

1. Escreva um programa que leia do usuário um vetor de "n" números (o valor de "n" deve ser lido antes dos elementos do vetor). Em seguida, o programa deve imprimir na tela (a) os números do vetor e (b) o índice do menor número. Caso o menor número ocorra mais de uma vez no vetor, o programa pode imprimir o índice de qualquer uma das ocorrências.
2. Escreva uma variação do programa anterior na qual, depois de lido o vetor, o programa rearranje os elementos de forma que o menor número fique na primeira posição do vetor; o vetor deve ser impresso na tela antes e depois desse rearranjo (você escolhe se o índice do menor número será impresso).

Assim, por exemplo, se o usuário digitar o vetor [ 20 6 13 9 11 5 1 25 16 ], então o vetor após o rearranjo pode estar em vários estados, incluindo

[ 1 6 13 9 11 5 20 25 16 ] e [ 1 20 6 13 9 11 5 16 25 ].

Observação: caso o menor número ocorra mais de uma vez no vetor digitado, então, ao final, basta que a primeira posição do vetor possua uma dessas ocorrências.

3. Escreva uma variação do programa anterior na qual, após a digitação do vetor, (a) o menor elemento seja, como antes, de alguma forma transferido para a primeira posição do vetor, e, em seguida, (b) o 2º menor elemento seja, de alguma forma, transferido para a segunda posição do vetor. O vetor deve ser impresso na tela após cada "transferência".

Assim, por exemplo, se o vetor digitado for [ 20 6 13 9 11 5 1 25 16 ], então o conteúdo impresso pelo programa pode ser, dentre outras possibilidades, o seguinte:

[ 20 6 13 9 11 5 1 25 16 ]      ou      [ 20 6 13 9 11 5 1 25 16 ]  
[ 1 6 13 9 11 5 20 25 16 ]                      [ 1 20 6 13 9 11 5 16 25 ]  
[ 1 5 13 9 11 6 20 25 16 ]                      [ 1 5 20 6 13 9 11 16 25 ].

Dica: observe que o 2º menor elemento do vetor pode ser entendido como o menor elemento dentre aqueles que estão nas posições 1 a n-1 do vetor após a primeira "transferência".

4. Escreva uma variação do programa anterior na qual, após a digitação do vetor:
  - a) O menor número seja colocado na 1ª posição do vetor.
  - b) Em seguida, o 2º menor número seja colocado na 2ª posição do vetor.
  - c) Em seguida, o 3º menor número seja colocado na 3ª posição do vetor.Como antes, o vetor deve ser impresso na tela após cada "transferência".
5. Escreva uma variação do programa anterior que, após a digitação do vetor, ordene os números em ordem crescente. O vetor ordenado deve ser impresso ao final (você escolhe se os passos intermediários da ordenação serão impressos ou não).

6. Escreva um programa que leia do usuário (a) um vetor de "n" números e (b) um dos números do vetor, que chamaremos de "x". Em seguida, o programa deve rearranjar os números do vetor, de forma que à esquerda de "x" fiquem apenas os números menores ou iguais a "x", e à direita de "x" fiquem apenas os números maiores que "x".

Assim, por exemplo, se o vetor for [ 20 6 13 9 11 5 1 25 16 ] e  $x = 9$ , então o vetor rearranjado pode ficar, dentre outras possibilidades, assim:

[ 5 6 1 9 11 20 13 25 16 ].

Observação: naturalmente, se o programa ordenar o vetor, então a tarefa solicitada terá sido realizada, mas a questão é que o enunciado exige muito menos do que uma ordenação, e portanto a resposta pode ser um programa que execute muito mais rapidamente do que executaria se fosse ordenar o vetor inteiro (a diferença pode ser difícil de perceber para vetores pequenos, mas pode ser percebida com vetores grandes).