

Universidad Mariano Gálvez de Guatemala

Facultad de Ingeniería en Sistemas de la Información y la Computación

Campus Villa Nueva, Guatemala

Ingeniería en Sistemas -5090

Curso: DERECHO INFORMÁTICO

Licenciado/Ingeniero titular: Ing. Carlos Arias

Lab #10

ESTUDIANTES: Carlos Eduardo García Cortez

CARNET: 5090-24-14824

FECHA: 21/04/2025

Link del repositorio

https://github.com/Xplod883/Lab10_Arreglos_CarlosGarcia.git

Serie 1: Arreglos

En las siguientes capturas de pantalla se muestra el código ejecutado:

```
-índice0arreglo1  
-índice1arreglo2  
-índice2arreglo3  
-índice3arreglo4  
-índice4arreglo5  
-índice5arreglo6  
-índice6arreglo7  
-índice7arreglo8  
-índice8arreglo9  
-índice9arreglo10
```

Ejecución normal

```
#include <iostream>  
  
int main() {  
    int arreglo[10];  
  
    for (int i = 0; i < 10; i++) {  
        arreglo[i] = i + 1;  
    }  
  
    for (int i = 0; i < 10; i++) {  
        std::cout << "Índice" << i << "arreglo" << arreglo[i] << std::endl;  
    }  
  
    return 0;  
}
```

Consola de depuración

```
-índice0arreglo1  
-índice1arreglo2  
-índice2arreglo3  
-índice3arreglo4  
-índice4arreglo5  
-índice5arreglo6  
-índice6arreglo7  
-índice7arreglo8  
-índice8arreglo9  
-índice9arreglo10
```

Con Break Point y watch.

Parte 2: Estadísticas con arreglos

En las siguientes capturas de pantalla se muestra el código ejecutado:

```

Ingrese 10 numeros enteros:
Numero 1.4
Numero 2.2
Numero 3.6
Numero 4.2
Numero 5.5
Numero 6.5
Numero 7.2
Numero 8.6
Numero 9.7
Numero 10.8

Resultados:
Suma total: 47
Promedio: 4.7
Valor minimo: -2147483648
Valor maximo: 2
Cantidad de pares: 7
Cantidad de impares: 3

C:\Users\alumno\source\repos\Lab_10_Actividad2\x64\Debug\Lab_10_Actividad2.exe (proceso 18572) se cerró con el código 0 (0x0).
Presione cualquier tecla para cerrar esta ventana. . .|

```

Normal

```

Ingrese 10 numeros enteros:
Numero 1.3
Numero 2.5
Numero 3.21
Numero 4.5
Numero 5.5
Numero 6.32
Numero 7.2
Numero 8.5
Numero 9.5
Numero 10.67

Resultados:
Suma total: 150
Promedio: 15
Valor minimo: -2147483648
Valor maximo: 2
Cantidad de pares: 2
Cantidad de impares: 8

```

Usando break points en: `cin >> numeros[i], if (numeros[i] < maximo) {, y maximo = numeros[i];`

Parte 3: Arreglos en orden inverso

```

Arreglo original: 1020304050607080
Arreglo en orden inverso: 2020202020202020

C:\Users\alumno\source\repos\Lab_10_Actividad2\x64\Debug\Lab_10_Actividad2.exe (proceso 4748) se cerró con el código 0 (0x0).
Presione cualquier tecla para cerrar esta ventana. . .|

```

Normal

```
Arreglo original: 1020304050607080
Arreglo en orden inverso: |
```

Usando breakpoint en `cout << *(arr + 1) << "\n";` y `imprimirImpreso(arreglo, TAM);`

No se ejecuta el orden inverso al tener break point en la linea `cout`.

La expresión `*(arr + i)` es fundamental en el manejo de punteros en C++ y funciona de la siguiente manera:

`arr` es un puntero: Cuando pasamos un arreglo a una función, realmente estamos pasando un puntero al primer elemento del arreglo.

Aritmética de punteros: `(arr + i)` realiza aritmética de punteros:

- `arr` apunta a la dirección de memoria del primer elemento (índice 0)
- `arr + i` calcula la dirección de memoria del elemento en la posición `i`
- Cada suma de `i` avanza en memoria el tamaño de un `int` (4 bytes típicamente)
- Operador de desreferencia `*`: El operador `*` obtiene el valor almacenado en la dirección de memoria calculada.

En nuestro caso particular dentro de `imprimirInverso`:

- Cuando `i = tam - 1` (7 en nuestro ejemplo), `*(arr + 7)` accede al último elemento (80)
- Cuando `i = 0`, `*(arr + 0)` accede al primer elemento (10).

Parte 4: Filtro por promedio

Escribe un programa que:

1. Ingrese 15 números enteros.
2. Calcule el promedio de los números.
3. Cree un nuevo arreglo que contenga solo los valores mayores al promedio.
4. Imprima ambos arreglos.

Extra (si el tiempo lo permite):

- Ordene el nuevo arreglo de mayor a menor.
- Utilice funciones auxiliares.

```
Ingresa 15 numeros enteros:
Numero 1: 1
Numero 2: 2
Numero 3: 3
Numero 4: 4
Numero 5: 5
Numero 6: 6
Numero 7: 7
Numero 8: 8
Numero 9: 9
Numero 10: 10
Numero 11: 11
Numero 12: 12
Numero 13: 13
Numero 14: 141
Numero 15: 15

El promedio es: 16.4667
Arreglo original:
1 2 3 4 5 6 7 8 9 10 11 12 13 141 15

Valores mayores al promedio:
141

Valores ordenados descendente:
141
```

Programa del código

Cada tarea está encapsulada en su propia función, y cada arreglo de mayores se crea con el tamaño exacto necesario, y se Incluyen funciones para:

- Ingreso de datos
- Cálculo de promedio
- Filtrado de valores
- Ordenamiento descendente
- Impresión de arreglos.