

# Algoritmos y Estructuras de datos en C

## Conceptos para Proyecto 3

Pedro Villar

FAMAF

3 de octubre de 2023

# Contenido

- 1 Introducción
- 2 GCC en Linux
- 3 Sintaxis en C
- 4 Primer código en C
- 5 Conceptos - Proyecto 1
- 6 Ejercicios Resolución

# Introducción

- Cada bloque de código se encierra en corchetes. La base de cada bloque es la función principal `main()`. Un programa puede contener varias funciones, cada una con su juego de parámetros.
- Al final de cada operador se pone un punto y coma. Si se te olvidó ponerlo, el compilador te mostrará el mensaje de error “Semicolon is expected”.
- Los espacios se usan para separar partes de operador.

# GCC

Es como la caja de herramientas que nos ofrece el GNU kernel. gcc es un compile driver, se encarga de la compilación, es por eso que le puedo pedir una etapa en particular para que termine la compilación en ese punto. **Secuencia del GCC:**

1. Compilamos.
2. Traducimos a Assembler.
3. De Assembler vamos a lenguaje binario (0 y 1), lenguaje máquina.
4. Linkeo. El linker se encarga de buscar los símbolos necesarios en las librerías del lenguaje.

# GCC Comandos

- `gcc file.c`: crea el ejecutable, por defecto devuelve un archivo `a.out` .
- `gcc file.c -o myFile` para que se cree un archivo ejecutable llamado `myFile`.
- `gcc -E file.c`: solamente hace el preprocesamiento.
- `gcc -S file.c`: genera la salida en `Assembler`.

# Sintaxis Básica C - 1

1. **Inclusión de Librerías:** Para usar funciones y definiciones de una librería, se utiliza la directiva `#include` seguida del nombre de la librería entre ángulos (por ejemplo, `#include <stdio.h>`).
2. **Declaración de Variables:** Para declarar una variable, se utiliza el formato `tipo nombre_variable;` (por ejemplo, `int numero;`).
3. **Asignación de Valores:** Puedes asignar valores a una variable con el operador de asignación `=` (por ejemplo, `numero = 10;`).
4. **Tipos de Datos:** C ofrece tipos de datos como `int`, `char`, `float`, y `double` para almacenar diferentes tipos de información.

## Sintaxis Básica C - 2

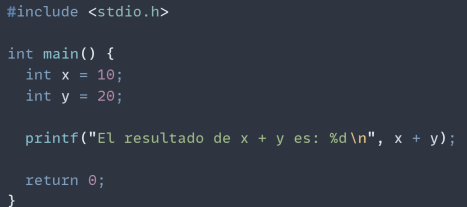
1. **Declaración de Funciones:** Para declarar una función, especificas su tipo de retorno, nombre y parámetros entre paréntesis (por ejemplo, `int suma(int a, int b);`).
2. **Definición de Funciones:** Para definir una función, debes proporcionar su implementación con un bloque de código entre llaves .
3. **Llamada de Funciones:** Para usar una función, simplemente llámala por su nombre, pasando los argumentos necesarios (por ejemplo, `resultado = suma(5, 3);`).

## Sintaxis Básica C - 3

1. **Funciones Estándar:** C incluye una variedad de funciones predefinidas, como `printf` para salida de texto, `scanf` para entrada de texto, `strlen` para obtener la longitud de una cadena, y muchas más.
2. **Librerías Estándar:** Puedes usar funciones y constantes de las librerías estándar, como `<stdio.h>` para entrada/salida, `<math.h>` para operaciones matemáticas, y `<string.h>` para manipulación de cadenas.



# Primer Código en C



```
#include <stdio.h>

int main() {
    int x = 10;
    int y = 20;

    printf("El resultado de x + y es: %d\n", x + y);

    return 0;
}
```

codesnap.dev

Figura: Código que suma dos números

# Printf

printf es una función que se utiliza para imprimir texto y valores en la consola o en un archivo en C.

- Incluir la Librería: incluir la librería estándar de entrada/salida (stdio.h) al principio de tu programa.
- Llamar a printf: la función printf se llama con el siguiente formato: **printf("Texto y formato", argumentos);**
  - "Texto y formato": Aquí proporcionas el texto que deseas imprimir. Puedes incluir texto literal (como palabras o frases) y códigos de formato que indican cómo se imprimirán los valores.
  - argumentos: Puedes proporcionar valores que deseas imprimir. Estos valores se insertarán en el texto utilizando los códigos de formato.

# Códigos de Formato

Los códigos de formato son marcadores que indican dónde se insertarán los valores en el texto. Algunos ejemplos comunes son:

- **%d**: Para imprimir un valor entero.
- **%f**: Para imprimir un valor de punto flotante (números decimales).
- **%s**: Para imprimir una cadena de texto.
- **%c**: Para imprimir un caracter.

# Scanf

scanf es una función que se utiliza para obtener entrada de usuario desde la consola y almacenarla en variables en un programa en C.

- Incluir la Librería: incluir la librería estándar de entrada/salida (stdio.h) al principio de tu programa.
- Llamar a scanf: la función scanf en tu programa. La función scanf se llama con el siguiente formato: **scanf("Formato de entrada", &variable);**
  - **Formato de entrada:** Debes especificar los mismos códigos de formato que usarás para leer los datos.
  - **&variable:** Debes proporcionar la dirección de la variable donde deseas almacenar la entrada del usuario. El operador & se utiliza para obtener la dirección de una variable.

## Código de ejemplo con printf y scanf

```
#include <stdio.h>

int main() {
    int edad;
    printf("Por favor, introduce tu edad: ");
    scanf("%d", &edad);

    printf("Tienes %d años.\n", edad);

    return 0;
}
```

codesnap.dev

**Figura:** Explicación del código en la siguiente filmina

## Explicación del código

En este ejemplo, hemos declarado una variable `edad` para almacenar la edad del usuario. Usamos `printf` para mostrar un mensaje solicitando la entrada del usuario, y luego utilizamos `scanf` con el código de formato `%d` para leer un valor entero que el usuario ingrese. El valor se almacena en la variable `edad`.

# Ejercicio 1

1. **Entrada/Salida** Hace un programa en C, que solicite el ingreso de los valores de las variables  $x, y, z$ , e imprima el resultado de las siguientes expresiones. Completá los resultados de la tablas para los dos estados dados.

Expresión	$(x \rightarrow 7, y \rightarrow 3, z \rightarrow 5)$	$(x \rightarrow 1, y \rightarrow 10, z \rightarrow 8)$
$x + y + 1$		
$z * z + y * 45 - 15 * x$		
$y - 2 == (x * 3 + 1) \% 5$		
$y / 2 * x$		
$y < x * z$		

Figura: Ejercicio 1 - Proyecto 3

# Explicación

1. **Solicitud de ingreso de datos:** para solicitar el ingreso de datos, se utilizará printf para mostrar en pantalla un mensaje de aviso luego con scanf se almacenarán los valores en una determinada variable previamente declarada.
2. **Imprimir el resultado de las expresiones:** para imprimir, se utiliza el printf con las determinadas expresiones, el problema se separa en 5 salidas para su fácil interpretación. Cada printf es una expresión de las del enunciado.



# Resolución

```
#include <stdio.h>

int main() {
    int x;
    int y;
    int z;

    printf("Ingrese el número x: ");
    scanf("%d", &x);
    printf("Ingrese el número y: ");
    scanf("%d", &y);
    printf("Ingrese el número z: ");
    scanf("%d", &z);

    printf("El resultado de  $x + y + 1$  es: %d\n", x + y + 1);
    printf("El resultado de  $z * z + y * 45 - 15 * x$  es: %d\n", z * z + y * 45 - 15 * x);
    printf("El resultado de  $y - 2 = (x + 3 + 1) \bmod 5$  es: %d\n", y - 2 == (x + 3 + 1) % 5);
    printf("El resultado de  $y / 2 * x$  es: %d\n", y / 2 * x);
    printf("El resultado de  $y < x * z$  es: %d\n", y < x * z);
    return 0;
}
```

codesnap.dev

Figura: Ejercicio 1 - Proyecto 3