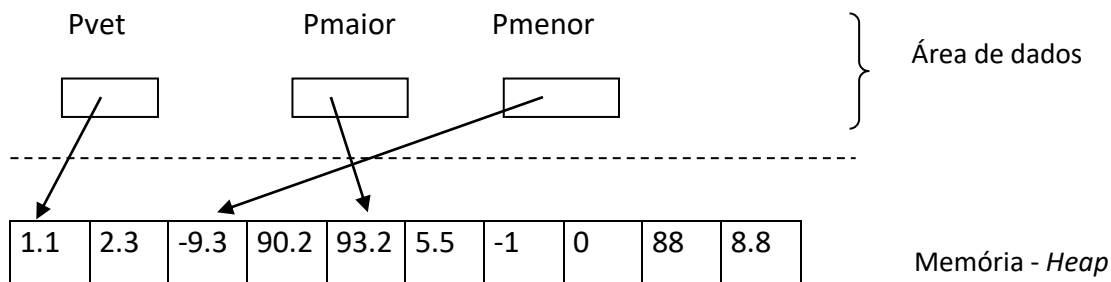


**EXERCÍCIOS****Assuntos:** vetor, struct, funções e alocação dinâmica

1) Escreva um programa que crie um vetor dinâmico, de reais, de **N** posições (o tamanho do vetor, **N**, e os dados do vetor devem ser fornecidos pelo usuário) e encontre o menor e o maior elemento. Porém, ao invés de guardar o índice ou o valor em uma variável estática, crie dois ponteiros que apontem para as respectivas posições do vetor dinâmico. Veja o exemplo abaixo, com um vetor de 10 posições:



```
float *Pvet, *Pmaior, *Pmenor;
```

Usando os ponteiros **Pmaior** e **Pmenor**, imprima o maior e o menor número do vetor:

Maior Número: 93.2

Menor Número: -9.3

2) Escreva um programa que crie um **vetor** dinâmico para reais, representando as notas de alunos, de **n** posições. Verificar a nota que mais se repete. Caso tenha mais de uma delas que se repete, escolher a maior entre elas. Por exemplo, num conjunto de **N=50** notas, supor que as notas: 8,5 e 5,5 são as notas que mais se repetem, por exemplo, ambas repetem 15 vezes. Então escolher, a nota 8,5. Imprimir os elementos do vetor e a nota que mais se repete. O programa deve ter a seguinte divisão de módulos:

- **função 1:** ler **N**, alocar dinamicamente o vetor de **N** reais, ler as informações do vetor. Comunicar o vetor pelo **return** e o **N** pela lista de parâmetros;
- **função 2:** receber o valor de **N**, os valores do vetor e encontra a nota que mais se repete. A comunicação dessa nota deve ser pelo **return**;
- **função 3:** receber o valor de **N**, os valores do vetor e faz a impressão do vetor;
- **main:** declarar as variáveis do problema, chamar as funções e imprimir a nota que mais se repete encontrada, após a impressão dos elementos do vetor.

3) Escreva um programa que crie um **vetor** dinâmico para reais, representando as notas de alunos, de **n** posições. Supor que as notas aceitas são inteiras ou com parte fracionária 0,5. Isto é, não tem parte fracionária com outros valores. Acrescenta 0,5 ponto para todas as notas com valor quebrado. Imprimir os

elementos do vetor duas vezes, como lido e depois de alterados os valores. O programa deve ter a seguinte divisão de módulos:

- a. **função 1:** ler **N**, alocar dinamicamente o vetor, ler as informações do vetor. Comunicar o vetor **e** o **N** pela lista de parâmetros;
- b. **função 2:** receber o valor de **N**, os valores do vetor, acrescentar 0,5 ponto somente nas notas quebradas.
- c. **função 3:** receber o valor de **N**, os valores do vetor e faz a impressão do vetor;
- d. **main:** declarar as variáveis do problema; chama a função1 para as leituras; a função3 para a impressão dos elementos lidos; a função2 para a alteração das notas quebradas e novamente a função3, agora, com as notas alteradas. Não definir dois vetores, apenas um, A alteração das notas deverá ser feita no próprio vetor lido.

- 4) Escreva um programa que crie uma **struct** aluno (ra – inteiro; nome – string; nota – real) aloca dinamicamente **n** posições, para receber as informações de **n** alunos. Calcular o valor médio das notas. Imprimir todos os dados dos alunos e o valor encontrado. O programa deve ter a seguinte divisão de módulos:

- a. Função1: ler **N**, alocar dinamicamente o espaço para os alunos, ler as informações. Comunicar as informações pelo return e o **N** pela lista de parâmetros;
- b. Função2: receber o valor de **N**, os valores dos alunos e calcular o valor médio das notas. A comunicação do valor médio deve ser pelo return;
- c. Função3: receber o valor de **N**, os valores dos alunos e fazer a impressão de todos os alunos no formato de tabela;
- d. main: declarar as variáveis do problema, chamar as funções e imprimir o valor médio.

- 5) Escreva um programa que cria uma **struct** produto (código: inteiro; nome: string; preco: real; qtdd: inteira;) e cria um vetor dinâmico para **N** produtos (o tamanho do vetor, **N**, e os dados do vetor devem ser lidos). Buscar um determinado produto tendo como entrada o código e imprimir todos os dados desse produto. O código do produto a ser buscado deve ser lido. O programa deve ter a seguinte divisão de módulos:

- a. Função1: ler **N**, alocar dinamicamente o vetor, ler as informações do vetor. Comunicar as informações pela lista de parâmetros;
- b. Função2: receber o valor de **N**, os valores do vetor e buscar o produto cujo **código deve ter sido lido no main** e passado na lista de parâmetros. A posição, dentro do vetor, do produto buscado deve retornar pelo return. Se não encontrou o produto retornar -1 pelo return. Não imprimir mensagem dentro da função;
- c. Função3: imprime os dados de **UM** produto.
- d. main: declarar as variáveis do problema e chama as funções para:
  - i. alocar espaço e ler dados dos produtos;
  - ii. ler o código a ser buscado;

- iii. buscar o produto desejado, com o código lido;(chamar a função busca\_produto);
- iv. com a posição encontrada na função anterior, imprimir os dados do produto. Antes deve testar se o retorno foi -1. Isso significa que não encontrou o produto. Não imprima, na função, a mensagem. As mensagens devem ficar no main;

6) Continuando o programa anterior sobre **struct** produto, acrescente **as seguintes funções**:

- a. **Função:** busca o produto de maior preço e o de menor preço. As posições desses produtos devem voltar pela lista de parâmetros. No main, incluir a chamada dessa função e a impressão desses produtos, chamando a função imprimir\_um\_produto, construída anteriormente, duas vezes;
- b. **Função:** buscar um determinado produto (como a função anterior: busca\_produto) com a diferença é que a busca será pelo nome do produto. Retornar pelo return a posição de localização. Da mesma forma o nome do produto deve ser lido no main e passado como parâmetro. No main, fazer a chamada dessa função e posteriormente imprimir os dados do produto buscado, utilizando-se a função imprimir\_um\_produto. Se retornou -1, imprimir no main mensagem.
- c. **Função:** Aumentar em 10% o preço de todos os produtos. No main chamar a função e de forma repetida, **imprimir os dados de todos os produtos.**