Oscar Andres Perez Carmona - 2162110

Yeferson Valencia Vanegas - 2200134

1. Compilarlo usando usando gcc y ejecutarlo.

2¿Qué observa en el comportamiento del código tanto al compilarlo como al ejecutar?

Hay variaciones de tiempo

| banderas | Tiempo real |
| --- | --- |
| O | 3.734 seg |
| O0 | 3.174 seg |
| O1 | 2.564 seg |
| O2 | 2.163 seg |
| O3 | 2.365 seg |
| **3-¿Qué pasa si usa las opciones -O, -O0, -Ofast, -Og y Oz?** |  |
| O | 4.766 seg |
| Ofast | 3.203 seg |
| Og | 2.898 seg |
| Oz | 2.998 |
| dmain | 4.219 |

Generalmente, a medida que se aumenta el nivel de optimización, el compilador realiza más transformaciones en el código para intentar mejorar su rendimiento lo que queda en evidencia al observar los tiempos de ejecución aunque investigando por internet también puede hacer que la depuración sea más difícil y aumentar los tiempos de compilación.

4. Busqué por internet un código de ejemplo simple que use punteros y repita los puntos del 1 al 3. NO PUEDEN HABER DOS CÓDIGOS IGUALES POR PAREJAS DEL CURSO POR LO QUE DEBEN PONERSE DE ACUERDO CON SUS COMPAÑEROS DE NO REPETIR.

#include <iostream>

using namespace std;

void sumarMatrices(int \*matrizA, int \*matrizB, int \*resultado, int filas, int columnas) {

for (int i = 0; i < filas; ++i) {

for (int j = 0; j < columnas; ++j) {

\*(resultado + i \* columnas + j) = \*(matrizA + i \* columnas + j) + \*(matrizB + i \* columnas + j);

}

}

}

int main() {

int filas, columnas;

cout << "Ingrese el número de filas de las matrices: ";

cin >> filas;

cout << "Ingrese el número de columnas de las matrices: ";

cin >> columnas;

int \*matrizA = new int[filas \* columnas];

int \*matrizB = new int[filas \* columnas];

int \*resultado = new int[filas \* columnas];

cout << "Ingrese los elementos de la matriz A:\n";

for (int i = 0; i < filas; ++i) {

for (int j = 0; j < columnas; ++j) {

cin >> \*(matrizA + i \* columnas + j);

}

}

cout << "Ingrese los elementos de la matriz B:\n";

for (int i = 0; i < filas; ++i) {

for (int j = 0; j < columnas; ++j) {

cin >> \*(matrizB + i \* columnas + j);

}

}

sumarMatrices(matrizA, matrizB, resultado, filas, columnas);

cout << "La suma de las matrices es:\n";

for (int i = 0; i < filas; ++i) {

for (int j = 0; j < columnas; ++j) {

cout << \*(resultado + i \* columnas + j) << " ";

}

cout << endl;

}

// Liberar memoria

delete[] matrizA;

delete[] matrizB;

delete[] resultado;

return 0;

}

La inclusión de las banderas de optimización al compilar el código puede tener un impacto significativo en el rendimiento del programa. Al activar esta optimización, el compilador GCC aplica una serie de técnicas de optimización básicas al código, lo que puede resultar en un programa compilado que se ejecuta más rápido y/o utiliza menos recursos del sistema. En el contexto del código para sumar matrices utilizando punteros la aplicación de la bandera mejora la eficiencia de las operaciones de bucles y cálculos aritméticos, lo que potencialmente reduciría el tiempo de ejecución del programa.