Logotipo

Descripción generada automáticamente

# Método de RADIX

# 

**Fecha:** 14/03/2023

**Materia:** Estructura de datos

**Profesor/Profesora:** Ismael López Buenrostro

**Nombre:** Luis Rigoberto Ramírez Llamas

**Registro:** 21100298

**Carrera:** Drllo. Software

**Grupo:** 4E1

**Descripción del programa**

Este programa es un código en C++ a través de visual studio, en donde se introduce un conjunto 1 a 50 de números enteros ya sean ordenados o desordenados, positivos con la finalidad de ordenarlos con el método de radix

En este código se usa una clase Métodos y una función Radix a través de métodos y funciones gracias a la implementación de un arreglo dinámico, lastimosamente este código colapsa cuando se introduce un numero negativo (Por no leer completamente antes de programarlo)

**Lista de requerimientos**

Fabricar un programa que realice el proceso de ordenación por el método de Radix el cual deberá tener las siguientes características.

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

b)El rango de datos deberá ser entre1000-9999o-9999 y -1000.

c)Deberá ingresar los valores correspondientes.

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

a.Los valores que se van a ordenar serán positivos.

b.Los valores que se van a ordenar serán negativos.

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

f)El sistema está diseñado en c++.

g)La ordenación será de menor a mayor.

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

**Contenido del PSP**

|  |  |  |  |
| --- | --- | --- | --- |
| Código fuente: | Estimado  (min) | Real  (min) | Error |
| Tiempo de Diseño | 60 | 80 | 33% |
| Tiempo de Codificación. | 200 | 250 | 25% |
| Tiempo estimado de fabricación | 260 | 330 | 27% |
| Total, de líneas de código nuevas. | 60 | 62 | 3% |
| Total, de líneas de código reusadas. | 30 | 26 | 13% |
| Total, de líneas de código del programa. | 90 | 86 | 4% |
| Cantidad de errores de compilación. | 30 | 20 | 33% |
| Cantidad de errores de ejecución. | 15 | 9 | 40% |

**Código Fuente**

#include <iostream>

#include <cmath>

using namespace std;

class Metodos

{

public:

void Radix(int Numeros[50], int Tam, char Respuesta);

};

void Metodos::Radix(int Numeros[50], int Tam, char Respuesta)

{

int NumMax = Numeros[0];

for (int i = 1; i < Tam; i++)

{

if (Numeros[i] > NumMax)

{

NumMax = Numeros[i];

}

}

int PosDig = 1;

int\* SubNum = new int[Tam];

while (NumMax / PosDig > 0)

{

int DigNum[10] = { 0 };

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

{

int Dig = (Numeros[i] / PosDig) % 10;

DigNum[Dig]++;

}

for (int i = 1; i < 10; i++)

{

DigNum[i] += DigNum[i - 1];

}

for (int i = Tam - 1; i >= 0; i--)

{

int Dig = (Numeros[i] / PosDig) % 10;

SubNum[DigNum[Dig] - 1] = Numeros[i];

DigNum[Dig]--;

}

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

{

Numeros[i] = SubNum[i];

}

PosDig \*= 10;

if (Respuesta == 'S')

{

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

{

cout << "\t" << Numeros[i];

}

cout << endl;

}

}

if (Respuesta == 'N')

{

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

{

cout << "\t" << Numeros[i];

}

}

}

int main()

{

Metodos Rad;

int Numeros[50];

int Tam;

char R;

cout << "Desea ver el proceso de ordenamiento? \n[S] Si\n[N] No" << endl;

cin >> R;

cout << "Cuantos numeros vas a ingresar: " << endl;

cin >> Tam;

if (Tam <= 0 || Tam > 50)

{

cout << "Tamaño no compatible" << endl;

return 0;

}

else

{

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

{

cout << "Introduce el " << i + 1 << " numero: " << endl;

cin >> Numeros[i];

}

}

cout << "Estos son tus numeros ordenados : \n" << endl;

Rad.Radix(Numeros, Tam, R);

return 0;

}