

INTRODUÇÃO À PROGRAMAÇÃO

LISTA DE EXERCÍCIOS 10

1. Faça um programa para ler e armazenar um conjunto de números em um vetor (máximo 50 números e a leitura de um número igual a zero indica fim da leitura dos dados). A seguir peça para o usuário digitar um valor inteiro e informe se o mesmo pertence ou não ao conjunto de números armazenados.

O programa deve implementar e usar a função `busca`, que recebe como parâmetro um vetor de números inteiros (`vet`) de tamanho `n` e um valor `x`. A função deve retornar 1 se `x` pertence a esse vetor e 0 caso contrário. A função deve obedecer ao seguinte protótipo:

```
int busca(int vet[], int n, int x);
```

2. Faça um programa para ler e armazenar um conjunto de números em um vetor (máximo 50 números e a leitura de um número igual a zero indica fim da leitura dos dados). A seguir peça para o usuário digitar um valor inteiro e informe quantos números maiores e quantos números menores que o valor digitado existem neste conjunto de números armazenados no vetor.

O programa deve implementar e usar as funções `maiores` e `menores` que recebe como parâmetro um vetor de números inteiros (`vet`) de tamanho `n` e um valor `x`. As funções devem retornar quantos números maiores e menores do que `x` existem nesse vetor. As funções devem obedecer aos seguintes protótipos:

```
int maiores(int vet[], int n, int x);
```

```
int menores(int vet[], int n, int x);
```

3. Escreva um programa para ler 20 números inteiros em ordem crescente (os números devem ser digitados pelo usuário em ordem crescente). O programa deve armazenar os valores em um vetor, e em seguida, pedir para usuário informar um número inteiro a ser buscado no vetor. Caso o valor seja encontrado, o programa deve exibir o índice onde o valor está. Caso contrário, deve exibir a mensagem “número não encontrado”.

O programa deve implementar e usar a função `busca_ordenado`, que recebe como parâmetro um vetor de inteiros (`vet`) de tamanho `n` e um valor `x` a ser buscado. Caso o valor buscado pertença ao vetor, a função deve retornar o índice do valor no vetor. Caso contrário, deve retornar -1. A função deve obedecer ao seguinte protótipo:

```
int busca_ordenado(int vet[], int n, int x);
```

Importante: A sua função deve ser eficiente e levar em consideração a ordenação dos elementos do vetor. Ela deve garantir no mínimo que o valor buscado seja encontrado testando no máximo metade dos valores do vetor.

4. Escreva um programa que leia duas sequências de dez números inteiros e imprima na tela quantos desses números estão presentes nas duas sequências e quantas ocorrências há de números idênticos na mesma posição das duas sequências.

Por exemplo, para as seguintes sequências:

```
1 20 102 42 37 20 11 53 11 1
11 20 1000 1110 532 42 37 87 87 1
```

O programa deverá imprimir na tela:

```
Ha 5 numeros presentes nas duas sequencias.
Ha 2 ocorrencias de numeros na mesma posição das duas sequencias.
```

Note que os números 1, 20, 42, 37 e 11 aparecem nas duas sequências (5 números comuns nas duas sequências). Na segunda posição de ambas as sequências temos o número 20 e na última posição o número 1 (isto é, 2 ocorrências de números na mesma posição).