

LABORATÓRIO 25

FUNÇÕES COM VETORES

EXERCÍCIOS DE REVISÃO

VOCÊ DEVE ACOMPANHAR PARA OBTER INFORMAÇÕES COMPLEMENTARES

1. Por que não se usa `const` quando o parâmetro da função é um dos tipos básicos (`int`, `float`, `char`, `bool`, etc.) ?

```
void exibir (const int);  
void exibir (int);
```

2. Escreva uma função que receba um vetor de `double's` e retorne o valor do maior elemento no vetor. A função não deve alterar o conteúdo do vetor. Use a função para encontrar o maior valor presente em um arquivo texto. O arquivo contém uma quantidade qualquer de valores ponto-flutuantes separados por espaços, tabulações ou saltos de linha.

Nome do arquivo: **valores.txt**
O maior valor é 9.5

Dica: como não se conhece o conteúdo do arquivo, não é possível prever quantos valores ele contém. Será necessário ler o arquivo duas vezes, uma para contar e outra para ler os elementos.

3. Escreva uma função que receba um vetor e um valor inteiro. O vetor deve ser recebido através de ponteiros que indicam início e fim de faixa. A função deve colocar o valor inteiro em todas as posições da faixa. O valor inteiro deve ser lido de um arquivo binário que contém apenas um número codificado como um inteiro de 32 bits.

Nome do arquivo: **valor.bin**
Vetor preenchido com valor 15.

Dica: utilize o editor hexadecimal do Visual Studio para criar um arquivo binário com um valor inteiro de 32 bits. Uma forma fácil de fazer isso é criando um arquivo com a extensão `.bin`. Um dígito hexadecimal representa 4 bits. Dois dígitos representam 1 byte. Processadores Intel e AMD x86 são *little-endian*, ou seja, valores que ocupam vários bytes são organizados dos bytes menos significativos para os mais significativos, sendo assim o valor inteiro 15 é "0F 00 00 00".

EXERCÍCIOS DE FIXAÇÃO

VOCÊ DEVE FAZER OS EXERCÍCIOS PARA FIXAR O CONTEÚDO

1. Peça ao usuário para entrar com pares de números até que um dos valores do par seja zero (ou até um máximo de 10 pares). O programa deve armazenar os pares em um vetor. Escreva uma função que receba o vetor de pares e exiba os pares colocando sempre o menor número como primeiro elemento do par.

```
Digite pares de valores (0 para encerrar):
```

```
9 3
```

```
2 8
```

```
7 5
```

```
0
```

```
Pares organizados:
```

```
(3,9)
```

```
(2,8)
```

```
(5,7)
```

2. Escreva um programa que utilize as seguintes funções:

Preencher: recebe como argumento um vetor de double's e o tamanho do vetor. A função pede ao usuário que entre com valores para preencher o vetor. Ela encerra a entrada de valores quando o vetor encher ou quando o usuário digitar um valor não-numérico, e retorna o número de valores armazenados.

Mostrar: recebe um vetor de double's e seu tamanho e exibe o conteúdo do vetor.

Inverter: recebe como argumento um vetor de double's e seu tamanho e inverte a ordem dos valores armazenados.

O programa deve usar estas funções para preencher o vetor, mostrar o vetor, inverter o vetor, mostrar novamente o vetor, inverter todos exceto o primeiro e último elementos, e então mostrar o vetor mais uma vez. Utilize const sempre que apropriado.

```
Entre com até 10 valores:
```

```
21 32 85 43 17 80 fim
```

```
21 32 85 43 17 80
```

```
80 17 43 85 32 21
```

```
80 32 85 43 17 21
```

3. Refaça a questão anterior modificando as funções para usar dois parâmetros ponteiros que indiquem uma faixa de valores dentro do vetor. A função Preencher deve retornar um ponteiro para a posição seguinte à última posição preenchida, no lugar de retornar o número de valores armazenados. As outras funções podem usar esse ponteiro como segundo argumento para identificar o fim dos dados.

EXERCÍCIOS DE APRENDIZAGEM

VOCÊ DEVE ESCREVER PROGRAMAS PARA REALMENTE APRENDER

1. Construa uma função que receba dois ponteiros indicando uma faixa de elementos dentro de um vetor e um valor inteiro e retorne quantas vezes esse valor acontece dentro da faixa. Para testar a função construa um programa que inicializa um vetor para os elementos { 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0 } e mostra a quantidade de zeros e uns que tem dentro do vetor.

```
Existem 5 zeros e 6 uns na faixa indicada.
```

Sugestão: teste a função com outras faixas.

2. Construa uma programa que leia uma lista de até 100 números de um arquivo texto e armazene-os em um vetor. O programa deve passar o vetor para uma função que deve encontrar e retornar o menor elemento, o maior elemento, e suas respectivas posições dentro do vetor. Defina um registro para ser o tipo de retorno da função. Utilize const nos parâmetros da função sempre que possível.

```
Arquivo: números.txt
A posição 30 contém o menor número (3)
A posição 12 contém o maior número (98)
```

3. Escreva um programa que leia 10 valores do usuário e os insira em um vetor de forma que os elementos fiquem ordenados. Para isso, antes de cada inserção, use uma função Locate para localizar e retornar a posição correta do elemento dentro do vetor. Em seguida use outra função, OpenSpace, que receba uma posição e abra espaço no vetor nessa posição, “empurrando” os demais elementos para a próxima posição. Caso a operação de abrir espaço não seja possível, porque o vetor está cheio, a função deve retornar falso. Use estas funções no programa principal para inserir os elementos e depois exibir os elementos ordenados.

```
Digite 10 valores:
3 12 6 7 4 8 2 9 14 1

Os valores ordenados:
1 2 3 4 6 7 8 9 12 14
```

4. Construa uma função que receba dois vetores de inteiros, e crie um vetor dinâmico com o conteúdo dos dois vetores. A função deve retornar o endereço do vetor resultante, que deve ser exibido na tela pelo programa principal.

```
Vetor A: 1 4 5 7
Vetor B: 3 2 8 9
União: 1 4 5 7 3 2 8 9
```

Dica: não esqueça de liberar a memória utilizada pelo vetor dinâmico.