Aluno: Erivaldo Barros

Construa um algoritmo de ordenação, utilizando o método *bubble sort* estudado. (Lembre-se que se trata de uma série de instruções que devem ser seguidas passo a passo).

Para isso, você deve criar um método em que o tamanho do vetor seja dez e deve estar em ordem crescente.

O vetor deverá:

- comparar seus elementos dois a dois adjacentes;

- se os elementos não estiverem em ordem, então ordenar;

- senão, avançar para o próximo par;

- repetir a operação até que nenhuma troca possa ser feita no vetor inteiro.

**Var**

random : vetor [1..10] de inteiro

loop1, troca : inteiro

ordem : logico

**Inicio**

*//loop para alocar numeros digitados do usuário.*

para loop1 de 1 ate 10 faca

escreva("Digite qualquer número inteiro [",loop1," de 10 ]: ")

leia(random[loop1])

fimpara

*//declarando ordem como verdadeiro para a condição de repetição*

ordem <- verdadeiro

enquanto ordem = verdadeiro faca

ordem <- falso *//condição quando o loop PARA finalizar a ordem, parar o loop ENQUANTO.*

para loop1 de 1 ate 9 faca *//ate 9 porque não tem como comparar 10 com outra posição.*

se random[loop1] > random[loop1+1] entao *//se a posição atual for maior que a próxima, será aplicado os valores abaixo.*

troca <- random[loop1] *//variavel troca, para armazenar o maior valor e devolve-lo para proxima posição.*

random[loop1] <- random[loop1+1] *//atribuindo a posição menor na maior.*

random[loop1+1] <- troca *//devolvendo a posição menor a posição maior, e finalizando a troca*

ordem <- verdadeiro *//para manter o loop enquanto funcionando, enquanto houver trocas de posições, se essas trocas acabarem, então o loop enquanto acaba, pois foi declarado falso no início dele.*

fimse

fimpara

fimenquanto

limpatela

escreval("")

escreval("Ordenando números digitados...")

escreval("")

para loop1 de 1 ate 10 faca *//mais um loop na mesma variável loop1, para poder printar as posições ordenadas anteriormente...*

escreval(loop1,"º -",random[loop1])

fimpara

**Fimalgoritmo**