Se puede editar el código si problema si desea hacer algún comentario, solo que al copiarlo de vscode parece una imagen pero si es texto.

/\*Primer laboratorio: Escriba un algoritmo que presente un menú en donde dependiendo de la escogencia genere 1. Una cantidad finita de números enteros aleatorios y 2. Una cantidad de colores y sean mostrados por la pantalla, debe usar la función random\*/

#include <iostream>

#include <time.h>

#include <stdlib.h>

#include <windows.h>

#include <conio.h>

using namespace std;

//Funcion que cambia el color del texto en funcion del numero que se le pase

void cambiarColor(int color){

    HANDLE hConsole;

    hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

    SetConsoleTextAttribute(hConsole, color);

}

//Funcion que devuelve el nombre del color en funcion del numero que se le pase

string ObtenerNombreColor(int color){

    switch (color)

    {

    case 1:

        return "Azul";

        break;

    case 2:

        return "Verde";

        break;

    case 3:

        return "Aguamarina";

        break;

    case 4:

        return "Rojo";

        break;

    case 5:

        return "Morado";

        break;

    case 6:

        return "Amarillo";

        break;

    case 7:

        return "Blanco";

        break;

    case 8:

        return "Gris";

        break;

    case 9:

        return "Azul claro";

        break;

    case 10:

        return "Verde claro";

        break;

    case 11:

        return "Aguamarina claro";

        break;

    case 12:

        return "Rojo claro";

        break;

    case 13:

        return "Morado claro";

        break;

    case 14:

        return "Amarillo claro";

        break;

    case 15:

        return "Blanco brillante";

        break;

    default:

        return "Color no valido";

        break;

    }

}

int main(){

    int opcion;

    int cantidad = 0;

    //Bucle para mostrar el menu y ejecutar las opciones hasta que el usuario elija salir

    while(opcion != 3){

        /\* Se muestran las opciones del menu en pantalla \*/

        system("cls");

        cout<<"Bienvenido... Ingresa la opcion que deseas ejecutar:\n\n";

        cout<<"1. Generar rango de numeros enteros aleatorios.\n";

        cout<<"2. Generar cantidad de colores.\n";

        cout<<"3. Salir";

        cout<<"\n\n";

        /\* Se pide la opcion al usuario \*/

        cin>>opcion;

        cout<<"\n\n";

        //Switch para ejecutar un procedimiento segun la opcion seleccionada

        switch (opcion)

        {

        //Generando tantos numeros aleatorios como indique el usuario

        case 1:

            srand(time(NULL));

            cout<<"Elige la cantidad de numeros aleatorios que quiere generar: ";

            cin>> cantidad;

            cout<<"Numeros aleatorios: \n\n";

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

                int numAleatorio = (1 + rand() % 10);

                cout<<"Numero generado: " << numAleatorio << "\n";

            }

            cout<<"\nPresiona cualquier tecla para continuar...\n";

            getch();

            break;

        //Generando tantos colores aleatorios como indique el usuario

        case 2:

            srand(time(NULL));

            cout<<"Elige la cantidad de colores que quieres generar: ";

            cin>> cantidad;

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

                int color = (1 + rand() % 15);

                cambiarColor(color);

                cout<<"Color generado: " << ObtenerNombreColor(color) << "\n";

            }

            cambiarColor(7);

            cout<<"\nPresiona cualquier tecla para continuar...\n";

            getch();

            break;

        case 3:

        cout<<"Gracias, tenga buen dia\n\n";

        break;

        default:

            cout << "Opcion no valida\n";

            cout<<"\nPresiona cualquier tecla para continuar...\n";

            getch();

            break;

        }

    }

}

/\*Segundo laboratorio parte 1: Imaginemos que el restaurant tiene en total N platos Suponiendo que   hay platos, de color 1 (rojo), otro de color 2 Azul  y color 3 (amarillo) y que alguien lava platos  y los coloca uno sobre otro  y que los mesoneros  toman  de los platos  que se lavan para servir. Escriba un programa en C++ que simule el proceso anterior de lavar y servir, utilizando para ello una Pila, y muestre en un momento determinado cuantos platos hay disponible y de que color son esos platos

\*/

#include <iostream>

#include <stack>

#include <windows.h>

#include <cstdlib>

#include <time.h>

#include <conio.h>

using namespace std;

void cambiarColor(int color){

    HANDLE hConsole;

    hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

    SetConsoleTextAttribute(hConsole, color);

}

int main(){

    stack<int> pila;

    int num\_platos;

    int counter\_rojo = 0, counter\_amarillo = 0, counter\_azul = 0;

    srand(time(NULL));

    cout<<"Ingrese la cantidad de platos limpios que hay apilados: ";

    cin>>num\_platos;

    cout<<"\n";

    if (num\_platos <= 0){

        cout<<"No hay platos, no es posible continuar\n\n";

        return 0;

    }

    cout<<"En el momento en que quieras pausar la simulacion presiona cualquier tecla.\n\n";

    while (!kbhit()){

        inicio:

        int accion = rand() % 2 + 1;

        if(accion == 1 && !pila.empty()){

            int numero = pila.top();

            string mostrar;

            switch (numero){

            case 1:

                mostrar = "rojo";

                cambiarColor(4);

                counter\_rojo -= 1;

                break;

            case 2:

                mostrar = "azul";

                cambiarColor(1);

                counter\_azul -= 1;

                break;

            case 3:

                mostrar = "amarillo";

                cambiarColor(6);

                counter\_amarillo -= 1;

                break;

            }

            pila.pop();

            cout<<"Se uso un plato " << mostrar << ".";

            cout<<"\n";

        }

        else if(accion == 2 && pila.size() < num\_platos){

            pila.push(rand() % 3 + 1);

            int numero = pila.top();

            string introducido;

            switch (numero){

            case 1:

                introducido = "rojo";

                cambiarColor(4);

                counter\_rojo += 1;

                break;

            case 2:

                introducido = "azul";

                cambiarColor(1);

                counter\_azul += 1;

                break;

            case 3:

                introducido = "amarillo";

                cambiarColor(6);

                counter\_amarillo += 1;

                break;

            }

            cout<<"Se lavo un plato " << introducido << ".";

            cout<<"\n";

        }

        Sleep(500);

    }

    cambiarColor(7);

    cout<<"\nQuedan " << pila.size() << " platos en la pila\n";

    cambiarColor(4);

    cout<<"Quedaron " << counter\_rojo << " platos rojos\n";

    cambiarColor(1);

    cout<<"Quedaron " << counter\_azul << " platos azules\n";

    cambiarColor(6);

    cout<<"Quedaron " << counter\_amarillo << " platos amarillos\n\n";

    cambiarColor(7);

    int choice = 0;

    cout<<"1. Continuar con la simulacion\n";

    cout<<"2. Salir\n\n";

    cin>> choice;

    switch (choice){

        case 1:{

            goto inicio;

            break;

        }

        case 2:{

            cout<<"Tenga buen dia\n";

            break;

        }

        default:{

            cout<<"Opcion incorrecta, continuara la simulacion del programa.\n";

            Sleep(1000);

            goto inicio;

            break;

        }

    }

    return 0;

}

/\*Segundo laboratorio parte 2: Utilizando una Pila escriba un programa en C++ que evalue una expresión algebraica ingresada y determine si los separadores se encuentran debidamente balanceado \*/

#include <iostream>

#include <stack>

using namespace std;

int main() {

    stack<char> pila;

    string operacion;

    bool balanceada = true;

    cout << "Ingrese la operacion algebraica: ";

    cin >> operacion;

    // Iterar sobre cada carácter de la operación

    for (size\_t i = 0; i < operacion.size(); i++) {

        char c = operacion[i]; // Carácter actual

        if (c == '(' || c == '{' || c == '[') {

            // Apilar aperturas

            pila.push(c);

            cout << "Apilando: " << c << endl;

        } else if (c == ')' || c == '}' || c == ']') {

            cout << "Apilando: " << c << endl;

            // Verificar cierres

            if (pila.empty()) {

                balanceada = false;

                cout << "Error: Parentesis de cierre sin apertura previa: " << c << endl;

                break;

            }

            char tope = pila.top();

            pila.pop(); // Desapilar

            cout << "Desapilando: " << c << " y " << tope << endl;

            // Verificar

            if ((c == ')' && tope != '(') ||

                (c == '}' && tope != '{') ||

                (c == ']' && tope != '[')) {

                balanceada = false;

                cout << "Error: Parentesis no coinciden: " << tope << " y " << c << endl;

            }

        } else {

            // Ignorar otros caracteres

            cout << "Apilando: " << c << endl;

            cout << "desapilando: " << c << endl;

        }

    }

    // Verificar si quedan aperturas sin cerrar

    if (!pila.empty()) {

        balanceada = false;

        while (!pila.empty()) {

            char elemento = pila.top();

            pila.pop();

            cout << "Error: Parentesis de apertura sin cerrar: " << elemento << endl;

        }

    }

    // Mostrar resultado final

    if (balanceada) {

        cout << "La operacion esta balanceada." << endl;

    } else {

        cout << "La operacion no esta balanceada." << endl;

    }

    return 0;

}

/\*Laboratorio 3 parte 1: Utilizando solo 1 arreglo. Y dada la siguiente definición de la estructura

Typedef divisor = array [n] of integer;

Elabore un algoritmo en C, que almacene en esa estructura y solo en esa los números que

ingresan por teclado y sus 5 primeros divisores\*/

#include <iostream>

#include<conio.h>

using namespace std;

int main (){

    int num,div = 5;

    cout<<"Digite cuantos numeros tendra el arreglo: ";

    cin>>num;

    int numero[6 \* num];

    int indice = 0;

    for(int i = 0 ; i<num; i++){

        int numeros;

        cout<<"Elementos para el arreglo: ";

        cin>>numeros;

        numero[indice++] = numeros;

    //calcular los divisores

    int contador = 0;

    for (int j = 1; j<= numeros && contador<div;j++ ){

        if (numeros%j == 0){

            numero[indice++]= j;

            contador++;

        }

    }

    while(contador<div){

        numero[indice++]=0;

        contador++;

    }

}

    cout << "\nContenido del arreglo:\n";

    for(int i = 0; i < indice; i++){

        cout<<numero[i]<<" ";

    }

    cout<<endl;

    return 0;

}

/\*Laboratorio 3 parte 2: Dada la siguiente definición de la estructura arreglo

Typedef cola = array [n] of interger;

Integer fin,inicio;

Escriba un algoritmo en C, que simule la atención de una cola en una taquilla

bancaria, cuya capacidad máxima es de 8 personas resuelva las colisiones (cuando no haya

espacio) y controle cuando la cola está llena o vacía\*/

#include <iostream>

#include <queue>

#include<time.h>

#include <conio.h>

#include <windows.h>

using namespace std;

//Funcion que muestra en pantalla el contenido de la cola

void print\_queue(queue<int> queue){

    while(!queue.empty()){

        cout<< "    " << queue.front() << "\n\n";

        queue.pop();

    }

}

int main(){

    queue<int> cola;

    int proceso, counter = 0;

    while (!kbhit()){

        srand(time(NULL));

        //Numero aleatorio entre el 1 y el 2

        proceso = rand() % 4 + 1;

        cout<<"Cola del banco: \n";

        cout<<"---------------\n\n";

            if(cola.size() < 8 && (proceso == 1 or proceso == 3)){

                counter++;

                cola.push(counter);

                print\_queue(cola);

            }

            else if(!cola.empty() && (proceso == 2 or proceso == 4)){

                cola.pop();

                print\_queue(cola);

            }

            else if(cola.size() > 7){

                cout<<"La cola esta llena\n\n";

                print\_queue(cola);

            }

            else if(cola.empty()){

                cout<< "No hay clientes en la cola \n\n";

            }

            if(counter >= 7){

                counter = 0;

            }

            Sleep(1000);

            system("cls");

    }

    //Mostrar contenido de la cola

    cout<<"Contenido de la cola:\n";

    while(!cola.empty()){

        cout<<cola.front()<<" ";

        cola.pop();

    }

}

/\*Laboratorio 4 parte 1: Usando Matrices. Elabore un programa que Ingrese

20 números mayores que 50. Calcule y almacene sus

5 primeros divisores. \*/

#include <iostream>

#include <conio.h>

#include <ctime>

#include <ctype.h>

const int NUMEROS = 20;

const int DIVISORES = 5;

const int menor = 50;

const int rango = 100;

using namespace std;

int main(){

    srand(time(0));

    int opcion, cont = 0;

    int num [NUMEROS][DIVISORES + 1];

    char continuar ='s';

    do{

    cout <<"\tIngrese la opcion que desee\n";

    cout <<"\t1.-Encontrar divisores con numeros randoms\n";

    cout <<"\t2.-Encontrar divisores con numeros colocados manualmente\n";

    cout <<"\t3.-Salir\n";

    cout <<"\tSu opcion es: ";

    cin >>opcion;

    system("cls");

    switch(opcion){

    case 1:

    do{

    // Generar números aleatorios

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

        num[i][0] = menor + 1 + rand() % rango;

        // Calcular divisores

        int cont = 0;

        for (int d = 1; d <= num[i][0] && cont < DIVISORES; d++) {

            if (num[i][0] % d == 0) {

                num[i][cont + 1] = d;

                cont++;

            }

        }

        // Completar con ceros si no hay 5 divisores

        while (cont < DIVISORES) {

            num[i][cont + 1] = 0;

            cont++;

        }

    }

        //Mostrar resultados

        cout << "Numeros | Divisores\n";

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

            cout <<num[i][0]<<"\t|";

            for(int j = 1; j <= DIVISORES; j++){

                cout<<num[i][j]<< " ";

            }

            cout<<endl;

    }

    cout <<"Desea continuar? (s/n): ";

    cin >>continuar;

    continuar = tolower(continuar);

    system("cls");

    }while( continuar == 's');

    break;

    case 2:

    do{

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

        cont = 0;

    do{

        cout <<"Ingrese los numeros (mayores a 50)\n";

        cin >>num[i][0];

        if(num[i][0] < menor){

            cout <<"\tDeben ser numeros mayores a 50.Intente de nuevo\n";

        }

    }while(num[i][0] < menor);

    for( int d= 1; d <= num[i][0] && cont < DIVISORES; d++){

        if(num[i][0] % d == 0){

            num[i][cont + 1] = d; //Almacenamos divisores de num[i][1]

            cont++;

        }

    } //Para rellenar con 0 si no hay 5 divisores

    while (cont < DIVISORES){

        num[i][cont + 1] = 0;

        cont++;

    }

}

 //Mostrar resultados

    cout << "Numeros | Divisores\n";

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

        cout <<num[i][0]<<"\t|";

        for(int j = 1; j <= DIVISORES; j++){

            cout<<num[i][j]<< " ";

        }

        cout<<endl;

}

    cout <<"Desea continuar? (s/n): ";

    cin >>continuar;

    continuar = tolower(continuar);

    system("cls");

    }while( continuar == 's');

    break;

    case 3:

    cout <<"Saliendo...";

    break;

    default:

    cout << "\tOpcion invalida\n";

    break;

}

    if (opcion != 3) {

    cout << "\nDesea volver al menu principal? (s/n): ";

    cin >> continuar;

    continuar = tolower(continuar);

    system("cls"); // Limpiar pantalla

}

    }while(opcion != 3 and (continuar == 's' or continuar == 'S'));

    getch();

    return 0;

}

/\*Laboratorio parte 2: Usando arreglo y registro. Elabore un programa que Ingrese 20 números mayores que 50 Calcule y almacene sus 5 primeros divisores.\*/

//Programa que usa registros y arreglos para almacenar 20 numeros enteros y calcular sus primeros 5 divisores

//Autores: Luis Flores y Jade Moya

#include <iostream>

#include <conio.h>

#include <windows.h>

#include <time.h>

using namespace std;

int main(){

    srand(time(NULL));

    //Declarando variables

    const int n = 20;

    const int rango = 1000 + 50;

    int num;

    char opc = '0';

    struct elemento{

        int valor;

        int divisor1;

        int divisor2;

        int divisor3;

        int divisor4;

        int divisor5;

    };

    elemento divisores[n];

    while (opc != '3'){

        //Inicializando los divisores en 0

        for (int i = 0; i < n; i++){

            divisores[i].valor = 0;

            divisores[i].divisor1 = 0;

            divisores[i].divisor2 = 0;

            divisores[i].divisor3 = 0;

            divisores[i].divisor4 = 0;

            divisores[i].divisor5 = 0;

        }

        //Menu para elegir funcionamiento del programa

        system("cls");

        cout<<"1. Elegir numeros de manera aleatoria \n";

        cout<<"2. Elegir numeros manualmente \n";

        cout<<"3. Salir \n\n";

        opc = getch();

        system("cls");

        switch (opc){

            case '1':

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

                    num = rand() % rango;

                    divisores[i].valor = num;

                    for (int j = 1; j <= num; j++){

                        if (num % j == 0){

                            if (divisores[i].divisor1 == 0){

                                divisores[i].divisor1 = j;

                            } else if (divisores[i].divisor2 == 0){

                                divisores[i].divisor2 = j;

                            } else if (divisores[i].divisor3 == 0){

                                divisores[i].divisor3 = j;

                            } else if (divisores[i].divisor4 == 0){

                                divisores[i].divisor4 = j;

                            } else if (divisores[i].divisor5 == 0){

                                divisores[i].divisor5 = j;

                            }

                        }

                    }

                }

                cout<<"Numeros generados correctamente... \n\n";

                cout<<"Mostrando valores del arreglo con sus divisores: \n\n";

                cout<<"Valor\tDivisor 1\tDivisor 2\tDivisor 3\tDivisor 4\tDivisor 5\n";

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

                    cout<<divisores[i].valor<<"\t"<<divisores[i].divisor1<<"\t\t"<<divisores[i].divisor2<<"\t\t"<<divisores[i].divisor3<<"\t\t"<<divisores[i].divisor4<<"\t\t"<<divisores[i].divisor5<<endl;

                }

                cout<<"\n\nPresione cualquier tecla para continuar... \n";

                getch();

                break;

            case '2':{

                //Ingresar numeros manualmente

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

                    cout<<"Ingrese el valor del elemento "<<i+1<<": ";

                    cin>>num;

                    if (num > 50){

                        divisores[i].valor = num;

                        for (int j = 1; j <= num; j++){

                            if (num % j == 0){

                                if (divisores[i].divisor1 == 0){

                                    divisores[i].divisor1 = j;

                                } else if (divisores[i].divisor2 == 0){

                                    divisores[i].divisor2 = j;

                                } else if (divisores[i].divisor3 == 0){

                                    divisores[i].divisor3 = j;

                                } else if (divisores[i].divisor4 == 0){

                                    divisores[i].divisor4 = j;

                                } else if (divisores[i].divisor5 == 0){

                                    divisores[i].divisor5 = j;

                                }

                            }

                        }

                    }

                    else{

                        cout<<"El numero debe ser mayor a 50. \n";

                        i--;

                    }

                }

                cout<<"Mostrando valores del arreglo con sus divisores: \n\n";

                cout<<"Valor\tDivisor 1\tDivisor 2\tDivisor 3\tDivisor 4\tDivisor 5\n";

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

                    cout<<divisores[i].valor<<"\t"<<divisores[i].divisor1<<"\t\t"<<divisores[i].divisor2<<"\t\t"<<divisores[i].divisor3<<"\t\t"<<divisores[i].divisor4<<"\t\t"<<divisores[i].divisor5<<endl;

                }

                cout<<"\n\nPresione cualquier tecla para continuar... \n";

                getch();

                break;

            }

            case '3':{

                cout<<"Saliendo del programa... \n";

                Sleep(1000);

                break;

            }

            default:{

                cout<<"Opcion no valida. Intente de nuevo. \n";

                Sleep(1000);

            }

        }

    }

}

/\*Primer parcial: Utilizando la Estructura Matrices escriba un programa que capture la siguiente tabla: Distrito Candidato A Candidato B Candidato C 1 194 48 206 2 180 20 320 3 221 90 821 4 432 50 946 Calcule sin utilizar más variables que la Matriz y sus subíndices los votos obtenidos por cada candidato y por región y que especifique quién ganó? revise si su programa funciona si en lugar de 4 distritos tengo 4000 distritos y 500 candidatos. \*/

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <iomanip>

using namespace std;

void imprimirCuadro(int\*\* matriz, int filas, int columnas) {

    // Imprimir encabezado de columnas

    cout << setw(10) << "Distrito";

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

        cout << setw(12) << "Candidato " << char('A' + j);

    }

    cout << endl;

    // Imprimir separador de encabezado

    cout << setw(10) << "--------";

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

        cout << setw(12) << "-----------";

    }

    cout << endl;

    // Imprimir valores de la matriz

    for (int i = 0; i < filas; i++) {

        cout << setw(10) << i + 1;

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

            cout << setw(12) << matriz[i][j];

        }

        cout << endl;

    }

}

void generarVotosAleatorios(int\*\* matriz, int filas, int columnas) {

    for (int i = 0; i < filas; i++) {

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

            matriz[i][j] = rand() % 1000; // Votos entre 0 y 999

        }

    }

}

int main() {

    srand(time(0)); // Inicializar semilla para números aleatorios

    int cantidad\_distrito = 0, cantidad\_candidatos = 0;

    int opcion;

    cout << "\tSISTEMA DE VOTACIONES" << endl;

    cout << "\nIngrese cuantos distritos hay: ";

    cin >> cantidad\_distrito;

    cout << "Ingrese cuantos candidatos hay: ";

    cin >> cantidad\_candidatos;

    // Crear la matriz dinámica

    int\*\* candidato = new int\*[cantidad\_distrito];

    for (int i = 0; i < cantidad\_distrito; i++) {

        candidato[i] = new int[cantidad\_candidatos]();

    }

    do {

        system("cls");

        cout << "\t\tMENU PRINCIPAL" << endl;

        cout << "1. Ingresar votos manualmente" << endl;

        cout << "2. Generar votos aleatorios" << endl;

        cout << "3. Mostrar resultados" << endl;

        cout << "4. Salir" << endl;

        cout << "Seleccione una opcion: ";

        cin >> opcion;

        switch(opcion) {

            case 1: {

                // Llenar la matriz manualmente

                for (int i = 0; i < cantidad\_distrito; i++) {

                    for (int j = 0; j < cantidad\_candidatos; j++) {

                        cout << "\nDistrito " << i + 1 << " - Candidato " << char('A' + j) << endl;

                        cout << "Ingrese la cantidad de votos: ";

                        cin >> candidato[i][j];

                    }

                }

                cout << "\nVotos ingresados correctamente!\n";

                system("pause");

                break;

            }

            case 2: {

                generarVotosAleatorios(candidato, cantidad\_distrito, cantidad\_candidatos);

                cout << "\nVotos aleatorios generados exitosamente!\n";

                system("pause");

                break;

            }

            case 3: {

                // Mostrar la matriz

                cout << "\nCUADRO DE VOTOS" << endl << endl;

                imprimirCuadro(candidato, cantidad\_distrito, cantidad\_candidatos);

                // Calcular el ganador del candidato

                int mayor\_candidato = 0;

                int votos\_mayor\_candidato = 0;

                for (int j = 0; j < cantidad\_candidatos; j++) {

                    int suma\_candidato = 0;

                    for (int i = 0; i < cantidad\_distrito; i++) {

                        suma\_candidato += candidato[i][j];

                    }

                    if (suma\_candidato > votos\_mayor\_candidato) {

                        votos\_mayor\_candidato = suma\_candidato;

                        mayor\_candidato = j;

                    }

                    cout << "\nTotal votos Candidato " << char('A' + j) << ": " << suma\_candidato;

                }

                // Calcular el distrito con más votos

                int mayor\_distrito = 0;

                int votos\_mayor\_distrito = 0;

                for (int i = 0; i < cantidad\_distrito; i++) {

                    int suma\_distrito = 0;

                    for (int j = 0; j < cantidad\_candidatos; j++) {

                        suma\_distrito += candidato[i][j];

                    }

                    if (suma\_distrito > votos\_mayor\_distrito) {

                        votos\_mayor\_distrito = suma\_distrito;

                        mayor\_distrito = i;

                    }

                    cout << "\nTotal votos Distrito " << i + 1 << ": " << suma\_distrito;

                }

                cout << "\n\nGanador: Candidato " << char('A' + mayor\_candidato)

                     << " con " << votos\_mayor\_candidato << " votos." << endl;

                cout << "Distrito con más participación: Distrito " << mayor\_distrito + 1

                     << " con " << votos\_mayor\_distrito << " votos." << endl;

                system("pause");

                break;

            }

            case 4:

                break;

            default:

                cout << "Opcion no valida!" << endl;

                system("pause");

        }

    } while (opcion != 4);

    // Liberar memoria

    for (int i = 0; i < cantidad\_distrito; i++) {

        delete[] candidato[i];

    }

    delete[] candidato;

    return 0;

}