

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

Ingeniero: Carlos Alejandro Arias

Curso: Física 1

Código de Curso: 012

Código de Carrera: 5090



Laboratorio 5

Byron Ignacio Salazar Orellana

Sección: A

No. Carné: 23-5096

Fecha: 21/02/2024

Introducción

En el presente documento se nos presentan una serie de programas, los cuales ponen a prueba los conocimientos adquiridos en la clase de Programación 1, como estudiantes es necesario invertir tiempo en una práctica adecuada para el desarrollo del conocimiento suficiente para programar en lenguaje C de una forma fluida, eficiente y también contar con la capacidad de detectar errores o posibles alternativas que puede tener cada uno de los programas.

```

1 #include <iostream>
2 // Para este programa necesitamos que nos de el resultado del factorial del numero ingresado
3 // para ello ingresamos como entero la variable factorial, y la variable numero el cual se debe ingresar
4 int factorial(int numero) {
5     int resultado = 1;
6     // tenemos el if, el cual compara el numero para saber si es igual que 0 y luego lo retorna
7     if (numero == 0) {
8         return 1;
9     }
10    // en esta parte, el while va a comparar el número con 0, de ser falso, se le resta 1.
11    while (numero > 0) {
12        resultado *= numero;
13        numero--;

```

Para este código se nos solicita obtener el factorial del número ingresado por el usuario, es por ello que el programa evalúa la resta del número mismo por 1 hasta llegar a 0, y esto se va multiplicando.

```

    // se retorna el resultado de la resta del número con 1.
    return resultado;
}
//en el main tenemos la ejecución del programa.
int main() {
    int numero;
    // se le solicita al usuario que ingrese el número entero.
    std::cout << "Ingrese un numero entero: ";
    std::cin >> numero;
    // si numero es menor que cero, es decir el número es negativo, se marca como error.
    if (numero < 0) {
        std::cout << "El factorial no está definido para números negativos." << std::endl;
        return 1;
    }
    // el resultado es igual al producto factorial del número ingresado.
    int resultado = factorial(numero);

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

    return 0;
}

```

Luego se muestra el resultado de la multiplicación de cada una de las restas del mismo número con 1, hasta que este sea 0, ese es el factorial del número ingresado.

```
D:\Factorial.exe
Ingrese un numero entero: 5
El factorial de 5 es: 120

-----
Process exited after 3.166 seconds with return value 0
Presione una tecla para continuar . . . |
```

Para el segundo programa el cual es para contar los dígitos de un numero se asigna una variable la cual es contador, que será 0 y luego iremos sumándole.

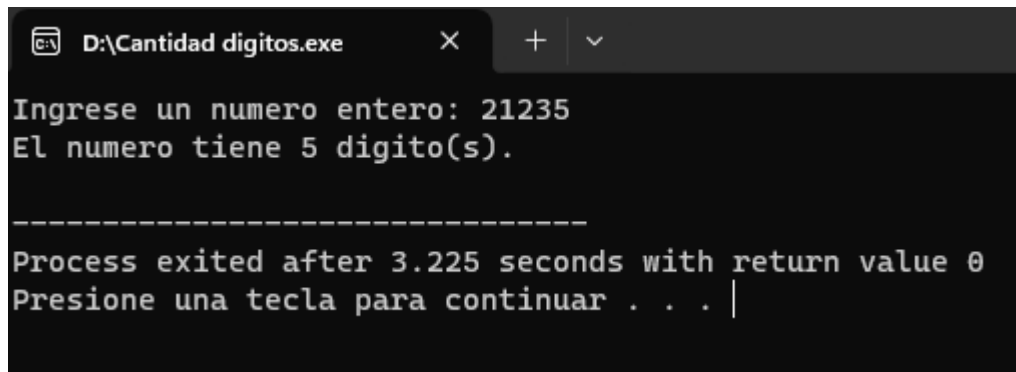
```
1  #include <iostream>
2  // Para este programa es necesario crear una variable de contador
3  int contarDigitos(int numero) {
4      int contador = 0;
5      // para el do debemos hacer una division y asignacion del numero con 10, luego ir sumando con ++
6      do {
7          numero /= 10;
8          contador++;
9      } while (numero != 0);
10 // se evalua si el numero es diferente a 0
11     return contador;
12 }
```

El numero debe ser distinto a 0, se va sumando cada uno de los dígitos hasta que sean 0 los dígitos restantes.

```
} int main() {
    int numero;
    // se le solicita al usuario ingresar un numero
    std::cout << "Ingrese un numero entero: ";
    std::cin >> numero;
    // esta variable es la que nos va ayudar a contar los digitos de un numero
    int cantidadDigitos = contarDigitos(numero);
    // se da el resultado al final para que el usuario lo pueda apreciar
    std::cout << "El numero tiene " << cantidadDigitos << " digito(s)." << std::endl;

    return 0;
}
```

Luego se le solicita a un usuario ingresar un numero entero, se le asigna el valor a la variable número y por consiguiente se muestra la suma de los dígitos obtenida.



```
D:\Cantidad digitos.exe X + v
Ingrese un numero entero: 21235
El numero tiene 5 digito(s).

-----
Process exited after 3.225 seconds with return value 0
Presione una tecla para continuar . . . |
```

Para el tercer programa se nos pide ordenar los números de menor a mayor, es por ello que utilizando una codificación de if pueden ser ordenados, también está la opción de hacerlo por arreglos.

```
1  #include <iostream>
2  // Para ordenar los numeros de mayor a menor se utilizamos un if
3  void ordenarNumeros(int &a, int &b, int &c) {
4  // por medio de esta cadena de if que evalúa todos los números entre si
5  // el numero mayor entre a y b
6      if (a > b) {
7          int temp = a;
8          a = b;
9          b = temp;
10     }
11     // el numero mayor entre a y c
12     if (a > c) {
13         int temp = a;
14         a = c;
15         c = temp;
16     }
17     // el numero mayor entre b y c
18     if (b > c) {
19         int temp = b;
20         b = c;
21         c = temp;
22     }
23 }
24 // en la funcion principal se ejecuta el programa anterior
```

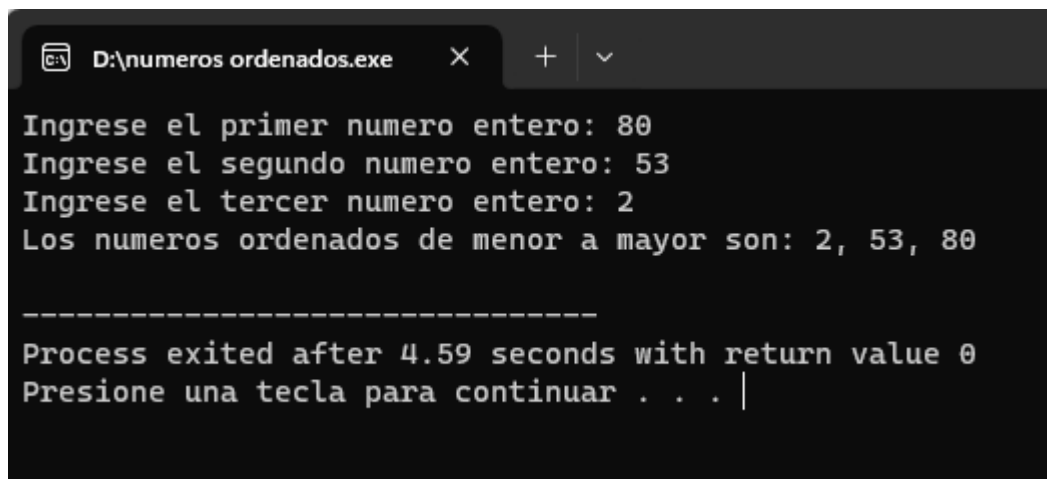
Esta función principal evalúa todos los números entre sí, para obtener un resultado con respecto al orden, y luego brindarle la información al usuario.

```

}
// en la funcion principal se ejecuta el programa anterior
int main() {
    int num1, num2, num3;
    // se solicita cada uno de los numeros y se les asigna un valor
    std::cout << "Ingrese el primer numero entero: ";
    std::cin >> num1;
    std::cout << "Ingrese el segundo numero entero: ";
    std::cin >> num2;
    std::cout << "Ingrese el tercer numero entero: ";
    std::cin >> num3;
    // luego se ordenan con la programacion del if anteriormente mencionada
    ordenarNumeros(num1, num2, num3);
    // se da el resultado al usuario.
    std::cout << "Los numeros ordenados de menor a mayor son: " << num1 << ", " << num2 << ", " << num3 << std::endl;

    return 0;
}
```

Se le solicita al usuario que ingrese los números enteros, luego de ser ordenados de menor a mayor, se muestra la permutación correspondiente. Por medio de un arreglo pueden compararse más números.



```
D:\numeros ordenados.exe  X  +  v

Ingrese el primer numero entero: 80
Ingrese el segundo numero entero: 53
Ingrese el tercer numero entero: 2
Los numeros ordenados de menor a mayor son: 2, 53, 80

-----
Process exited after 4.59 seconds with return value 0
Presione una tecla para continuar . . . |
```

Para el cuarto programa se nos pide sumar los dígitos de un número ingresado.

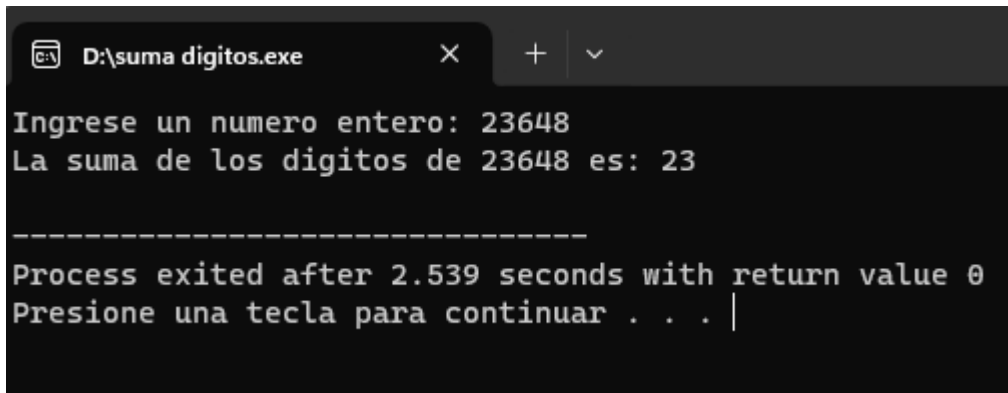
```
1  #include <iostream>
2  // para sumar los dígitos de un numero empezamos con la estructura de la funcion
3  int sumarDigitos(int numero) {
4      int suma = 0;
5      // acá utilizamos un while, donde el numero de digitos debe ser diferente a 0
6      while (numero != 0) {
7          suma += numero % 10;
8          numero /= 10;
9      }
10     // la los digitos se van sumando conforme la asignacion que hace el programa.
11     return suma;
12     // se retorna la suma
13 }
```

Suma es igual a 0, para luego asignarle la suma de dígitos. Luego de ello se extrae la cantidad de cada dígito, y se ejecuta la suma de los mismos.

```
int main() {
    int numero;
    // se le solicita al usuario que ingrese el numero
    std::cout << "Ingrese un numero entero: ";
    std::cin >> numero;
    // la suma es igual a la suma de los digitos del numero
    int suma = sumarDigitos(numero);
    // se le da al usuario el resultado
    std::cout << "La suma de los digitos de " << numero << " es: " << suma << std::endl;

    return 0;
}
```

Se asigna a una variable el número ingresado por el usuario, se muestra el resultado de la suma y el usuario puede observar la cantidad.



```
D:\suma digitos.exe
Ingrese un numero entero: 23648
La suma de los digitos de 23648 es: 23

-----
Process exited after 2.539 seconds with return value 0
Presione una tecla para continuar . . . |
```

Para el último programa se nos pide hacer la suma de números pares o impares hasta llegar a una cantidad ingresada.

```
1  #include <iostream>
2  // para la suma de los numeros pares e impares hasta cierto numero tenemos este programa
3  int sumaParesImpares(int numero, int esPar) {
4      // suma sera igual a para luego ir sumando los numeros
5      int suma = 0;
6      for (int i = 1; i <= numero; i++) {
7          // so es par quiere decir que es divisible entre 2, si es impar no es divisible entre 2
8          if ((esPar && i % 2 != 0) || (!esPar && i % 2 == 0)) {
9              suma += i;
10         }
11     }
12     // si el usuario ingresa se suman los pares, si ingresa 1 se suman impares
13     return suma;
14 }
```

Suma es igual a 0, si el usuario es ingresa “0” el se toman los números divisibles entre 2 ya que son los pares, si el usuario ingresa 1 se toman los números que no son divisibles entre 2, ya que son los impares. Luego estos se suman.


```

    }
    return suma;
}

int main() {
    int numero;
    int opcion;
    // se le solicita al usuario que ingrese el número y la opción del mismo
    std::cout << "Ingrese un numero entero: ";
    std::cin >> numero;
    std::cout << "Ingrese 0 para sumar pares o 1 para sumar impares: ";
    std::cin >> opcion;
    // si la opcion no es valida se le hará saber al usuario que ingrese 1 o 0
    if (opcion != 0 && opcion != 1) {
        std::cout << "Opción inválida. Por favor, ingrese 0 para sumar pares o 1 para sumar impares." << std::endl;
        return 1;
    }
    // Luego se suman los numeros ya sean pares o impares y se le da el resultado al usuario.
    int resultado = sumaParesImpares(numero, opcion);
    if (opcion == 0)
        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;
}

```

Se le solicita al usuario que ingrese el número que quisiera, para luego ejecutar el proceso correspondiente, en donde el resultado es la suma de los números pares o impares, según sea el caso.

```

D:\Numeros Pares e Impares. X + v
Ingrese un numero entero: 20
Ingrese 0 para sumar pares o 1 para sumar impares: 0
La suma de los numeros pares hasta 20 es: 110

-----
Process exited after 4.601 seconds with return value 0
Presione una tecla para continuar . . . |

```

```

D:\Numeros Pares e Impares. X + v
Ingrese un numero entero: 20
Ingrese 0 para sumar pares o 1 para sumar impares: 1
La suma de los numeros impares hasta 20 es: 100

-----
Process exited after 4.572 seconds with return value 0
Presione una tecla para continuar . . . |

```

Conclusión

Podemos decir que cada uno de estos programas impulsa nuestro desarrollo en la programación, es ideal practicar de una forma constante hasta poder identificar de una forma sencilla la ejecución de cada programa y entender qué utilizar cuando se nos presenta un problema en donde debemos programar una aplicación, no solo en lenguaje C sino también en cualquier otro lenguaje de programación.