

Universidad Mariano Gálvez De Guatemala

Facultad de ingeniería, Matemática y ciencias Físicas

Campus Villa Nueva Guatemala

Ingeniería en Sistemas de Información y ciencias de la computación

Curso: programación 1

Código de curso: 012

Código de la carrera: 5090

Estudiante: Coosemans Spiegeler Max Eduardo

Sección: A

Numero de carné: 5090-23-3282

Fecha: 21/02/2024



## Introducción:

En la programación, es común encontrarnos con la necesidad de realizar operaciones básicas sobre números enteros, como sumarlos, calcular factoriales, contar dígitos, ordenarlos, o sumar los dígitos individuales de un número. En este ejercicio, abordaremos cada una de estas operaciones utilizando diferentes estructuras de control como bucles for, while y do-while.

## Características de las funciones implementadas

1. **Suma de números pares e impares:** La función recibe un número entero como parámetro y devuelve la suma de todos los números pares o impares hasta ese número, según una condición dada. Se utiliza un bucle for para iterar sobre los números y realizar la suma.
2. **Factorial con bucle while:** Esta función calcula el factorial de un número dado utilizando un bucle while. El factorial de un número entero positivo  $n$  es el producto de todos los enteros positivos menores o iguales a  $n$ .
3. **Contador de dígitos:** La función cuenta la cantidad de dígitos que tiene un número entero dado. Se utiliza un bucle do-while para garantizar que el número se procese al menos una vez, y se utiliza una división entera para extraer los dígitos.
4. **Ordenamiento de números (ascendente):** Esta función ordena tres números enteros de menor a mayor utilizando condicionales y métodos de intercambio. No se utilizan arrays, sino que se comparan los números y se intercambian sus posiciones si es necesario. Se pueden utilizar bucles for, while o do-while según la preferencia.
5. **Suma de dígitos de un número:** La función suma todos los dígitos de un número entero dado utilizando bucles for, while o do-while para iterar sobre los dígitos del número. Se extraen los dígitos mediante operaciones de módulo y división entera.

```
1. #include <iostream>
```

```
int sumaParesImpares(int num, bool sumarPares) {  
    int suma = 0;  
    for (int i = 1; i <= num; i++) {  
        if ((i % 2 == 0 && sumarPares) || (i % 2 != 0 && !sumarPares)) {  
            suma += i;  
        }  
    }  
    return suma;  
}
```

```
int main() {  
    int numero;  
    char opcion;
```

```
    std::cout << "Ingrese un numero entero: ";  
    std::cin >> numero;
```

```
    std::cout << "¿Desea sumar pares (P) o impares (I)? ";  
    std::cin >> opcion;
```

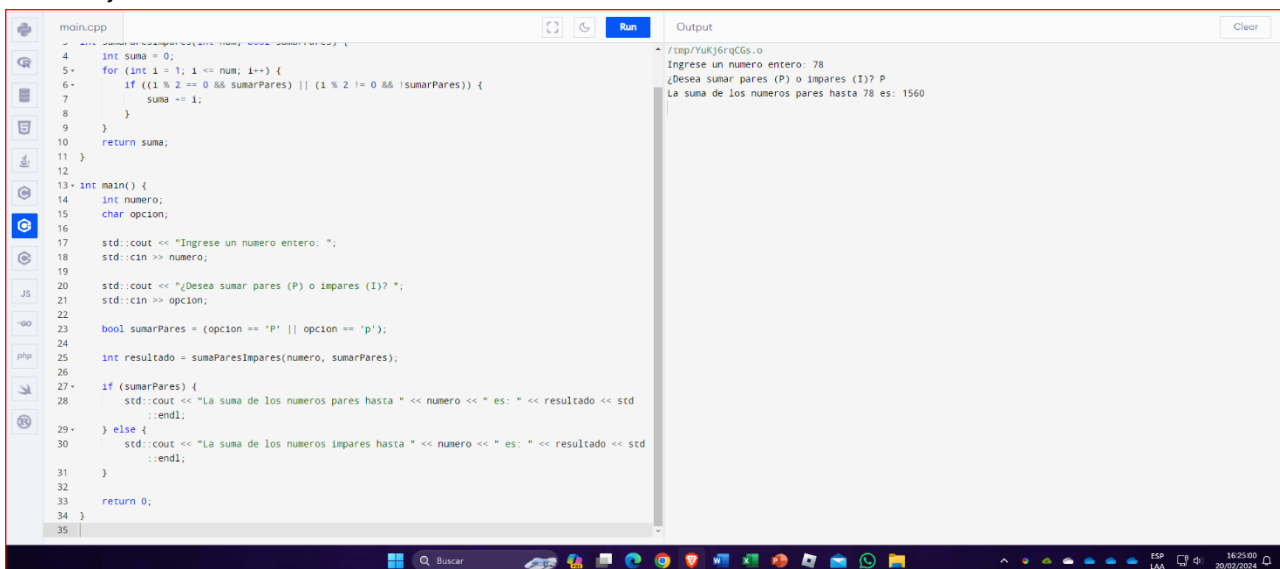
```
    bool sumarPares = (opcion == 'P' || opcion == 'p');
```

```
    int resultado = sumaParesImpares(numero, sumarPares);
```

```
    if (sumarPares) {  
        std::cout << "La suma de los numeros pares hasta " << numero << " es: " << resultado <<  
std::endl;  
    } else {  
        std::cout << "La suma de los numeros impares hasta " << numero << " es: " << resultado  
<< std::endl;  
    }  
}
```

```
    return 0;
```

```
}
```



The screenshot shows a C++ IDE with a file named 'main.cpp'. The code is the same as the one provided in the previous blocks. The 'Output' window on the right shows the program's execution. It prompts the user to 'Ingrese un numero entero: 78' and '¿Desea sumar pares (P) o impares (I)? P'. The output then displays 'La suma de los numeros pares hasta 78 es: 1560'. The IDE's taskbar at the bottom shows various application icons and the system clock indicating 16:25:00 on 20/02/2024.

2. **#include <iostream>**

**unsigned long long calcularFactorial(int n) {**

**unsigned long long factorial = 1;**

**int i = 1;**

**while (i <= n) {**

**factorial \*= i;**

**i++;**

**}**

**return factorial;**

**}**

**int main() {**

**int numero;**

**std::cout << "Ingrese un numero entero positivo: ";**

**std::cin >> numero;**

**if (numero < 0) {**

**std::cout << "El factorial no está definido para números negativos." <<**

**std::endl;**

**} else {**

**unsigned long long resultado = calcularFactorial(numero);**

**std::cout << "El factorial de " << numero << " es: " << resultado << std::endl;**

**}**

**return 0;**

**}**

The screenshot shows the Programiz C++ Online Compiler interface. On the left, the file 'main.cpp' is open, displaying the C++ code for calculating a factorial. The code includes the `<iostream>` header, defines a `calcularFactorial` function using a `while` loop, and implements a `main` function that prompts the user for a positive integer, checks for negative values, and prints the factorial result. On the right, the 'Output' panel shows the program's execution: it prompts 'Ingrese un numero entero positivo: 5' and outputs 'El factorial de 5 es: 120'. The bottom of the image shows a Windows taskbar with various application icons and a system clock indicating 16:27:26 on 20/02/2024.

```
3. #include <iostream>
```

```
#include <cmath>
```

```
int contarDigitos(int numero) {  
    // Manejo de números negativos convirtiéndolos a positivos  
    if (numero < 0) {  
        numero = -numero;  
    }  
}
```

```
int contador = 0;
```

```
// Bucle do-while para contar los dígitos
```

```
do {  
    numero /= 10;  
    contador++;  
} while (numero != 0);
```

```
return contador;
```

```
}
```

```
int main() {  
    int numero;
```

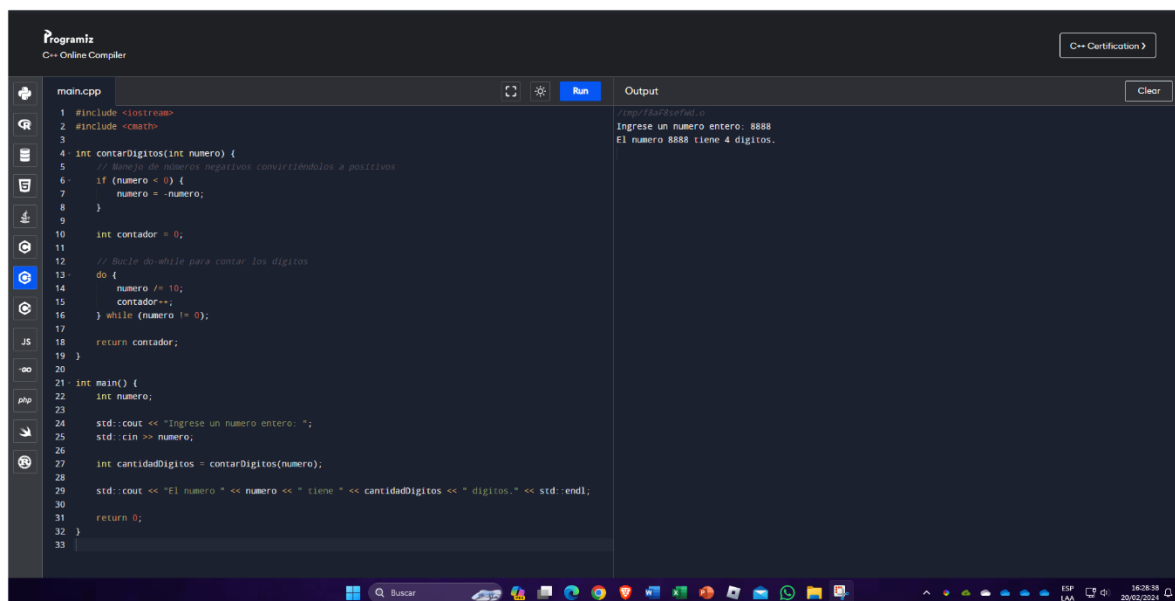
```
    std::cout << "Ingrese un numero entero: ";  
    std::cin >> numero;
```

```
    int cantidadDigitos = contarDigitos(numero);
```

```
    std::cout << "El numero " << numero << " tiene " << cantidadDigitos << " digitos." <<  
    std::endl;
```

```
    return 0;
```

```
}
```



The screenshot shows the Programiz C++ Online Compiler interface. The code editor on the left contains the C++ code for counting digits, and the output window on the right shows the result of running the program with the input 8888.

```
main.cpp  
1 #include <iostream>  
2 #include <cmath>  
3  
4 int contarDigitos(int numero) {  
5     // Manejo de números negativos convirtiéndolos a positivos  
6     if (numero < 0) {  
7         numero = -numero;  
8     }  
9  
10    int contador = 0;  
11  
12    // Bucle do-while para contar los dígitos  
13    do {  
14        numero /= 10;  
15        contador++;  
16    } while (numero != 0);  
17  
18    return contador;  
19 }  
20  
21 int main() {  
22     int numero;  
23  
24     std::cout << "Ingrese un numero entero: ";  
25     std::cin >> numero;  
26  
27     int cantidadDigitos = contarDigitos(numero);  
28  
29     std::cout << "El numero " << numero << " tiene " << cantidadDigitos << " digitos." <<  
30     std::endl;  
31     return 0;  
32 }  
33
```

Output

```
Ingrese un numero entero: 8888  
El numero 8888 tiene 4 digitos.
```

4. #include <iostream>

```
void ordenarNumeros(int& num1, int& num2, int& num3) {  
    if (num1 > num2) {  
        int temp = num1;  
        num1 = num2;  
        num2 = temp;  
    }  
    if (num2 > num3) {  
        int temp = num2;  
        num2 = num3;  
        num3 = temp;  
    }  
    if (num1 > num2) {  
        int temp = num1;  
        num1 = num2;  
        num2 = temp;  
    }  
}
```

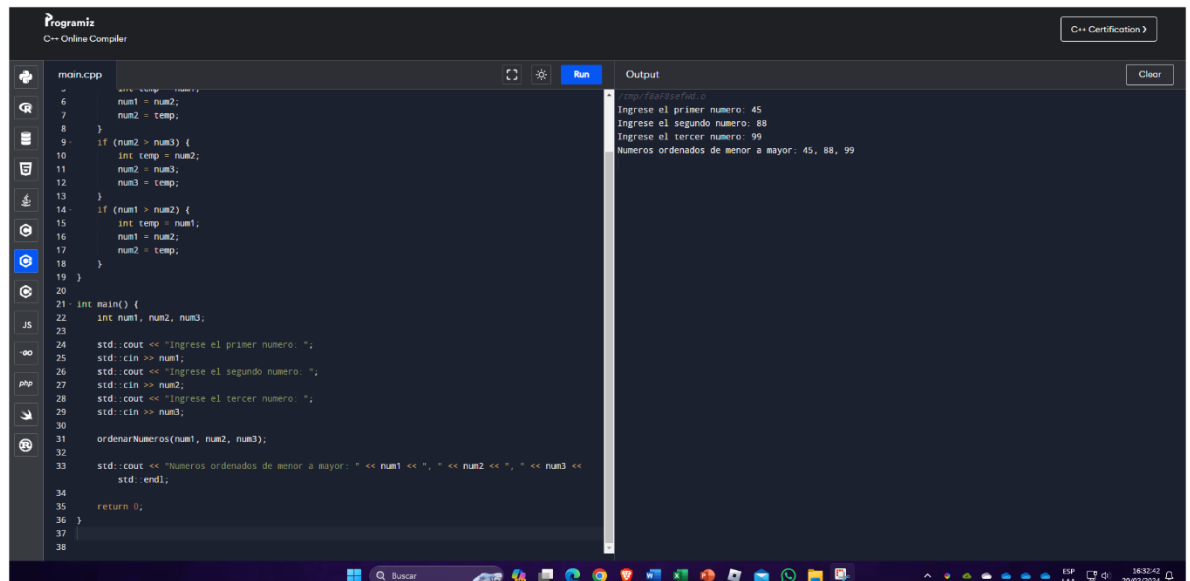
```
int main() {  
    int num1, num2, num3;
```

```
    std::cout << "Ingrese el primer numero: ";  
    std::cin >> num1;  
    std::cout << "Ingrese el segundo numero: ";  
    std::cin >> num2;  
    std::cout << "Ingrese el tercer numero: ";  
    std::cin >> num3;
```

```
    ordenarNumeros(num1, num2, num3);
```

```
    std::cout << "Numeros ordenados de menor a mayor: " << num1 << ", " << num2 << ", " <<  
num3 << std::endl;
```

```
    return 0;  
}
```



The screenshot shows the Programiz C++ Online Compiler interface. The code editor on the left contains the C++ program for sorting three numbers. The output window on the right shows the program's execution results for the input values 45, 88, and 99.

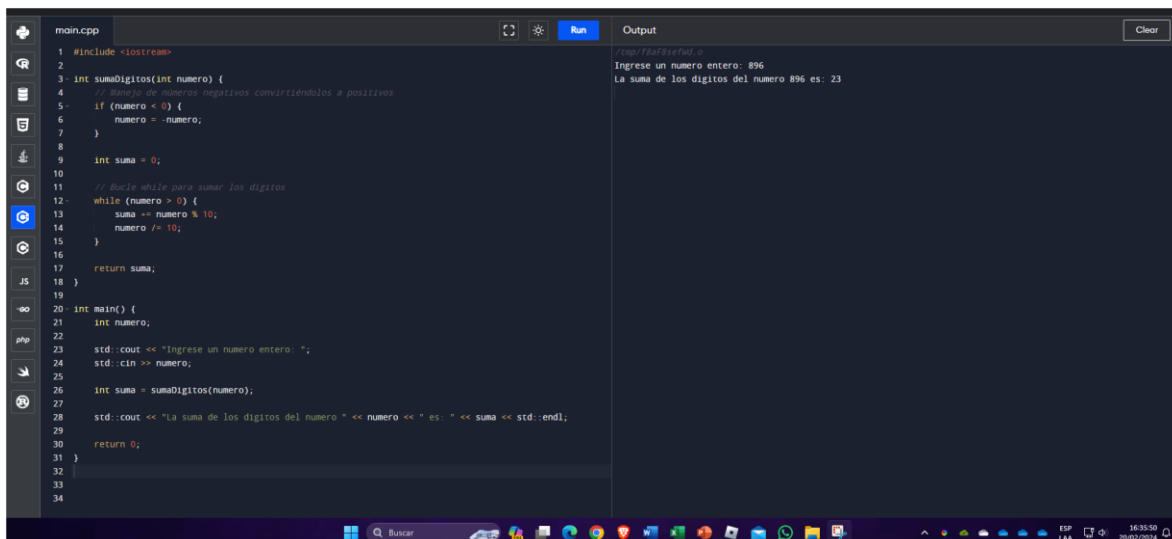
```
main.cpp  
1  #include <iostream>  
2  
3  void ordenarNumeros(int& num1, int& num2, int& num3) {  
4      if (num1 > num2) {  
5          int temp = num1;  
6          num1 = num2;  
7          num2 = temp;  
8      }  
9      if (num2 > num3) {  
10         int temp = num2;  
11         num2 = num3;  
12         num3 = temp;  
13     }  
14     if (num1 > num2) {  
15         int temp = num1;  
16         num1 = num2;  
17         num2 = temp;  
18     }  
19 }  
20  
21 int main() {  
22     int num1, num2, num3;  
23  
24     std::cout << "Ingrese el primer numero: ";  
25     std::cin >> num1;  
26     std::cout << "Ingrese el segundo numero: ";  
27     std::cin >> num2;  
28     std::cout << "Ingrese el tercer numero: ";  
29     std::cin >> num3;  
30  
31     ordenarNumeros(num1, num2, num3);  
32  
33     std::cout << "Numeros ordenados de menor a mayor: " << num1 << ", " << num2 << ", " << num3 <<  
34     std::endl;  
35     return 0;  
36 }  
37  
38
```

Output

```
Ingrese el primer numero: 45  
Ingrese el segundo numero: 88  
Ingrese el tercer numero: 99  
Numeros ordenados de menor a mayor: 45, 88, 99
```

5. #include <iostream>

```
int sumaDigitos(int numero) {  
    // Manejo de números negativos convirtiéndolos a positivos  
    if (numero < 0) {  
        numero = -numero;  
    }  
  
    int suma = 0;  
  
    // Bucle while para sumar los dígitos  
    while (numero > 0) {  
        suma += numero % 10;  
        numero /= 10;  
    }  
  
    return suma;  
}  
  
int main() {  
    int numero;  
  
    std::cout << "Ingrese un numero entero: ";  
    std::cin >> numero;  
  
    int suma = sumaDigitos(numero);  
  
    std::cout << "La suma de los digitos del numero " << numero << " es: " << suma << std::endl;  
  
    return 0;  
}
```



The screenshot shows a C++ IDE with a dark theme. The left sidebar contains icons for file explorer, search, and other IDE features. The main editor area displays the C++ code from the previous block, with line numbers 1 through 34. The code is for a program that calculates the sum of the digits of an integer. The right sidebar shows the 'Output' window, which contains the program's execution results: 'Ingrese un numero entero: 896' and 'La suma de los digitos del numero 896 es: 23'. The Windows taskbar is visible at the bottom of the screen.

```
main.cpp  
1 #include <iostream>  
2  
3 int sumaDigitos(int numero) {  
4     // Manejo de números negativos convirtiéndolos a positivos  
5     if (numero < 0) {  
6         numero = -numero;  
7     }  
8  
9     int suma = 0;  
10  
11     // Bucle while para sumar los dígitos  
12     while (numero > 0) {  
13         suma += numero % 10;  
14         numero /= 10;  
15     }  
16  
17     return suma;  
18 }  
19  
20 int main() {  
21     int numero;  
22  
23     std::cout << "Ingrese un numero entero: ";  
24     std::cin >> numero;  
25  
26     int suma = sumaDigitos(numero);  
27  
28     std::cout << "La suma de los digitos del numero " << numero << " es: " << suma << std::endl;  
29  
30     return 0;  
31 }  
32  
33  
34
```

Output  
Clear  
//main.cpp:20:3  
Ingrese un numero entero: 896  
La suma de los digitos del numero 896 es: 23



## conclusión

Estas funciones demuestran diferentes técnicas para manipular números enteros en programación, utilizando diversas estructuras de control como bucles for, while y do-while. Cada función resuelve un problema específico de manera eficiente y utilizando las características propias del lenguaje de programación. Es importante comprender y dominar estos conceptos básicos para poder abordar problemas más complejos en el desarrollo de software.

[Enlace a GitHub](#)