
 GOBIERNO DEL ESTADO DE MÉXICO		MANUAL DE PRÁCTICAS FO-TESJI-11100-12		 TECNOLÓGICO DE ESTUDIOS SUPERIORES <u>JILOTEPEC</u>	
Nombre de la práctica	LENGUAJE C Y VARIABLES C			No.	1
Asignatura:	METODOS NUMERICOS	Carrera:	ING. SISTEMAS COMPUTACIONALES	Duración de la práctica (Hrs)	10

Nombre del alumno: Lissette Garcia Nolasco

I. Competencia(s) específica(s):

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):

Aula

III. Material empleado:

Dev C++

Computadora

IV. Desarrollo de la práctica:

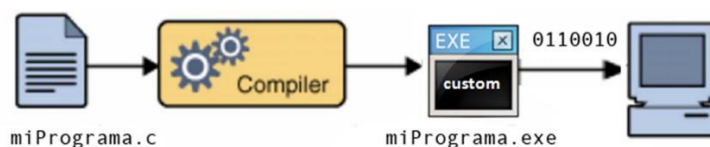
¿Qué es C?

- C es un lenguaje de programación estructurada desarrollado a principios de los años 70.
- Es un lenguaje de nivel medio, pero con muchas características de bajo nivel.
- Está disponible en un amplio abanico de plataformas

Características

- Lenguaje de programación de propósito general, muy adecuado para la programación de sistemas (UNIX fue escrito en C.)
- Lenguaje relativamente pequeño: solo ofrece sentencias de control sencillas y funciones.
- La E/S no forma parte del lenguaje, si no que se proporciona a través de una biblioteca de funciones.
- Permite la agrupación de instrucciones. Programación estructurada.
- Permite la separación de un programa en módulos que admiten compilación independiente. Diseño modular

¿Cómo funciona C?



OPERADORES ARITMÉTICOS

+	[adición]
-	[sustracción]
*	[multiplicación]
/	[división]
%	[módulo]
++	[incremento]
--	[decremento]

PALABRAS RESERVADAS

```
auto break case char const continue default do
double else enum extern float for goto if int
long register return short signed sizeof static
struct typedef union unsigned void volatile
while
```

TIPOS DE DATOS

Tipo	Descripción (bits)	Rango
short int	16	-32,768 to 32,767
unsigned short int	16	0 to 65,535
int	32	-2,147,483,648 to 2,147,483,647
unsigned int	32	0 to 4,294,967,295
long int	32	-2,147,483,648 to 2,147,483,647
unsigned long int	32	0 to 4,294,967,295
long long int	64	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
unsigned long long int	64	0 to 18,446,744,073,709,551,615

Tipo	Descripción (bits)	Rango
signed char	8	-128 a 127
unsigned char	8	0 a 255
float	8	1e-37 a 1e+37
double	16	-32768 a 32767
long double	32	3.4E-38 a 3.4 E+38

EJERCICIO:

Crear un archivo en C

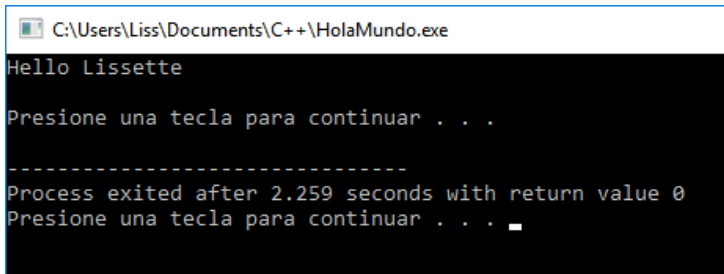
-Abrimos Dev-C++

-Escogemos el menú Archivo>Nuevo>Archivo fuente

Se escribe el siguiente texto en el editor

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main () {
5     printf("Hello Lissette \n \n");
6     system("Pause");
7     return 0;
8 }
```

Clic en botón ejecutar y compilar



```
C:\Users\Liss\Documents\C++\HolaMundo.exe
Hello Lissette
Presione una tecla para continuar . . .
-----
Process exited after 2.259 seconds with return value 0
Presione una tecla para continuar . . .
```

ESTRUCTURAS DE DECISION

Estructuras de decisión simple:

Ejecuta una determinada acción cuando se cumple una cierta condición y en caso contrario sigue el orden secuencial.

Si es verdadero, ejecuta una o varias opciones Si es falsa, entonces no hace nada y sigue la ejecución normal del programa.

Ejemplo 1.- Decisión simple

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int numero=34;
6     if(numero%2==0){
7         printf("El numero es par \n");
8     }
9     printf("Fin del programa\n");
10    system("Pause");
11    return 0;
12 }
13 }
```

C:\Users\Dotae_nbywxbb\Documents\Projects_c\ejemplo_manual_if_uno.exe

El numero es par
Fin del programa
Press any key to continue . . .

Ejemplo 2.- Decisión simple con condición falsa

Al no cumplirse la condición el programa evita esas líneas de código y se pasa a lo que encuentre después de ellas.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int numero=55;
6     if(numero%2==0){
7         printf("El numero es par \n");
8     }
9     printf("Fin del programa\n");
10    system("Pause");
11    return 0;
12 }
13 }
```

C:\Users\Dotae_nbywxbb\Documents\Projects_c\ejemplo_manual_if_uno.exe

Fin del programa
Press any key to continue . . .

Ejercicio 1.- Curso

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int cal;
6     puts("Ingrese la calificacion");
7     scanf("%d",&cal);
8
9     if(cal>=8){
10         puts("Felicidades aprobaste el curso!");
11     }
12
13     system("Pause");
14     return 0;
15 }
16 }
```

C:\Users\Dotae_nbywxbb\Documents\Projects_c\viernes1.exe

Ingrese la calificacion
8
Felicidades aprobaste el curso!
Press any key to continue . . .

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int cal;
6     puts("Ingrese la calificacion");
7     scanf("%d",&cal);
8
9     if(cal>=8){
10         puts("Felicidades aprobaste el curso!");
11     }
12
13     system("Pause");
14     return 0;
15 }
16
```

C:\Users\Dotae_nbywxbb\Documents\Projects_c\viernes1.exe

Ingrese la calificacion
7
Press any key to continue . . .

Estructuras de decisión simple anidada

Una estructura es anidada cuando existe una estructura dentro de otra.

El inicio y el final de la estructura anidada debe quedar totalmente dentro del inicio y el final de la estructura que permite dicho anidamiento.

Ejemplo 1.- Estructura de decisión simple anidada

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int numero=55;
6     if(numero<100){
7         printf("El numero es menor que 100 \n");
8         if(numero>50){
9             printf("y mayor que 50\n");
10        }
11    }
12
13    printf("Fin del programa\n");
14    system("Pause");
15    return 0;
16 }
17
18
```

C:\Users\Dotae_nbywxbb\Documents\Projects_c\ejemplo_manual_if_dos.exe

El numero es menor que 100
y mayor que 50
Fin del programa
Press any key to continue . . .

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int numero=35;
6     if(numero<100){
7         printf("El numero es menor que 100 \n");
8         if(numero>50){
9             printf("y mayor que 50\n");
10        }
11    }
12    printf("Fin del programa\n");
13    system("Pause");
14    return 0;
15 }
```

C:\Users\Dotae_nbywxbb\Documents\Projects_c\ejemplo_manual_if_dos.exe

El numero es menor que 100
Fin del programa
Press any key to continue . . .

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int numero=135;
6     if(numero<100){
7         printf("El numero es menor que 100 \n");
8         if(numero>50){
9             printf("y mayor que 50\n");
10        }
11    }
12    printf("Fin del programa\n");
13    system("Pause");
14    return 0;
15 }
```

C:\Users\Dotae_nbywxbb\Documents\Projects_c\ejemplo_manual_if_dos.exe

Fin del programa
Press any key to continue . . .

Ejercicio 1.- Crédito hipotecario

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int sueldo,suelUno,anti;
6     puts("Ingrese su antigüedad");
7     scanf("%d",&anti);
8     puts("Ingrese su sueldo");
9     scanf("%d",&sueldo);
10    suelUno=((sueldo*10)/100);
11
12    if(anti>=5){
13        if(suelUno>1000){
14            puts("Es sujeto a un credito hipotecario");
15        }
16    }
17    system("Pause");
18    return 0;
19 }
```

ESTRUCTURA DE DECISIÓN DOBLE

Permiten ejecutar dos conjuntos de sentencias alternativos en función del resultado de la evaluación de una condición lógica.

Si el resultado es “cierto” se ejecuta un grupo de sentencias.

Si el resultado es “falso” se ejecuta otro grupo de sentencias.

Ejemplo 1:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int numero=36;
6     if(numero<50){
7         printf("El numero es menor que 50 \n");
8     }
9     else{
10        printf("El numero es mayor que 50 \n");
11    }
12    printf("Fin del programa\n");
13    system("Pause");
14    return 0;
15 }
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int numero=60;
6     if(numero<50){
7         printf("El numero es menor que 50 \n");
8     }
9     else{
10        printf("El numero es mayor que 50 \n");
11    }
12    printf("Fin del programa\n");
13    system("Pause");
14    return 0;
15 }
```

Ejercicio 1 “Salario”:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int hst,ph,sal;
6     puts("Ingrese sus horas trabajadas");
7     scanf("%d",&hst);
8     puts("Ingrese el precio por hora");
9     scanf("%d",&ph);
10
11     if(hst<=40){
12         sal=hst*ph;
13         printf("Salario semanal: %d\n",sal);
14     }else{
15         sal=(40*ph+(ph*1.5*(hst-40)));
16         printf("Salario semanal: %d\n",sal);
17     }
18
19     system("Pause");
20     return 0;
21 }
22 }
```

```
Ingrese sus horas trabajadas
40
Ingrese el precio por hora
50
Salario semanal: 2000
Press any key to continue . . .
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int hst,ph,sal;
6      puts("Ingrese sus horas trabajadas");
7      scanf("%d",&hst);
8      puts("Ingrese el precio por hora");
9      scanf("%d",&ph);
10
11     if(hst<=40){
12         sal=(hst*ph);
13         printf("Salario semanal: %d\n",sal);
14     }else{
15         sal=(40*ph+(ph*1.5*(hst-40)));
16         printf("Salario semanal: %d\n",sal);
17     }
18
19     system("Pause");
20     return 0;
21 }
22
```

```
Ingrese sus horas trabajadas
41
Ingrese el precio por hora
50
Salario semanal: 2075
Press any key to continue . . .
```

Operador condicional

Es el único operador ternario de C, es decir usa tres operandos.

Sintaxis: `exp1 ? exp2 : exp3`

Si la `exp1` evalúa a cierto, la expresión completa evalúa al valor de `exp2`.

Si la `exp1` evalúa a falso, la expresión completa evalúa al valor de `exp3`.


```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int descuento=10;
6      int descuentoCliente=20;
7      int cliente=1;
8      int z=0;
9
10     z=(cliente)?descuentoCliente:descuento;
11     printf("El descuento es de %d \n",z);
12
13     printf("Fin del programa\n");
14     system("Pause");
15     return 0;
16 }
```

```
El descuento es de 20
Fin del programa
Press any key to continue . . .
```