Logotipo, nombre de la empresa

Descripción generada automáticamenteMétodo de Shell

Nombre: Oscar Alejandro Penilla Skakievich

Fecha: 02/09/22

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. El programa también podrá mostrar o no las iteraciones en cada vez que se repita para que el usuario pueda ver el proceso de su arreglo

Este lo que lo caracteriza es que en menos pasos cubre mas posibilidades ya que este depende de saber cuantos valores se le son agregados y así poderlos ordenar en pequeños grupos

**Requerimientos:**

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

b) Deberá ingresar los valores correspondientes.

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

a. La primera opción después de ingresar los datos estos se muestran ordenados.

b. La segunda opción mostrará todas cada una de las iteraciones del método.

d)El o los módulos de ordenación deberán ser métodos.

e) El sistema está diseñado en c++.

f) La ordenación será de menor a mayor.

g) Mostrar el mensaje de error correspondiente a cada posibilidad.

**PSP:**

|  |  |  |  |
| --- | --- | --- | --- |
| Contenido del PSP – Shell | | | |
| Código fuente: | Estimado | Real | Error |
| (min) | (min) |
| Tiempo de Diseño | 20 | 25 | 25% |
| Tiempo de Codificación. | 60 | 57 | 5% |
| Tiempo estimado de fabricación | 70 | 67 | 4% |
| Total, de líneas de código nuevas. | 50 | 41 | 18% |
| Total, de líneas de código reusadas. | 70 | 52 | 26% |
| Total, de líneas de código del programa. | 130 | 93 | 28% |
| Cantidad de errores de compilación. | 30 | 20 | 33% |
| Cantidad de errores de ejecución. | 15 | 10 | 33% |

**Código fuente:**

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

//Shell

//Oscar Alejandro Penilla 20300701

#include <iostream>

#include <cmath>

using namespace std;

class numeros

{

public:

int cantidad;

char iteracion;

int arreglo[50];

//constructores y destructures y funcion

numeros(int, int\*, char);

~numeros();

void shell(int\*, int, char);

};

numeros::numeros(int, int\*, char)

{

numeros::iteracion=' ';

numeros::cantidad = 0;

}

numeros::~numeros() { }

void numeros::shell(int\* arreglo, int cantidad, char mostrar) {

int i, a=1, aux;

bool b\_door;

int i\_redond = cantidad;

while (i\_redond > 1)//comenzamos a ordenar

{

i\_redond = ceil(float(i\_redond)/2);

b\_door = true;

while (b\_door == true)

{

b\_door = false;

i = 0;

while ((i + i\_redond) <= cantidad)

{

if (arreglo[i] > arreglo[i + i\_redond])

{

aux = arreglo[i];

arreglo[i] = arreglo[i + i\_redond];

arreglo[i + i\_redond] = aux;

b\_door = true;

}

i++;

}//aqui no se muestra

}//aqui se puede mostrar

if (mostrar == 'S') {

cout << "iteracion " << a << ": ";//para mostrar el arreglo antes completo

a++;

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

{

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

}

cout << "\n";

}

}

//para solo mostrar una vez nuestro arreglo

if (mostrar == 'N') {//en caso de que no se repita

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

cout << arreglo[i] << " "; //completo

}

}

int main()

{

int cant = 0;

char iteracion = 'S';

int arreglo[50];

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];

}

do

{

cout << "Quieres ver la iteracion? (S/N)" << endl;

cin >> iteracion;

numeros ob1(cant, arreglo, iteracion);

ob1.shell(arreglo, cant, iteracion);

} while (iteracion != 'S' && iteracion != 'N');

}