Lenguaje C nace en la década del 70. Lenguaje viejo que se sigue utilizando con mucha menor frecuencia que antes, pero es base de la mayoría de los lenguajes modernos. Es un lenguaje potente que resuelve problemas a bajo nivel con mayor eficacia que nuevos lenguajes.

Es un lenguaje de bajo nivel (cercano al hardware). Padre de los lenguajes javascript, c++, c#, entre otros.

Función Main 🡪 entry point o punto de entrada

C distingue entre comillas dobles a comillas simples, tienen usos diferentes. Las comillas dobles sirven para strings mientras que las comillas simples sirven para caracteres individuales.

Es un lenguaje interpretado a diferencia de JS que es compilado. Esto significa que el programa no va a correr hasta que se encuentre en optimas condiciones, no perdona faltas de “;” y otras cuestiones. No compilara hasta que se encuentre bien escrito.

Compilar significa traducir a lenguaje de maquina la información en el programa.

Se comenta con los símbolos // o /\* \*/.

Let o var ya no se utilizan para declarar variables, ya que hay que declarar QUE tipo de variable estamos declarando (carácter, string, numero entero, numero flotante, etcétera).

Tipos de datos:  
enteros: int 🡪 1, 999, -74 -> 4 bytes

Flotantes: float 🡪 3.14, -82.24, 5/8 -> 4 bytes

Caracteres: char 🡪 ‘j’, ‘.’, ‘^’, ‘1’ -> 1 byte

Dobles: double 🡪 más números despues de la coma (precisión) que un flotante/float. -> 8 bytes

Vacio: void -> 1 byte

Printf() sirve para mostrar en pantalla (imprimir). Para mostrar una variable desde printf se debe citar la mascara y posterior al string declarar la variable. Si son mas de 1 variable las que hay que mostrar, se deben colocar en orden de las mascaras, siempre separándolas por una “,”.

Scanf() sirve para escanear lo ingresado. “Scanf(tipo, direccionDeMemoria);”. La clausula de escape del scanf es el enter. El scanf necesita un tipo de dato declarado por la mascara y la dirección de memoria de la variable que va a llenar con el dato ingresado.

\n sirve para hacer un salto de línea.

Tipos de mascara:

Int %d

Float %f

Char %c

& 🡪 muestra la dirección de memoria de una variable

Char unCaracter;

Int unNumero;

Float otroNumero;

Printf(“Ingrese un numero: ”);

Scanf(“%c”, &unCaracter);

Printf(“Ingrese un numero: ”);

Scanf(“%d”, &unNumero);

Printf(“Ingrese un numero con coma: ”);

Scanf(“%f”, &otroNumero);

Printf(“El numero ingresado es: %d\n El numero con coma ingresado es: %f\nEl carácter es: %c”, unNumero, otroNumero, unCaracter);

<https://onlinegdb.com/Qc3NWBuJl>

ejercicio1-2: <https://onlinegdb.com/P2vA_Zn6wh>