

- 1) **Alocação dinâmica:** outra utilidade dos ponteiros é que eles permitem fazermos o que é chamado de alocação dinâmica. Isso significa que podemos reservar espaços de memória enquanto estamos executando o programa. Seria como criar variáveis com o programa executando. Por exemplo, considere um vetor que armazenará o preço de produtos. Não sabemos quantos produtos serão cadastradas enquanto estamos programando. Podemos fazer alocação dinâmica para resolver este problema. Implemente o programa abaixo, teste para diferentes valores de **n** e discuta o que significa o valor de **n** e de **sizeof(double)** no commando `malloc(n*sizeof(double))`.

```
double *produtos;
int n,i;

printf("Informe o número de produtos");
scanf("%d",&n);

// é necessário usar o comando malloc para alocar a memória
produtos = (double *)malloc(n*sizeof(double));

for (i = 0; i < n; i++){
    printf("Informe o valor do produto %d R$:",i+1);
    scanf("%lf", &produtos[i]);
}

printf("\nProdutos cadastrados\n");
for (i = 0; i < n; i++){
    printf("Produto %d - R$: %f\n",i+1, produtos[i]);
}

// ao terminar de usar o vetor, devemos liberar a memória
free(produtos);
```

- 2) Crie um programa que:
- (a) Aloque dinamicamente um array de 5 números inteiros,
  - (b) Peça para o usuário digitar os 5 números no espaço alocado,
  - (c) Mostre na tela os 5 números,
  - (d) Libere a memoria alocada.
- 3) Faça um programa que leia **n** inteiros (definidos pelo usuário) armazenando-os em uma memória alocada dinamicamente. Em seguida, mostre quantos dos **n** números são pares e quantos são ímpares.
- 4) Crie uma struct que armazene pontos inteiros (coordenadas **x** e **y** – valores inteiros). Crie dinamicamente um vetor de tamanho **n** (informado pelo usuário) e indique as coordenadas **x** e **y** de cada ponto. Ao final, mostrar todos os números digitados

Exemplo de saída:

Quantos pontos deseja digitar: 5

Entre com a coordenada x do ponto 1: 3

Entre com a coordenada y do ponto 1: 5

Entre com a coordenada x do ponto 2: 4

Entre com a coordenada y do ponto 2: 8

....

Entre com a coordenada x do ponto 5: 23

Entre com a coordenada y do ponto 5: 25

Pontos digitados: (3,5); (4,8); ... (23,25)

- 5) Modifique o programa anterior incluindo quatro ponteiros, que apontarão para os seguintes pontos: mais a esquerda, mais a direita, mais acima, mais abaixo. Mostre quem são esses pontos utilizando esses ponteiros.
- 6) Fazer os exercícios que envolvem alocação dinâmica e funções (presentes na lista de funções)