**4ª Lista de Exercícios**

1. Desenvolva um programa em linguagem C/C++ que receba dez valores inteiros. Em seguida ordene os valores utilizando o algoritmo de ordenação *BubbleSort*. Cada vez que o algoritmo detectar que dois valores devem mudar de posição o programa deverá exibir todos os valores do vetor. Ao final, o programa deverá exibir o vetor ordenado.

Exemplo:

**Entrada**

10 9 8 7 6 5 4 3 2 1

**Saída**

9 10 8 7 6 5 4 3 2 1

9 8 10 7 6 5 4 3 2 1

9 8 7 10 6 5 4 3 2 1

9 8 7 6 10 5 4 3 2 1

9 8 7 6 5 10 4 3 2 1

9 8 7 6 5 4 10 3 2 1

9 8 7 6 5 4 3 10 2 1

9 8 7 6 5 4 3 2 10 1

9 8 7 6 5 4 3 2 1 10

8 9 7 6 5 4 3 2 1 10

8 7 9 6 5 4 3 2 1 10

8 7 6 9 5 4 3 2 1 10

8 7 6 5 9 4 3 2 1 10

8 7 6 5 4 9 3 2 1 10

8 7 6 5 4 3 9 2 1 10

8 7 6 5 4 3 2 9 1 10

8 7 6 5 4 3 2 1 9 10

7 8 6 5 4 3 2 1 9 10

7 6 8 5 4 3 2 1 9 10

7 6 5 8 4 3 2 1 9 10

7 6 5 4 8 3 2 1 9 10

7 6 5 4 3 8 2 1 9 10

7 6 5 4 3 2 8 1 9 10

7 6 5 4 3 2 1 8 9 10

6 7 5 4 3 2 1 8 9 10

6 5 7 4 3 2 1 8 9 10

6 5 4 7 3 2 1 8 9 10

6 5 4 3 7 2 1 8 9 10

6 5 4 3 2 7 1 8 9 10

6 5 4 3 2 1 7 8 9 10

5 6 4 3 2 1 7 8 9 10

5 4 6 3 2 1 7 8 9 10

5 4 3 6 2 1 7 8 9 10

5 4 3 2 6 1 7 8 9 10

5 4 3 2 1 6 7 8 9 10

4 5 3 2 1 6 7 8 9 10

4 3 5 2 1 6 7 8 9 10

4 3 2 5 1 6 7 8 9 10

4 3 2 1 5 6 7 8 9 10

3 4 2 1 5 6 7 8 9 10

3 2 4 1 5 6 7 8 9 10

3 2 1 4 5 6 7 8 9 10

2 3 1 4 5 6 7 8 9 10

2 1 3 4 5 6 7 8 9 10

1 2 3 4 5 6 7 8 9 10

**Observação:** exiba todos os valores do vetor na linha, seguido imediatamente de <enter> (‘\n’ ou std::endl). Entre cada valor de um vetor em uma linha, deve haver APENAS um espaço.

1. Desenvolva um programa em linguagem C/C++ que receba dez valores inteiros. Em seguida ordene os valores utilizando o algoritmo de ordenação *SelectionSort*. Cada vez que o algoritmo detectar que dois valores devem mudar de posição o programa deverá exibir todos os valores do vetor. Ao final, o programa deverá exibir o vetor ordenado.

Exemplo:

**Entrada**

10 9 8 7 6 5 4 3 2 1

**Saída**

9 10 8 7 6 5 4 3 2 1

8 10 9 7 6 5 4 3 2 1

7 10 9 8 6 5 4 3 2 1

6 10 9 8 7 5 4 3 2 1

5 10 9 8 7 6 4 3 2 1

4 10 9 8 7 6 5 3 2 1

3 10 9 8 7 6 5 4 2 1

2 10 9 8 7 6 5 4 3 1

1 10 9 8 7 6 5 4 3 2

1 9 10 8 7 6 5 4 3 2

1 8 10 9 7 6 5 4 3 2

1 7 10 9 8 6 5 4 3 2

1 6 10 9 8 7 5 4 3 2

1 5 10 9 8 7 6 4 3 2

1 4 10 9 8 7 6 5 3 2

1 3 10 9 8 7 6 5 4 2

1 2 10 9 8 7 6 5 4 3

1 2 9 10 8 7 6 5 4 3

1 2 8 10 9 7 6 5 4 3

1 2 7 10 9 8 6 5 4 3

1 2 6 10 9 8 7 5 4 3

1 2 5 10 9 8 7 6 4 3

1 2 4 10 9 8 7 6 5 3

1 2 3 10 9 8 7 6 5 4

1 2 3 9 10 8 7 6 5 4

1 2 3 8 10 9 7 6 5 4

1 2 3 7 10 9 8 6 5 4

1 2 3 6 10 9 8 7 5 4

1 2 3 5 10 9 8 7 6 4

1 2 3 4 10 9 8 7 6 5

1 2 3 4 9 10 8 7 6 5

1 2 3 4 8 10 9 7 6 5

1 2 3 4 7 10 9 8 6 5

1 2 3 4 6 10 9 8 7 5

1 2 3 4 5 10 9 8 7 6

1 2 3 4 5 9 10 8 7 6

1 2 3 4 5 8 10 9 7 6

1 2 3 4 5 7 10 9 8 6

1 2 3 4 5 6 10 9 8 7

1 2 3 4 5 6 9 10 8 7

1 2 3 4 5 6 8 10 9 7

1 2 3 4 5 6 7 10 9 8

1 2 3 4 5 6 7 9 10 8

1 2 3 4 5 6 7 8 10 9

1 2 3 4 5 6 7 8 9 10

**Observação:** exiba todos os valores do vetor na linha, seguido imediatamente de <enter> (‘\n’ ou std::endl). Entre cada valor de um vetor em uma linha, deve haver APENAS um espaço.

1. Desenvolva um programa em linguagem C/C++ que receba dez valores inteiros. Em seguida ordene os valores utilizando o algoritmo de ordenação *InsertionSort*. Cada vez que o algoritmo detectar que dois valores devem mudar de posição o programa deverá exibir todos os valores do vetor. Ao final, o programa deverá exibir o vetor ordenado.

Exemplo:

**Entrada**

10 9 8 7 6 5 4 3 2 1

**Saída**

10 10 8 7 6 5 4 3 2 1

9 10 8 7 6 5 4 3 2 1

9 10 10 7 6 5 4 3 2 1

9 9 10 7 6 5 4 3 2 1

8 9 10 7 6 5 4 3 2 1

8 9 10 10 6 5 4 3 2 1

8 9 9 10 6 5 4 3 2 1

8 8 9 10 6 5 4 3 2 1

7 8 9 10 6 5 4 3 2 1

7 8 9 10 10 5 4 3 2 1

7 8 9 9 10 5 4 3 2 1

7 8 8 9 10 5 4 3 2 1

7 7 8 9 10 5 4 3 2 1

6 7 8 9 10 5 4 3 2 1

6 7 8 9 10 10 4 3 2 1

6 7 8 9 9 10 4 3 2 1

6 7 8 8 9 10 4 3 2 1

6 7 7 8 9 10 4 3 2 1

6 6 7 8 9 10 4 3 2 1

5 6 7 8 9 10 4 3 2 1

5 6 7 8 9 10 10 3 2 1

5 6 7 8 9 9 10 3 2 1

5 6 7 8 8 9 10 3 2 1

5 6 7 7 8 9 10 3 2 1

5 6 6 7 8 9 10 3 2 1

5 5 6 7 8 9 10 3 2 1

4 5 6 7 8 9 10 3 2 1

4 5 6 7 8 9 10 10 2 1

4 5 6 7 8 9 9 10 2 1

4 5 6 7 8 8 9 10 2 1

4 5 6 7 7 8 9 10 2 1

4 5 6 6 7 8 9 10 2 1

4 5 5 6 7 8 9 10 2 1

4 4 5 6 7 8 9 10 2 1

3 4 5 6 7 8 9 10 2 1

3 4 5 6 7 8 9 10 10 1

3 4 5 6 7 8 9 9 10 1

3 4 5 6 7 8 8 9 10 1

3 4 5 6 7 7 8 9 10 1

3 4 5 6 6 7 8 9 10 1

3 4 5 5 6 7 8 9 10 1

3 4 4 5 6 7 8 9 10 1

3 3 4 5 6 7 8 9 10 1

2 3 4 5 6 7 8 9 10 1

2 3 4 5 6 7 8 9 10 10

2 3 4 5 6 7 8 9 9 10

2 3 4 5 6 7 8 8 9 10

2 3 4 5 6 7 7 8 9 10

2 3 4 5 6 6 7 8 9 10

2 3 4 5 5 6 7 8 9 10

2 3 4 4 5 6 7 8 9 10

2 3 3 4 5 6 7 8 9 10

2 2 3 4 5 6 7 8 9 10

1 2 3 4 5 6 7 8 9 10

**Observação:** exiba todos os valores do vetor na linha, seguido imediatamente de <enter> (‘\n’ ou std::endl). Entre cada valor de um vetor em uma linha, deve haver APENAS um espaço.

1. Desenvolva um programa em linguagem C/C++ que receba dez valores inteiros. Em seguida ordene os valores utilizando o algoritmo de ordenação *MinMaxSort*. Cada vez que o algoritmo detectar que dois valores devem mudar de posição o programa deverá exibir todos os valores do vetor. Ao final, o programa deverá exibir o vetor ordenado.

Exemplo:

**Entrada**

10 9 8 7 6 5 4 3 2 1

**Saída**

1 9 8 7 6 5 4 3 2 10

1 2 8 7 6 5 4 3 9 10

1 2 3 7 6 5 4 8 9 10

1 2 3 4 6 5 7 8 9 10

1 2 3 4 5 6 7 8 9 10

**Observação:** exiba todos os valores do vetor na linha, seguido imediatamente de <enter> (‘\n’ ou std::endl). Entre cada valor de um vetor em uma linha, deve haver APENAS um espaço.