



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Ing. García Morales Karina.

Asignatura: Fundamentos de programación.

Grupo: 22

No de Práctica(s): 8

Integrante(s): Alvarado Pérez Norma Laura.

Semestre: 2018-2

Fecha de entrega: 24/ Abril / 2018

Observaciones:

CALIFICACIÓN: _____

Guía práctica de estudio 08:

Estructuras de selección

Objetivo: Elaborar programas en lenguaje C que incluyan las estructuras de selección if, if-else, switch y ternaria (o condicional) para la resolución de problemas básicos.

Desarrollo:

Una estructura de control de flujo en un lenguaje especifica el orden en que se realiza el procesamiento de datos, tenemos 3 estructuras de control de flujo (*secuencial, condicional, iterativa*), en esta práctica nos centraremos en la condicional o estructura de selección.

El lenguaje C posee 3 estructuras de selección:

if-else

```
if (condición) //SI condición es verdadera
{
    //Bloque A de instrucciones a realizar
}else{ //SI condición es falsa
    //Bloque B de instrucciones a realizar
}
```

switch: compara un selector que solo puede ser de tipo entero o un carácter contra cada uno de sus casos y ejecuta las instrucciones del caso con el que coincida

```
switch(selector) {
    case 1:
        //Instrucciones
        break;
    case 2:
        //Instrucciones
        break;
    .
    .
    .
    case n:
        //Instrucciones
        break;
    default:
        //Instrucciones
}
```

ternaria: Consta de 3 partes, una condición, una acción a realizar si se cumple y una acción a realizar si no se cumple.

The diagram shows the expression `par = (a % 2 == 0) ? true : false;`. Above the expression, three brackets with labels identify its parts: 'Condición' points to `(a % 2 == 0)`, 'EN CASO VERDADERO' points to `true`, and 'EN CASO FALSO' points to `false`. Below the expression, a yellow bracket labeled 'Sintaxis' spans the entire conditional expression `(a % 2 == 0) ? true : false;`. At the bottom, a blue comment states: `//Esta instrucción asigna el valor true ó false a la variable tipo boolean llamada "par"`.

```
par = (a % 2 == 0) ? true : false;
```

//Esta instrucción asigna el valor true ó false a la variable tipo boolean llamada "par"

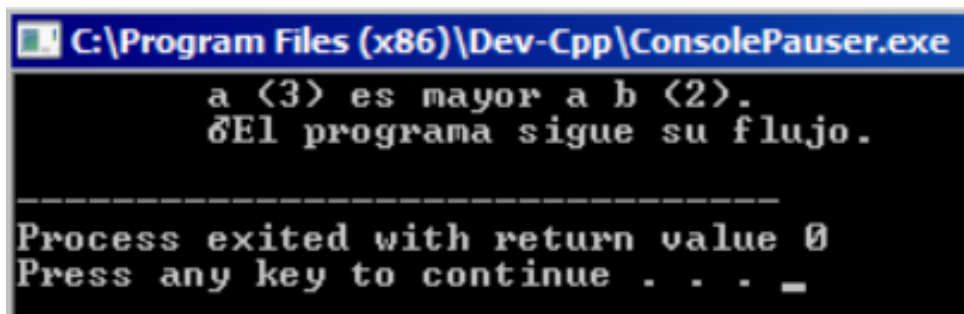
Ejercicios en el laboratorio

💜 **EJEMPLO 1:** En el siguiente ejemplo se visualiza la sentencia if-else, la condición es cierta.

```
#include<stdio.h>
/*
Este programa valida si el número a es mayor al número b.
*/
int main (){
    int a, b;
    a = 3;
    b = 2;

    if (a > b) {
printf("\ta (%d) es mayor a b (%d).\n",a,b);
    }
    else
        printf("\tb (%d) es mayor a la variable a (%d).\n");

printf("\t\vEl programa sigue su flujo.\n");
return 0;
}
```



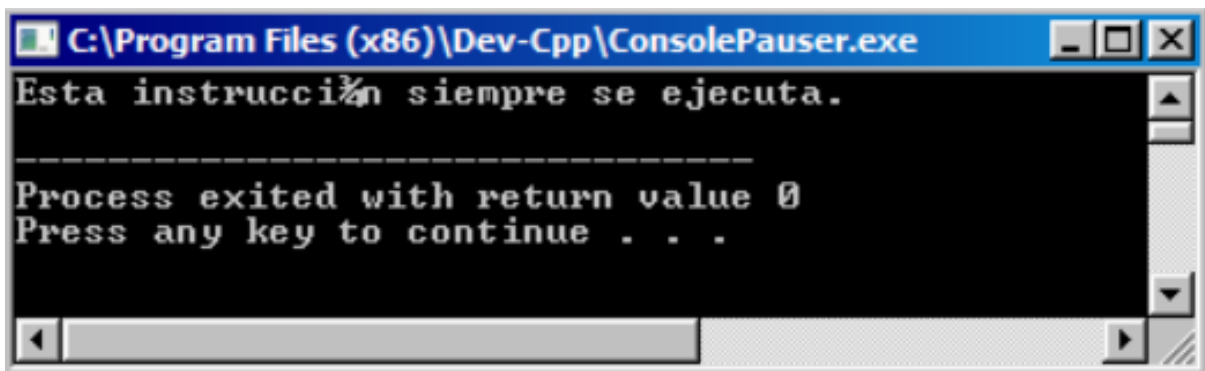
```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
a <3> es mayor a b <2>.
El programa sigue su flujo.
-----
Process exited with return value 0
Press any key to continue . . . _
```

💜 **EJEMPLO 2:** En este ejemplo vemos que el 0 es un valor considerado como FALSO y que los valores positivos o negativos son considerados como VERDADERO

```
#include<stdio.h>
/*
  Este programa comprueba que las condiciones son numéricas
  0 -> falso
  ? 0 -> Verdadero
*/
int main(){
  if (0){
    printf("Esta instrucción nunca se ejecuta\n");
    printf("porque la condición siempre es falsa (0).\n");
  }

  if (-38)
    // El bloque de código de esta estructura if
    // solo consta de una línea porque los comentarios
    // no son tomados en cuenta por el compilador.
    // La condición siempre es verdadera (diferente de 0)
    printf("Esta instrucción siempre se ejecuta.\n");

  return 0;
}
```



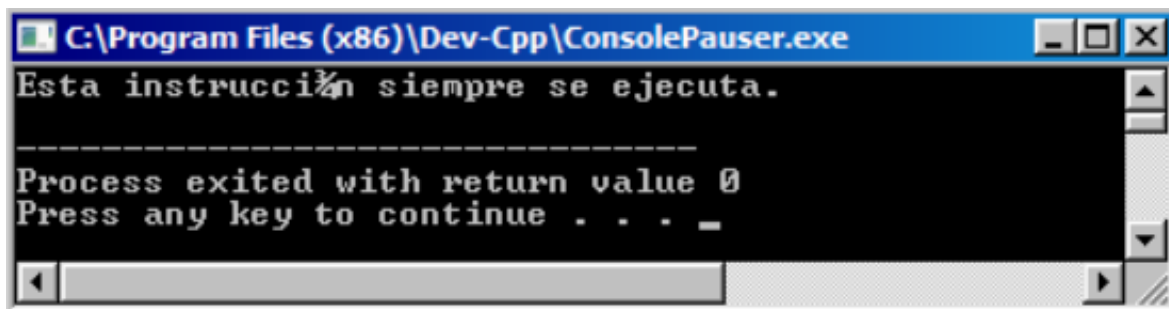
```

#include<stdio.h>
/*
    Este programa comprueba que las condiciones son numéricas
    0 -> falso
    ? 0 -> Verdadero
*/
int main(){
    if (0){
printf("Esta instrucción nunca se ejecuta\n");
printf("porque la condición siempre es falsa (0).\n");
    }

    if (1)
// El bloque de código de esta estructura if
// solo consta de una línea porque los comentarios
// no son tomados en cuenta por el compilador.
// La condición siempre es verdadera (diferente de 0)
printf("Esta instrucción siempre se ejecuta.\n");

    return 0;
}

```



```

C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Esta instrucción siempre se ejecuta.
-----
Process exited with return value 0
Press any key to continue . . . _

```

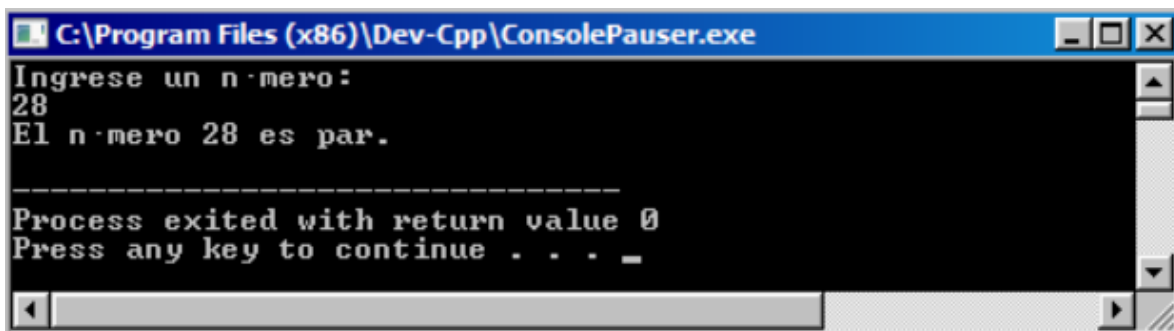
💜 **EJEMPLO 3:** Ejemplo de la sentencia if-else, se observa que cuando solo ejecutan una sentencia no es necesario poner {} llaves.

```
#include <stdio.h>
/*
    Este programa permite validar si un número es par o impar.
    El número se lee desde la entrada estándar (el teclado).
*/
int main() {
    int num;

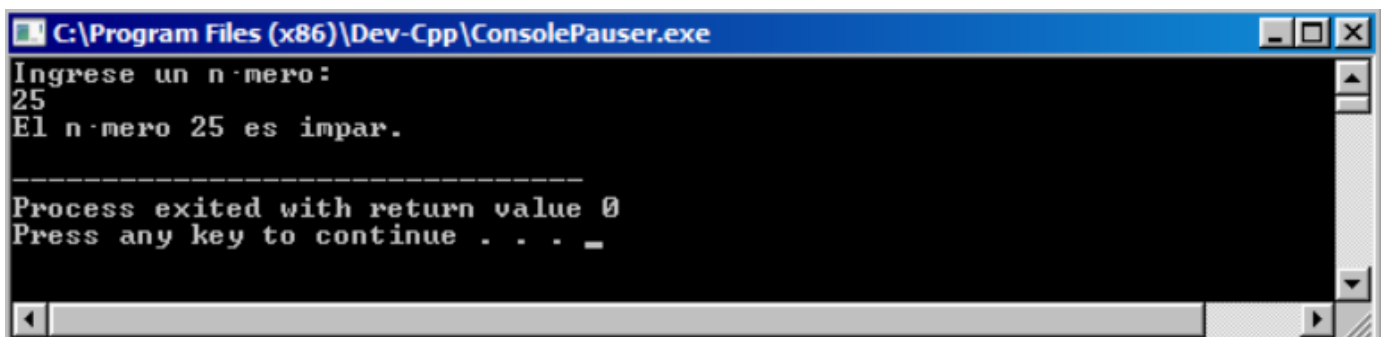
    printf("Ingrese un número:\n");
    scanf("%d",&num);

    if ( num%2 == 0 )
        printf("El número %d es par.\n",num);
    else
        printf("El número %d es impar.\n",num);

    return 0;
}
```



```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Ingrese un n-mero:
28
El n-mero 28 es par.
-----
Process exited with return value 0
Press any key to continue . . . _
```



```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Ingrese un n-mero:
25
El n-mero 25 es impar.
-----
Process exited with return value 0
Press any key to continue . . . _
```

```

#include <stdio.h>
/*Este programa ordena en forma descendente tres valores enteros dados.Los valores se leen desde la entrada estándar (el teclado).*/
int main(){
    int uno, dos, tres;

    printf ("Ingrese 3 números separados por espacios:\n");
    scanf ("%d %d %d", &uno, &dos, &tres);

    if (uno > dos){
        if (dos > tres){
            printf("%d es mayor a %d que es mayor a %d\n", uno, dos, tres);
        }else {
            if (uno > tres){
                printf("%d es mayor a %d que es mayor a %d\n", uno, tres, dos);
            }else {
                printf("%d es mayor a %d que es mayor a %d\n", tres, uno, dos);
            }
        }
    }
    else {
        if (dos > tres){
            if (tres > uno) {
                printf("%d es mayor a %d que es mayor a %d\n", dos, tres, uno);
            } else {
                printf("%d es mayor a %d que es mayor a %d\n", dos, uno, tres);
            }
        } else {
            printf("%d es mayor a %d que es mayor a %d\n", tres, dos, uno);
        }
    }
    return 0;
}

```

```

C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Ingrese 3 n-meros separados por espacios:
5 2 7
7 es mayor a 5 que es mayor a 2
-----
Process exited with return value 0
Press any key to continue . . .

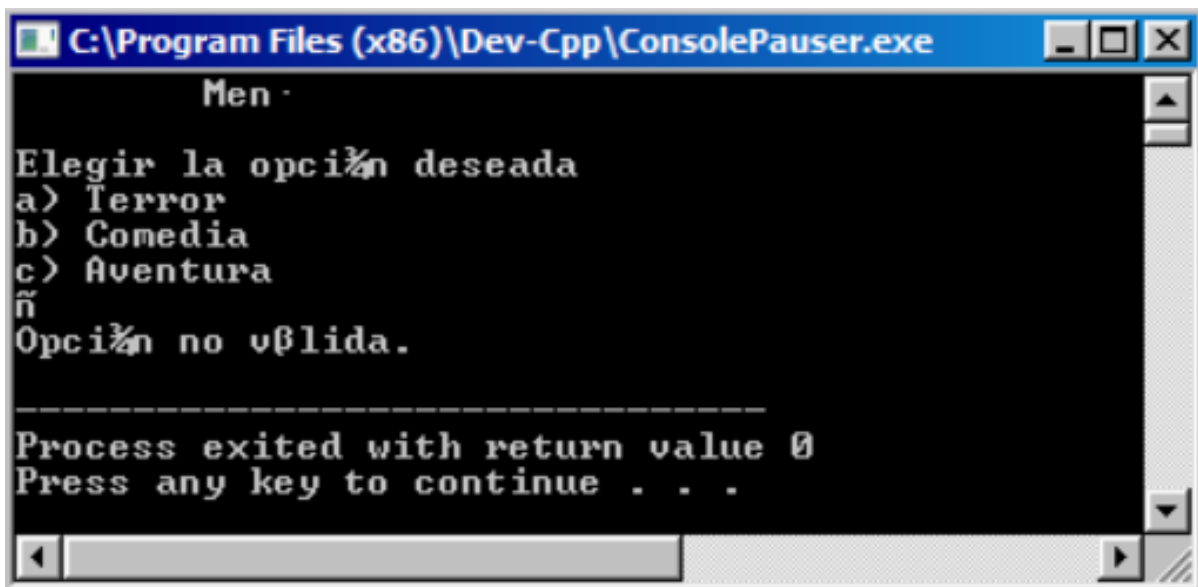
```

♥ EJEMPLO 5: Uso de la sentencia *switch* con un selector de tipo carácter.

```
#include <stdio.h>
#define p printf
int main() {
    char op = '\0';

    p("\tMenú\n\n");
    p("Elegir la opción deseada\n");
    p("a) Terror\n");
    p("b) Comedia\n");
    p("c) Aventura\n");
    scanf("%c", &op);

    switch(op) {
        default:
            p("Opción no válida.\n");
            break;
        case 'a':
            p("El resplandor.\nLa noche del demonio.\n");
            break;
        case 'b':
            p("Virgen a los 40.\nTodo poderoso.\n¿Y dónde estan las rubias?\n");
            break;
        case 'c':
            p("Rapido y furioso.\n");
            break;
    }
    return 0;
}
```




```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Men
Elegir la opción deseada
a) Terror
b) Comedia
c) Aventura
b
Virgen a los 40.
Todo poderoso.
¿Y dónde están las rubias?

-----
Process exited with return value 0
Press any key to continue . . .
```

Variables tipo enum

Es un tipo de dato constante, los valores que se ingresan son elementos enteros y constantes, por lo tanto, se escriben en mayúsculas, si no se le asignan valores, el primer elemento posee el valor de 0 y el segundo de 1 y así hasta concluir con los elementos, sin embargo, es posible cambiar el valor de un elemento asignándolo de forma directa, esto cambia la secuencia ya que los siguientes elementos continúan la numeración a partir del elemento anterior.

`enum identificador {VALOR1, VALOR2, ... , VALORN};`

0 1 n

`enum identificador {VALOR1, VALOR2=58, VALOR3, VALOR4, ... , VALORN};`

0 58 59 60 n

♥ EJEMPLO 6: Manejo de variable tipo *enum* (enumeración)

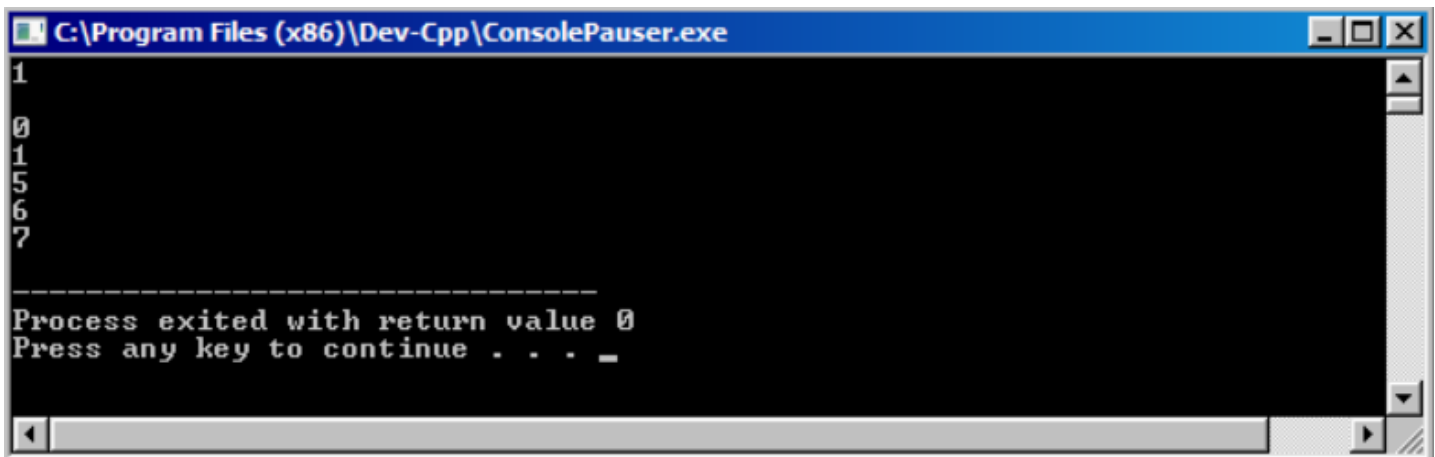
```
#include <stdio.h>
/*
    Este programa crea diversas variables tipo enum (enumerador) y
    permite visualizar la manera en la que se maneja el tipo de dato.
*/
int main(){
    // declaración de la enumeración
    enum boolean {NO, YES};

    // declaración de una variable tipo enumeración
    enum boolean valorBooleano;
    valorBooleano = YES;

    // Se comprueba que el valor de una enumeración es entero
    printf("%d\n", valorBooleano);

    // Se comprueba que el valor de una enumeración se puede reasignar
    enum diasSemana {LUNES, MARTES, MIERCOLES=5, JUEVES, VIERNES};
    printf("\n%d", LUNES);
    printf("\n%i", MARTES);
    printf("\n%d", MIERCOLES);
    printf("\n%i", JUEVES);
    printf("\n%d\n", VIERNES);

    return 0;
}
```



```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
1
0
1
5
6
7

-----
Process exited with return value 0
Press any key to continue . . . _
```

```

#include <stdio.h>
/*
Este programa crea diversas variables tipo enum (enumerador) y
permite visualizar la manera en la que se maneja el tipo de dato.
*/
int main(){
    // declaración de la enumeración
    enum boolean {NO, YES};

    // declaración de una variable tipo enumeración
    enum boolean valorBooleano;
    valorBooleano = NO;

    // Se comprueba que el valor de una enumeración es entero
    printf("%d\n", valorBooleano);

    // Se comprueba que el valor de una enumeración se puede reasignar
    enum diasSemana {LUNES, MARTES, MIERCOLES=8, JUEVES, VIERNES};
    printf("\n%d", LUNES);
    printf("\n%i", MARTES);
    printf("\n%d", MIERCOLES);
    printf("\n%i", JUEVES);
    printf("\n%d\n", VIERNES);

    return 0;
}

```

```

C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
0
0
1
8
9
10

-----
Process exited with return value 0
Press any key to continue . . .

```

♥ EJEMPLO 7: Manejo de variable tipo *enum* (enumeración) y *switch*

```
#include <stdio.h>

int main() {
    // Los valores de una enumeración son enteros y constantes
    enum diasSemana {LUNES, MARTES, MIERCOLES, JUEVES, VIERNES, SABADO, DOMINGO};
    int op;
    printf("Ingrese el día de la semana.\n");
    printf("1) Lunes\n");
    printf("2) Martes\n");
    printf("3) Miércoles\n");
    printf("4) Jueves\n");
    printf("5) Viernes\n");
    printf("6) Sábado\n");
    printf("7) Domingo\n");
    scanf("%d", &op);
    switch(op-1){
        case LUNES:
        case MARTES:
            printf("Inicio de semana.\n");
            break;
        case MIERCOLES:
            printf("Mitad de semana.\n");
            break;
        case JUEVES:
            printf(";Casi inicia el fin de semana!\n");
            break;
        case VIERNES:
        case SABADO:
            printf(";Fin de semana!\n");
            break;
        case DOMINGO:
            printf("Día de descanso.\n");
            break;
        // No se necesita default
    }
    return 0; // Valor entero en hexadecimal
}
```

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Ingrese el día de la semana.
1) Lunes
2) Martes
3) Miércoles
4) Jueves
5) Viernes
6) Sábado
7) Domingo
4
¡Casi inicia el fin de semana!

-----
Process exited with return value 0
Press any key to continue . . .
```

```
#include <stdio.h>
int main() {
    // Los valores de una enumeración son enteros y constantes
    enum diasSemana {LUNES, MARTES, MIERCOLES, JUEVES, VIERNES, SABADO, DOMINGO};
    int op;
    printf("Ingrese el día de la semana.\n");
    printf("1) Lunes\n");
    printf("2) Martes\n");
    printf("3) Miércoles\n");
    printf("4) Jueves\n");
    printf("5) Viernes\n");
    printf("6) Sábado\n");
    printf("7) Domingo\n");
    scanf("%d", &op);
    switch(op) {
        case LUNES:
        case MARTES:
            printf("Inicio de semana.\n");
            break;
        case MIERCOLES:
            printf("Mitad de semana.\n");
            break;
        case JUEVES:
            printf("¡Casi inicia el fin de semana!\n");
            break;
        case VIERNES:
        case SABADO:
            printf("¡Fin de semana!\n");
            break;
        case DOMINGO:
            printf("Día de descanso.\n");
            break;
        // No se necesita default
    }
    return 0; // Valor entero en hexadecimal
}
```

se cambió el valor a evaluar,
nunca va a tomar el caso 0 y
los casos irán desfasados ya
que se ingresará directamente
el valor del menú visualizado

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Ingrese el día de la semana.
1> Lunes
2> Martes
3> Miércoles
4> Jueves
5> Viernes
6> Sábado
7> Domingo
3
¿Casi inicia el fin de semana?

-----
Process exited with return value 0
Press any key to continue . . .
```

♥ EJEMPLO 7: Manejo de condicional ternaria

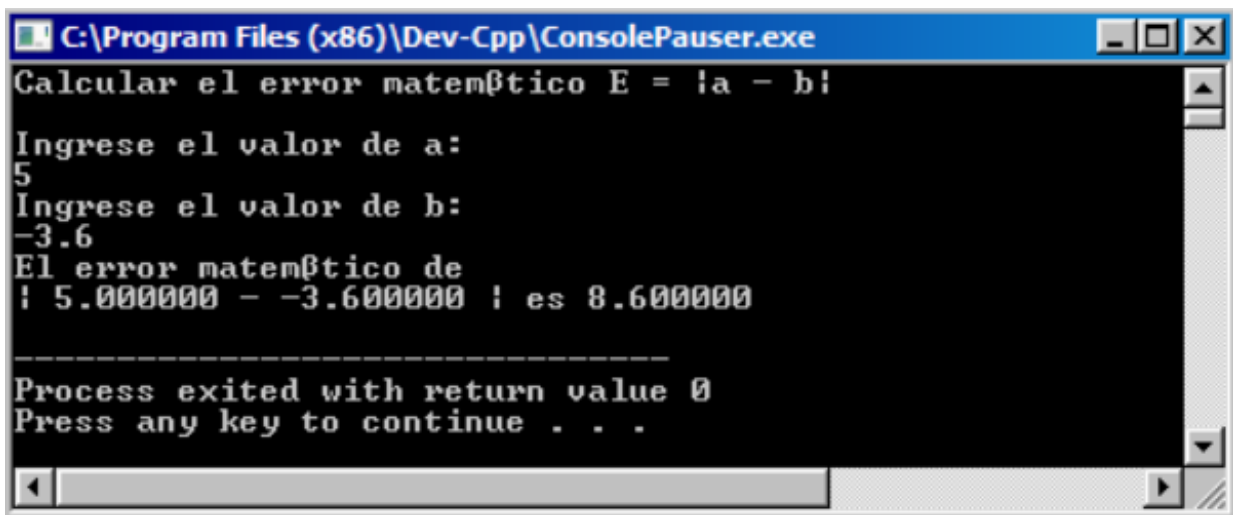
```
#include <stdio.h>
/*
    Este programa permite calcular el error matemático a partir de dos
    valores (a y b) ingresados desde la entrada estándar (el teclado), a partir
    de la fórmula:
     $E = |a - b|$ 
    Donde a es el valor real y b es el valor aproximado o viceversa.
*/
int main() {
    double a, b, res;

    printf("Calcular el error matemático E = |a - b|\n\n");
    printf("Ingrese el valor de a:\n");
    scanf("%lf", &a);
    printf("Ingrese el valor de b:\n");
    scanf("%lf", &b);

    res = a < b ? b-a : a-b;

    printf("El error matemático de\n");
    printf("| %lf - %lf | es %lf\n", a, b, res);

    return 0;
}
```



```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Calcular el error matemático E = |a - b|

Ingrese el valor de a:
5
Ingrese el valor de b:
-3.6
El error matemático de
| 5.000000 - -3.600000 | es 8.600000

-----
Process exited with return value 0
Press any key to continue . . .
```

Ejercicios propuestos



PROBLEMA: Desarrolle un programa que pida la edad de una persona y muestre al grupo que pertenece. Considere la siguiente tabla:

Grupo	Rango de edad (Años)
Bebé	0-3
Niño	4-12
Adolescente	13-17
Joven	18-39
Adulto	40-59
Adulto Mayor	60+

```
INICIO
    edad: ENTERO
    ESCRIBIR "\t\nIngrese su edad"
    LEER edad
    SI edad >= 60 ENTONCES
        ESCRIBIR "\t\nAdulto Mayor"
    FIN SI
    DE LO CONTRARIO
        SI edad >= 40 ENTONCES
            ESCRIBIR "\t\nAdulto"
        FIN SI
        DE LO CONTRARIO
            SI edad >= 18 ENTONCES
                ESCRIBIR "\t\nJoven"
            FIN SI
            DE LO CONTRARIO
                SI edad >= 13 ENTONCES
                    ESCRIBIR "\t\nAdolescente"
                FIN SI
                DE LO CONTRARIO
                    SI edad >= 4 ENTONCES
                        ESCRIBIR "\t\nNiño"
                    FIN SI
                    DE LO CONTRARIO
                        ESCRIBIR "\t\nBebé"
                    FIN DE LO CONTRARIO
                FIN DE LO CONTRARIO
            FIN DE LO CONTRARIO
        FIN DE LO CONTRARIO
    FIN DE LO CONTRARIO
FIN
```


💜 **PROBLEMA:** Resolver una ecuación de segundo grado, tomar en cuenta que las raíces pueden ser complejas.

INICIO

a, b, c, r, x1, x2, dis: REAL

b:=0

c:=0

r:=0

x1:=0

x2:=0

dis:=0

HACER

 ESCRIBIR "Ingresa el término al cuadrado: "

 LEER a

MIENTRAS a=0

 ESCRIBIR "Ingresa el término lineal: "

 LEER b

 ESCRIBIR "Ingresa el término independiente: "

 LEER c

 dis:=(b*b)-(4*a*c)

 SI dis<0 ENTONCES

 x1:= (((b*b)-(4*a*c*(-1)))^(1/2))/(2*a)

 x2:= (-1)* (((b*b)-(4*a*c*(-1)))^(1/2))/(2*a)

 r:=((-1)*b)/(2*a)

 ESCRIBIR "\nRaices imaginarias\n\n\tx1:" r "+" x1 "i\n\tx2:" r, x2 "i"

 FIN SI

 EN CASO CONTRARIO

 x1:= ((-1*b)+(((b*b)-(4*a*c))^(1/2)))/(2*a)

 x2:= ((-1*b)-(((b*b)-(4*a*c))^(1/2)))/(2*a)

 ESCRIBIR "\nRaices\n\n\tx1:" x1 "\n\tx2:" x2

 FIN DE LO CONTRARIO

FIN

💜 **PROBLEMA:** Elabore el código en lenguaje C que lea tres longitudes y determine si forman o no un triángulo. Si es un triángulo determine de qué tipo de triángulo se trata entre: equilátero, isósceles o escaleno. Considere que para formar un triángulo se requiere que “el lado mayor sea menor que la suma de los otros dos lados”.

```
#include<stdio.h>
int main(){
    float mayor, menor1, menor2;
    do{
        printf("\nIngresa el lado a: ");
        scanf("%f", &menor1);
        printf("\nIngresa el lado b: ");
        scanf("%f", &menor2);
        mayor=(menor1>menor2)?menor1:menor2;
        menor1=(menor1<menor2)?menor1:menor2;
        printf("\nIngresa el lado c: ");
        scanf("%f", &menor2);
        menor2=(mayor<menor2)?mayor:menor2;
        mayor=(mayor>menor2)?mayor:menor2;
    }while (menor1==0 || menor2==0 || mayor==0);

    //printf("\n%f\n%f\n%f", mayor, menor1, menor2);
    if ((menor1+menor2)>mayor){
        //printf("\nTriangulo");
        if (menor1==menor2&&menor2==mayor){
            printf("\nTriangulo Equilatero");
        }else if (menor1==menor2 || menor2==mayor || mayor==menor1){
            printf("\nTriangulo Isosceles");
        }else{
            printf("\nTriangulo Escaleno");
        }
    }
    return 0;
}
```



PROBLEMA: En la siguiente tabla se muestran las categorías a las que pertenecen los signos del zodiaco:

SIGNO		ELEMENTO	SIGNO		ELEMENTO
1	Aries	Fuego	7	Libra	Aire
2	Tauro	Tierra	8	Escorpión	Agua
3	Géminis	Aire	9	Sagitario	Fuego
4	Cáncer	Agua	10	Capricornio	Tierra
5	Leo	Fuego	11	Acuario	Aire
6	Virgo	Tierra	12	Piscis	Agua

Se requiere escribir un algoritmo que:

- 1) Muestre el listado de los signos del zodiaco, con sus números asociados.
- 2) Pida al usuario un número (dato entero) asociado a un signo del zodiaco.
- 3) Muestre el elemento al que pertenece el signo del zodiaco seleccionado.

Nota: Si el número introducido por el usuario no está asociado a ningún signo del zodiaco, se mostrará el mensaje: "ERROR: número no está asociado a ningún signo."

INICIO

```
op: ENTERO
ESCRIBIR "\n\t\t\t\t\tSignos"
ESCRIBIR "\n\t\t\t\t\t1.Aries\t\t\t\t\t7.Libra"
ESCRIBIR "\n\t\t\t\t\t2.Tauro\t\t\t\t\t8.Escorpio"
ESCRIBIR "\n\t\t\t\t\t3.Géminis\t\t\t\t\t9.Sagitario"
ESCRIBIR "\n\t\t\t\t\t4.Cáncer\t\t\t\t\t10.Capricornio"
ESCRIBIR "\n\t\t\t\t\t5.Leo\t\t\t\t\t11.Acuario"
ESCRIBIR "\n\t\t\t\t\t6.Virgo\t\t\t\t\t12.Piscis\n"
LEER op
SI op=1 || op=5 || op=9 ENTONCES
    ESCRIBIR "Elemento: Fuego"
FIN SI
EN CASO CONTRARIO
    SI op=2 || op=6 || op=10 ENTONCES
        ESCRIBIR "Elemento: Tierra"
    FIN SI
    EN CASO CONTRARIO
        SI op=3 || op=7 || op=11 ENTONCES
            ESCRIBIR "Elemento: Aire"
        FIN SI
        EN CASO CONTRARIO
            SI op=4 || op=8 || op=12 ENTONCES
                ESCRIBIR "Elemento: Agua"
            FIN SI
            FIN EN CASO CONTRARIO
        FIN EN CASO CONTRARIO
    FIN EN CASO CONTRARIO
FIN DE LO CONTRARIO
```

FIN

Conclusiones

- ✦ Hicimos uso constante de la estructura condicional e interpretamos la diferencia entre usar **if**, **if-else**, **switch** y la forma **ternaria** del lenguaje C.
- ✦ Aprendimos la interpretación del VERDADERO y FALSO de la computadora para entender como sigue el flujo de las sentencias dentro de la estructura condicional.