

UNIVERSIDAD MARIANO GALVEZ DE GUATEMALA, SOLOLA

Ingeniería En Sistemas

Tercer Semestre

Curso: Programación 1

Catedrático: Ing. Neftalí García Lopez

Estudiante: Pascual Evelio Yaxon Tziquin

No. Carné: 2290-24-26037



Tema

TAREA SEMANA 6

CODIGOS COMENTADOS:

- Matriz con el ciclo for

```
using namespace std; // Sirve para evitar escribir std:: antes de cada función

int main() { // Inicio de la función principal del programa
    int filas = 3, columnas = 4; // Declarar e inicializar las variables para el número de filas y columnas de la matriz

    vector<vector<int>> matriz(filas, vector<int>(columnas)); // Crear una matriz de tamaño filas x columnas usando vectores.

    for (int i = 0; i < filas; i++) { // Bucle para recorrer cada fila de la matriz.
        for (int j = 0; j < columnas; j++) { // Bucle para recorrer cada columna de la fila actual.
            matriz[i][j] = (i + 1) * (j + 1); // Asignar a cada elemento de la matriz el producto de (i+1) y (j+1).
        }
    }

    for (const auto& fila : matriz) { // Bucle para recorrer cada fila de la matriz.
        for (int valor : fila) { // Bucle para recorrer cada valor de la fila actual.
            cout << valor << "\t"; // Imprimir el valor seguido de una tabulación.
        }
        cout << endl; // Imprimir un salto de línea al final de cada fila.
    }

    return 0; // Indicar que el programa terminó correctamente
}
```

- Ciclo for each con arrays

```
#include <iostream> //Incluye la biblioteca estándar de entrada/salida para usar cout y endl
#include <vector> //Incluye la biblioteca de vectores para usar la estructura vector

using namespace std; //Usa el espacio de nombres estándar para evitar escribir std:: antes de cout, endl, etc.

int main() { //Función principal del programa
    vector<int> numeros = { 1, 2, 3, 4, 5 }; //Crea un vector de enteros con los valores 1, 2, 3, 4, 5

    cout << "Elementos del vector: "; //Imprime un mensaje indicando que se mostrarán los elementos del vector
    for (const auto& num : numeros) { //Recorre cada elemento del vector (num es una referencia constante a cada elemento)
        cout << num << " "; //Imprime el valor actual de num seguido de un espacio
    }
    cout << endl; //Imprime un salto de línea después de mostrar todos los elementos

    for (auto& num : numeros) { //Recorre cada elemento del vector (num es una referencia a cada elemento)
        num += 1; //Incrementa el valor de num en 1
    }

    cout << "Elements modificados: "; //Imprime un mensaje indicando que se mostrarán los elementos modificados
    for (const auto& num : numeros) { //Recorre cada elemento del vector nuevamente (num es una referencia constante)
        cout << num << " "; //Imprime el valor actual de num seguido de un espacio
    }
    cout << endl; // Imprime un salto de línea después de mostrar todos los elementos modificados

    return 0; // Indica que el programa terminó correctamente
}
```

- Búsqueda binaria

```
#include <iostream> // Incluye la biblioteca estándar de entrada/salida para usar cout, cin, etc.
#include <algorithm> // Incluye la biblioteca de algoritmos para usar funciones como sort
using namespace std; // Usa el espacio de nombres estándar para evitar escribir std:: antes de cout, cin, etc.

int main() { // Función principal del programa
    int arreglo[] = { 1, 5, 8, 2, 3, 9, 4 }; // Declara un arreglo de enteros con valores iniciales
    int clave; // Variable para almacenar el número que el usuario desea buscar
    bool encontrado = false; // Variable booleana para indicar si se encontró el número
    int indice = -1; // Variable para almacenar el índice del número encontrado (inicializado en -1)

    int tamano = sizeof(arreglo) / sizeof(arreglo[0]); // Calcula el tamaño del arreglo
    sort(arreglo, arreglo + tamano); // Ordena el arreglo en orden ascendente usando la función sort

    cout << "Array ordenado: "; // Imprime un mensaje indicando que se mostrará el arreglo ordenado
    for (int n : arreglo) { // Recorre cada elemento del arreglo (n es el valor actual)
        cout << n << " "; // Imprime el valor actual seguido de un guion
    }
    cout << endl; // Imprime un salto de línea después de mostrar el arreglo ordenado

    cout << "Ingresa un número a buscar: "; // Solicita al usuario que ingrese un número para buscar
    cin >> clave; // Almacena el número ingresado por el usuario en la variable clave

    int izquierda = 0; // Inicializa el índice izquierdo para la búsqueda binaria
    int derecha = tamano - 1; // Inicializa el índice derecho para la búsqueda binaria

    while (izquierda <= derecha) { // Bucle para realizar la búsqueda binaria
        int medio = izquierda + (derecha - izquierda) / 2; // Calcula el índice medio (evita desbordamiento)

        if (arreglo[medio] == clave) { // Si el valor en el índice medio es igual a la clave
            encontrado = true; // Marca como encontrado
            indice = medio; // Almacena el índice donde se encontró el número
            break; // Sale del bucle
        }
        else if (arreglo[medio] < clave) { // Si el valor en el índice medio es menor que la clave
            izquierda = medio + 1; // Busca en la mitad derecha del arreglo
        }
        else { // Si el valor en el índice medio es mayor que la clave
            derecha = medio - 1; // Busca en la mitad izquierda del arreglo
        }
    }

    if (encontrado) { // Si se encontró el número
        cout << "Número encontrado en la posición: " << indice << endl; // Imprime la posición del número
    }
    else { // Si no se encontró el número
        cout << "Número no encontrado" << endl; // Imprime un mensaje indicando que no se encontró
    }

    return 0; // Indica que el programa terminó correctamente
}
```

40 No se encontraron problemas. 0 cambios | 0 autores, 0 cambios | Línea: 30 Carácter: 93 SPC CRLF

- Búsqueda array

```
IANA 6 COMETAR CÓDIGO (Ambito global)
#include <iostream> // Incluye la biblioteca estándar de entrada/salida
using namespace std; // Usa el espacio de nombres estándar para evitar escribir std:: antes de cout, cin, etc

int main() { // Función principal del programa
    int arreglo[] = { 1, 5, 8, 2, 3, 9, 4 }; // Declara un arreglo de enteros con valores iniciales
    int clave; // Variable para almacenar el número que el usuario desea buscar
    bool encontrado = false; // Variable booleana para indicar si se encontró el número

    cout << "Ingresa un número a buscar: "; // Solicita al usuario que ingrese un número para buscar
    cin >> clave; // Almacena el número ingresado por el usuario en la variable clave

    int tamano = sizeof(arreglo) / sizeof(arreglo[0]); // Calcula el tamaño del arreglo

    for (int i = 0; i < tamano; i++) { // Recorre cada elemento del arreglo usando un índice i
        if (arreglo[i] == clave) { // Si el valor en la posición i es igual a la clave
            cout << "Número encontrado en la posición: " << i << endl; // Imprime la posición del número
            encontrado = true; // Marca como encontrado
            break; // Sale del bucle
        }
    }

    if (!encontrado) { // Si no se encontró el número después de recorrer el arreglo
        cout << "Número no encontrado en el array" << endl; // Imprime un mensaje indicando que no se encontró
    }

    return 0; // Indica que el programa terminó correctamente
}
```

- Ordenamiento burbuja

```
#include <iostream> // Incluye la biblioteca estándar de entrada/salida para usar cout, endl, etc.
using namespace std; // Usa el espacio de nombres estándar para evitar escribir std:: antes de cout, endl, etc.

// Función para ordenar un arreglo usando el algoritmo de Bubble Sort
void bubbleSort(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) { // Bucle externo para recorrer el arreglo
        for (int j = 0; j < n - i - 1; j++) { // Bucle interno para comparar elementos adyacentes
            if (arr[j] > arr[j + 1]) { // Si el elemento actual es mayor que el siguiente
                swap(arr[j], arr[j + 1]); // Intercambia los elementos
            }
        }
    }
}

// Función para imprimir un arreglo
void printArray(int arr[], int n) {
    for (int i = 0; i < n; i++) { // Recorre cada elemento del arreglo
        cout << arr[i] << " "; // Imprime el valor actual seguido de un espacio
    }
    cout << endl; // Imprime un salto de línea después de mostrar todos los elementos
}

// Función principal del programa
int main() {
    int arr[] = { 5, 2, 9, 1, 5, 6 }; // Declara un arreglo de enteros con valores iniciales
    int n = sizeof(arr) / sizeof(arr[0]); // Calcula el tamaño del arreglo

    bubbleSort(arr, n); // Llama a la función bubbleSort para ordenar el arreglo
    printArray(arr, n); // Llama a la función printArray para imprimir el arreglo ordenado

    return 0; // Indica que el programa terminó correctamente
}
```

Ordenamiento por selección

```
#include <iostream> // Incluye la biblioteca estándar de entrada/salida para usar cout, endl, etc.
using namespace std; // Usa el espacio de nombres estándar para evitar escribir std:: antes de cout, endl, etc.

// Función para ordenar un arreglo usando el algoritmo de Selection Sort
void selectionSort(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) { // Bucle externo para recorrer el arreglo
        int minIdx = i; // Supone que el elemento actual es el mínimo
        for (int j = i + 1; j < n; j++) { // Bucle interno para encontrar el mínimo en el resto del arreglo
            if (arr[j] < arr[minIdx]) { // Si encuentra un elemento menor que el mínimo actual
                minIdx = j; // Actualiza el índice del mínimo
            }
        }
        swap(arr[i], arr[minIdx]); // Intercambia el elemento actual con el mínimo encontrado
    }
}

// Función principal del programa
int main() {
    int arr[] = { 64, 25, 12, 22, 11 }; // Declara un arreglo de enteros con valores iniciales
    int n = sizeof(arr) / sizeof(arr[0]); // Calcula el tamaño del arreglo

    selectionSort(arr, n); // Llama a la función selectionSort para ordenar el arreglo

    for (int i = 0; i < n; i++) { // Recorre cada elemento del arreglo
        cout << arr[i] << " "; // Imprime el valor actual seguido de un espacio
    }

    return 0; // Indica que el programa terminó correctamente
}
```

LINK DE REPOSITORIO EN GITHUB:

<https://github.com/E-lev10/C-digos-comentados/upload>