Logotipo, nombre de la empresa

Descripción generada automáticamenteMétodo de Insercion

Nombre: Oscar Alejandro Penilla Skakievich

Fecha: 28/08/22

Grupo:5D1

Registro: 20300701

**Descripción del programa:**

Se va a crear un programa que ordene un conjunto de números enteros usando el método de Selección, los datos se los dará el usuario y este mismo decidirá hasta cuantos ingresar dentro de un rango de 1 a 50. Este en su diferencia hacia los demás será que los datos conforme se agreguen serán ordenados por lo cual se actualizaría el arreglo, claramente, dentro del rango, y otra opción será cuando el usuario ingrese todos los datos y después este arreglo de ordene, además de que en cada inserción que será el proceso se deberá mostrar el arreglo.

**Requerimientos:**

1.Ordenará un conjunto de números enteros, la cantidad mínima deberá ser 1 y el máximo 50.

2.Deberá ingresar los valores correspondientes.

a. Deberá tener dos opciones de ordenación (Elija como preguntar esta opción).

b. La primera opción los datos serán ordenados como van siendo introducidos(elija un valor el cual detendrá el ingreso de datos).

a. La segunda opción la ordenación se realizará después de ingresados los datos (véase archivo adjunto).

**b. Sin importar cual caso sea se deberá mostrar el arreglo en cada inserción.**

c. El o los módulos de ordenación deberán ser métodos.

d. El sistema está diseñado en C++.

e. La ordenación será de menor a mayor.

f. Mostrar el mensaje de error correspondiente a cada posibilidad.

**PSP:**

|  |  |  |  |
| --- | --- | --- | --- |
| Código fuente: | Estimado | Real | Error |
| (min) | (min) |
| Tiempo de Diseño | 20 | 40 | 100% |
| Tiempo de Codificación. | 120 | 160 | 33% |
| Tiempo estimado de fabricación | 120 | 180 | 50% |
| Total, de líneas de código nuevas. | 20 | 33 | 65% |
| Total, de líneas de código reusadas. | 100 | 65 | 35% |
| Total, de líneas de código del programa. | 120 | 98 | 18% |
| Cantidad de errores de compilación. | 50 | 40 | 20% |
| Cantidad de errores de ejecución. | 50 | 20 | 60% |

**Código fuente:**

// Seleccion bien.cpp : Este archivo contiene la función "main". La ejecución del programa comienza y termina ahí.

//insercion

//Oscar Alejandro Penilla 20300701

#include <iostream>

using namespace std;

class numeros

{

public:

int cantidad = 0;

char ingreso = ' ';

int arreglo[50];

//constructores y destructures y funcion

numeros(int, int\*);

~numeros();

void insercion(int\*, int);

};

numeros::numeros(int, int\*)// en este programa va a haber dos metodos

{

numeros::cantidad = cantidad;

numeros::arreglo[50] = arreglo[50];

}

numeros::~numeros()

{

}

//este solo ordena y muestra las iteraciones

void numeros::insercion(int\* arreglo, int cantidad) {

int aux = 0, pos = 0;

for (int i = 0; i <= cantidad - 1; i++)

{

pos = i;//va a ser la posicion

aux = arreglo[i];//una variable auxiliar depende de i

while (pos > 0 && (arreglo[pos - 1] > aux)) {//mientras posicion sea mayor a 0 y el numero del arreglo sea mayor que aux

arreglo[pos] = arreglo[pos - 1];

pos--;//reducimos la posicion para el siguiente ciclo

}

arreglo[pos] = aux;//se hace el intercambio de variable

cout << "Iteracion " << i << ": ";//mostramos

for (int j = 0; j < cantidad; j++)

{//"Sin importar cual caso sea se deberá mostrar el arreglo en cada inserción."

cout << arreglo[j] << " ";

}

cout << "\n";

}

}

int main()

{

int cant = 0;

char ingreso = 'S';

int arreglo[50];

do

{//esto se puede reducir con un simple if y la propia funcion ya sabria que soltar

cout << "A-Quiere ingresar los valores antes \no \nB- Ingresarlos simultaneamente que se ordenen?" << endl;

cin >> ingreso;

switch (ingreso)

{

case 'A':

{

do

{

cout << "Cual es la cantidad de digitos que quieres ingresar? Min 1 max 50" << endl;

cin >> cant;

if (cant < 1 || cant > 50) {

cout << "Valor fuera de rango\n";

}

} while (cant < 1 || cant > 50);

cout << "Ingresa los datos: " << endl;

for (int i = 0; i < cant; i++)

{

cin >> arreglo[i];

}

numeros ob1(cant, arreglo);

ob1.insercion(arreglo, cant);

}

break;

case 'B':

{

int i = 0;

do

{

cout << "Introduzca un numero(-1000 para salir)\nNumeros ingresados: " << i << ": ";

cin >> arreglo[i];

if (arreglo[i] == -1000) {

cout << "Ha salido\n";

i = 51;//sale

}

else {//entra a ordenarse

numeros ob1(cant, arreglo);

ob1.insercion(arreglo, i + 1);

i++;

}

} while (i <= 50);

}

break;

}

} while (ingreso != 'A' && ingreso != 'B');

}