

INF1005 - PROGRAMAÇÃO 1

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 o arquivo "numeros.txt", que contém 20 números inteiros 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 de um arquivo “numeros.txt” e imprima na tela quantos desses números estão presentes nas duas sequencias e quantas ocorrências há de números idênticos na mesma posição das duas sequencia.

Por exemplo, para um arquivo contendo o seguinte conteúdo:

```
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 sequencias (5 números comuns nas duas sequencias). Na segunda posição de ambas as sequencias temos o número 20 e na última posição o número 1 (isto é, 2 ocorrências de números na mesma posição).

5. Os organizadores da Loto Digital da PUC-Rio gostariam de automatizar o processo de verificação dos ganhadores. Durante a rodada loto, os números sorteados são armazenados no arquivo “resultado.txt” (total de 15 números). Os números apostados pelos jogadores são armazenados no arquivo “apostas.txt”, onde cada linha representa uma aposta de um jogador.

Faça um programa para ler o resultado da loto e em seguida, para cada aposta, verificar se aquela aposta é uma aposta ganhadora.

O seu programa deve implementar e usar a função genérica `estacontido` que receba dois vetores X e Y e dois números inteiros m e n ($m \leq n$) representando, respectivamente, a quantidade de elementos de cada um dos vetores. Esta função deve retornar 1, caso X esteja contido em Y ou 0 caso contrário. Lembre-se de que ao encontrar o 1º elemento de X que não pertença a Y a função já pode retornar 0.

6. Escreva um programa em C para gerenciar um sistema de reservas de mesas em uma casa de espetáculo. A casa possui 30 mesas de 5 lugares cada. O programa deverá permitir que o usuário escolha o código de uma mesa (inteiro entre 100 e 129) e forneça a quantidade de lugares desejados. O programa deverá informar se foi possível realizar a reserva e atualizar a reserva. Se não for possível, o programa deverá emitir uma mensagem e mostrar o número das mesas que ainda possuem a quantidade de lugares desejados. O programa deve terminar quando o usuário digitar o código 0 (zero) para uma mesa ou quando todos os 150 lugares estiverem ocupados. OBRIGATORIAMENTE, modularize sua solução!

Exemplo da execução do programa:

```
Entre com o código da mesa que deseja reservar (entre 100 e 129) ou
0 para terminar: 100
Entre com a quantidade de lugares que deseja: 3
Reserva Ok. Mesa 100, numero de cadeiras reservadas: 3.
Entre com o codigo da mesa que deseja reservar (entre 100 e 129) ou
0 para terminar: 101
Entre com a quantidade de lugares que deseja: 4
Reserva Ok. Mesa 101, numero de cadeiras reservadas: 4.
Entre com o codigo da mesa que deseja reservar (entre 100 e 129) ou
0 para terminar: 100
Entre com a quantidade de lugares que deseja: 3
Nao foi possivel realizar sua reserva.
As mesas que possuem 3 lugares disponiveis sao:
Mesa 102 - Num de lugares livres: 5
Mesa 103 - Num de lugares livres: 5
Mesa 104 - Num de lugares livres: 5
Mesa 105 - Num de lugares livres: 5
Mesa 106 - Num de lugares livres: 5
Mesa 107 - Num de lugares livres: 5
Mesa 108 - Num de lugares livres: 5
Mesa 109 - Num de lugares livres: 5
Mesa 110 - Num de lugares livres: 5
Mesa 111 - Num de lugares livres: 5
Mesa 112 - Num de lugares livres: 5
Mesa 113 - Num de lugares livres: 5
Mesa 114 - Num de lugares livres: 5
Mesa 115 - Num de lugares livres: 5
Mesa 116 - Num de lugares livres: 5
Mesa 117 - Num de lugares livres: 5
Mesa 118 - Num de lugares livres: 5
Mesa 119 - Num de lugares livres: 5
Mesa 120 - Num de lugares livres: 5
Mesa 121 - Num de lugares livres: 5
Mesa 122 - Num de lugares livres: 5
Mesa 123 - Num de lugares livres: 5
Mesa 124 - Num de lugares livres: 5
Mesa 125 - Num de lugares livres: 5
Mesa 126 - Num de lugares livres: 5
Mesa 127 - Num de lugares livres: 5
Mesa 128 - Num de lugares livres: 5
Mesa 129 - Num de lugares livres: 5
Entre com o codigo da mesa que deseja reservar (entre 100 e 129) ou
0 para terminar: 0
Pressione qualquer tecla para continuar...
```