/\*Faça um algoritmo de ordenação utilizando o método insertion sort.

Crie um método que execute as seguintes operações:

- Tamanho do vetor: 30;

- Utilize números ímpares;

- Opere em ordem crescente.

\*/

#include <iostream>

#include <stdlib.h>

#define TAM 30

void imprimeVetor(int vetor[]){

int w;

printf("\n");

for(w = 0; w < TAM; w++){

printf(" | %d | ", vetor[w]);

}

}

void insertion\_sort(int vetor[TAM]){

int i, j, atual;

for(i = 1; i < TAM; i++){

//Elemento atual em análise

atual = vetor[i];

//Elemento anterior ao analisado

j = i - 1;

//Analisando membros anteriores

while( (j >=0) && (atual < vetor[j])){

//Posiciona os elmeentos uma posição para frente

vetor[j + 1] = vetor[j];

//Faz o j andar para trás

j = j - 1;

}

//Agora que o espaço foi aberto, colocamos o atual (Menor número) na posição correta

vetor[j + 1] = atual;

//Mostra a lista atualizada

}

}

void bubble\_sort(int vetor[TAM]){

//Auxiliares

int x, y, aux;

//Valor da esquerda sendo analisado

for(x = 0; x < TAM; x++){

//Valor da direita sendo analisado

for(y = x + 1; y < TAM; y++){

//Confere se precisa fazer a troca

if(vetor[x] < vetor[y]){

aux = vetor[x];

vetor[x] = vetor[y];

vetor[y] = aux;

}

}

}

}

int main(){

int vetor[TAM];

vetor[0]=1;

for(int k=1;k<TAM;k++){

if (k%2 == 0){

vetor[k] = 2\*k+1;

}

vetor[k]= k+(k+1);

}

// Metodo para inverter o vetor de numerosw impares

bubble\_sort(vetor);

printf("Vetor com 30 numeros impares decrescentes\n");

imprimeVetor(vetor);

printf("\n");

insertion\_sort(vetor);

printf("\nVetor ordenado de modo crescente pelo metodo insert sort:\n");

imprimeVetor(vetor);

return 0;

}