 caso não esteja na matriz. Implemente a função main de forma a testar todas as funções implementadas.

5 – Considere um cadastro de produtos de um estoque com as seguintes informações para cada produto:

- Código de identificação do produto: inteiro;
- Nome do produto: com até 50 caracteres;
- Quantidade disponível no estoque: inteiro;
- Preço de venda: valor real;

A) Defina uma estrutura em C, denominada Produto, que contenha os campos apropriados para guardar as informações de um produto;

B) Crie um conjunto de  $n$  produtos ( $n$  é um valor fornecido pelo usuário) e peça ao usuário para entrar com as informações de cada produto;

C) Encontre o produto com o maior preço de venda;

D) Encontre o produto com a maior quantidade disponível no estoque;

6 – Conferindo resultados da loteria faça. Faça um programa que lê o número de elementos marcados em um jogo ( $n$ ), o número de elementos sorteados ( $s$ ), os  $n$  números de uma aposta e os  $s$  números sorteados. Em seguida, seu programa deve imprimir o total de acertos no jogo feito;