

## Estructuras de control selectivas

Aquí se mencionan las diferentes formas de manejar una selección, clasificando la estructura por su uso común.

Las estructuras de control selectivas son aquellas que evalúan una expresión, usualmente una condición booleana, y a partir del resultado permiten tomar decisiones entre una, dos o más opciones; a esto se le conoce como selección condicional. Existen tres tipos de estructuras selectivas: **selección simple**, **selección doble** y **selección múltiple**.

Una condición booleana es una expresión que puede tener como resultado sólo el valor de verdadero o de falso.

La condición puede utilizar datos de tipo entero, real o carácter y se forman generalmente utilizando los operadores relacionales, por ejemplo  $a > b$ ,  $5 == b$ .

## 4.1 Estructura selectiva simple (if)

Es aquella que después de evaluar una condición determina su valor, que es verdadero o falso, y sólo si el resultado de la condición es verdadero se realizará la instrucción o instrucciones definidas para la condición, su sintaxis es la siguiente:

1.  

```
if (condición 1 )  
    instrucción 1;
```
2.  

```
if (condición 1 )  
    {  
    instrucción 1;  
    instrucción 2;  
    instrucción 3;  
    }
```

El alcance sintáctico predeterminado para una estructura de control es de una instrucción, como se aprecia en la primera forma; cuando se requiere que se ejecute más de una, las instrucciones deberán agruparse mediante llaves, tal como se puede ver en la segunda forma.

**EJEMPLO 4.1** Determinar si un alumno aprobó un curso a partir del promedio que obtuvo de sus tres calificaciones de los parciales que se hicieron durante el semestre

```
#include <stdio.h>  
#include <conio.h>  
main()  
{  
    float cal1, cal2, cal3, prom;  
    printf("Dame la calificación del primer examen parcial");  
    scanf("%f",&cal1);  
    printf("Dame la calificación del segundo examen parcial");  
    scanf("%f",&cal2);  
    printf("Dame la calificación del tercer examen parcial");  
    scanf("%f",&cal3);
```

```
prom = (cal1 + cal2 + cal3) / 3;
if (prom >= 60)

printf ("Aprobo");

getch();
return 0;
}
```

Se puede observar que la condición se establece utilizando la variable *prom* que almacena el promedio del alumno, de tal modo que si el promedio calculado es mayor o igual que 60 la condición es verdadera y sólo entonces se realizará la instrucción asignada que es imprimir que el alumno aprobó; de lo contrario, el programa no realizará nada y simplemente terminará.

#### EJEMPLO 4.2 Imprimir si un número es positivo, negativo o cero

```
/*Positivo, negativo o cero*/
#include<stdio.h>
#include<conio.h>

main()
{
int num;
printf("Indica si el numero es positivo, negativo o cero\n");
printf("Dame el \n");
scanf("%i",&num);
if(num==0)
printf("Es cero\n");
if(num>0)
printf("Es positivo\n");
if(num<0)
printf("Es negativo\n");
getch();
return 0;
}
```

En este ejemplo, el usuario introducirá un número y en la pantalla aparecerá si éste es positivo, negativo o cero. Una vez que el usuario ya introdujo el número,

se almacena en la variable *num* y se evalúa la primera condición; si la variable *num* es igual que cero, entonces se imprimirá *Es cero*; después revisa la siguiente condición y si la variable *num* es mayor que cero, se imprimirá *Es positivo* y por último se evalúa la tercera condición y si la variable *num* es menor que cero se imprimirá *Es negativo*.

Independientemente de la condición que se cumpla, el programa evaluará las tres condiciones, pero se imprimirá el texto sólo donde la condición sea verdadera.

## Ejercicios resueltos

### EJERCICIO 4.1 Leer un número por el teclado y evaluar si es par o impar

#### Descripción

Introducir un número.

Evaluar si el número dividido entre dos tiene de residuo cero o uno.

Imprimir si el número es par o impar.

Variables		
Nombre	Tipo	Uso
<i>n</i>	Entero	Valor a revisar

#### Codificación

```

/*Par o impar*/
#include<stdio.h>
#include<conio.h>

main()
{
    int n;
    clrscr();
    printf( "Par o impar.\n\n" );
    printf( "Escribe el numero: " );
    scanf( "%d", &n);
    if(n % 2 == 0)
        printf("El numero %d es par.", n);

```

```
if (n % 2 != 0)
    printf("El numero %d es impar.",n);
    getch();
return 0;
}
```

### Explicación

Se introduce un número entero. Si al dividirse entre dos su residuo es cero, se cumple la primera condición, se imprime que el número es par y pasa a la siguiente condición. Si el número leído se divide entre dos y su residuo es diferente de cero, la condición no se cumple, por lo que no se ejecuta la siguiente instrucción y termina el programa.

Si introducimos un número impar, y al dividirse entre dos su residuo no es cero, la primera condición no se cumple y no sucede nada; enseguida se evalúa la siguiente condición; si al dividirse entre dos, su residuo es diferente de cero, se imprime que es impar y termina el programa.

### Ejecución

1.

*Par o impar.*

*Escribe el numero 8*

*El numero 8 es par.*

2.

*Par o impar.*

*Escribe el numero 15*

*El numero 15 es par*

## 4.2 Selectiva doble (if - else)

Es aquella que permite evaluar una condición booleana y elegir entre dos opciones. Si la condición es verdadera, ejecutará la instrucción que se encuentra a continuación del `if`, pero si la condición es falsa se ejecutará la instrucción que se encuentra a continuación del `else`. Por lo tanto, se seleccionan las instrucciones que se encuentran a continuación del `if` o las que están después de `else`, pero no ambas. La sintaxis es:

1.  

```
if (condición )
    instrucción;
    else
    instrucción;
```
2.  

```
if (condición )
    {
    instrucción 1;
    instrucción 2;
    }
    else
    {
    instrucción 3;
    instrucción 4;
    }
```

Recuerde que para cada estructura, si existe **más de una instrucción**, éstas se deberán agrupar con las llaves. En la forma (2) anterior se agrupan tanto para el `if` como para el `else`.

**EJEMPLO 4.3** Determinar si un alumno aprobó o reprobó un curso a partir del promedio que obtuvo en sus tres calificaciones parciales durante el semestre y mostrar la calificación

```
/*Aprobo o reprobó*/
#include <stdio.h>
#include <conio.h>
main()
{
    float cal1, cal2, cal3, prom;
    printf("Dame la calificación del primer examen parcial");
    scanf("%f",&cal1);
    printf("Dame la calificación del segundo examen parcial");
    scanf("%f",&cal2);
    printf("Dame la calificación del tercer examen parcial");
    scanf("%f",&cal3);
    prom = (cal1 + cal2 + cal3) / 3;
    if (prom >= 60)
```

```

printf ("Aprobo con %f",prom);
else
printf ("Reprobo con %f", prom);
getch();
return 0;
}

```

En este ejemplo nuevamente se calcula el promedio del alumno, el resultado de la variable *prom* será el valor que se utilice en la condición. Si el contenido de la variable *prom* es mayor o igual que 60 (condición verdadera) se imprime *Aprobo* y su respectivo promedio, de lo contrario *prom* se encuentra entre 0 y 59 (condición falsa), por lo que se imprimirá *Reprobo* con su promedio respectivo.

#### EJEMPLO 4.4 Convertir kilómetros a metros

```

/*Elegir cuál conversión realizar si de kilómetros a metros o viceversa*/
#include<stdio.h>
#include<conio.h>
main ()
{
float z,y;
int op;
printf("Conversion de kms a mts o mts a kms\n");
printf("Si deseas hacer la conversion de kms a mts elige la
opcion 1 y de mts a kms elige la opcion 2\n");
scanf("%d",&op);
if(op==1)
{
printf("Dame los kilometros a convertir\n");
scanf("%f",&z);
y=z*1000;
printf(" la conversion a metros es %f\n",y);
}
else
{
printf("Dame los metros a convertir\n");
scanf("%f",&y);
z=y/1000;

```

(continúa)

```

printf("La conversion a kilometros es %f\n",z);
}
getch();
return 0;
}

```

(continuación)

En este ejemplo, el usuario debe elegir la opción que desea realizar oprimiendo el número 1 o 2; esto se almacena en la variable *op*. Posteriormente el programa realizará las instrucciones correspondientes evaluando una estructura de control selectiva doble. Si la variable *op* es igual que uno se realizarán las siguientes instrucciones: aparecerá en la pantalla *Dame los kilometros a convertir* y una vez que el usuario introduzca la cantidad aparecerá en la pantalla *La conversion a metros es*, pero si la condición no es verdadera, entonces aparecerá en la pantalla *Dame los metros a convertir* y enseguida *La conversion a kilometros es* y el programa concluirá.

*Nota:* Cabe señalar que al elegir la opción 1, se realizará la conversión de kilómetros a metros, pero al elegir la opción 2 u oprimir cualquier tecla diferente a 1 se realizará la conversión de metros a kilómetros.

## Ejercicios resueltos

### EJERCICIO 4.2 Indicar si el año en que naciste fue bisiesto

#### Descripción

Introducir el año en que naciste.

Si el año introducido dividido entre cuatro da un residuo de cero, el año en que naciste fue bisiesto; de lo contrario, el año en que naciste no lo fue.

Variables		
Nombre	Tipo	Uso
<i>anio</i>	Entero	Año de nacimiento

#### Codificación

```

/* Año BISIESTO */
#include<stdio.h>
#include<conio.h>

```



```
main(){
    int anio;
    clrscr();
    printf( "Indica si el anio en que naciste fue bisiesto\n\n" );
    printf( "En que anio naciste " );
    scanf( "%d", &anio );
    if( anio % 4 == 0 )
        printf( "El anio en que naciste es bisiesto\n\n" );
    else
        printf( "El anio en que naciste NO es bisiesto\n\n" );

    getch();
    return 0;
}
```

### Explicación

Se ingresa el año en que naciste, si al dividirse entre cuatro su residuo es cero, la condición se cumple y se imprime *El anio en que naciste es bisiesto*.

Si el año introducido al dividirse entre cuatro produce un residuo distinto de cero, la condición no se cumple y se imprime *El anio en que naciste NO es bisiesto*.

### Ejecución

1.

*Indica si el anio en que naciste fue bisiesto.*

*En que anio naciste 1980.*

*El anio en que naciste es bisiesto.*

2.

*Indica si el anio en que naciste fue bisiesto.*

*En que anio naciste 1985.*

*El anio en que naciste NO es bisiesto.*

## EJERCICIO 4.3 Convertir grados de temperatura

### Descripción

Seleccionar una opción.

Evalúa la condición.

(continúa)

(continuación)

Realiza el cálculo.

Muestra el resultado.

Variables		
Nombre	Tipo	Uso
<i>Opc</i>	Entero	Opción de conversión.
<i>g</i>	Real	Grados.
<i>con</i>	Real	Conversión.

### Codificación

```

/*0F - 0C o 0C a 0F */
#include<stdio.h>
#include<conio.h>

main()
{
    float g, con;
    int opc;
    clrscr();
    printf( "Conversiones de grados de temperatura " );
    printf( "\n[1]0F - 0C \n[2]0C - 0F\n\n" );
    printf( "Selecciona una opcion: " );
    scanf( "%d",&opc );
    if( opc == 1 )
    {
        printf( "Introduce los 0F = " );
        scanf( "%f", &g );
        con = (g-32)/1.8;
        printf( "%f0F = %f0C", g, con );
    }
    else
    {
        printf( "Introduce los 0C = " );
        scanf( "%f", &g );
        con = g * 1.8 + 32;
        printf( "%f0C = %f0F", g, con );
    }
}

```

```

    }
    getch();
    return 0;
}

```

### Explicación

Se pide seleccionar un tipo de conversión; una vez elegida, se solicita la temperatura para realizar la operación de conversión, y finalmente se muestra el resultado.

### Ejecución

1. Conversiones de grados de temperatura.

[1]  $\theta F - \theta C$

[2]  $\theta C - \theta F$

Selecciona una opción 1

Introduce los  $\theta F = .$

$\theta F = \theta C$

2. Conversiones de grados de temperatura.

[1]  $\theta F - \theta C$

[2]  $\theta C - \theta F$

Selecciona una opción 2

Introduce los  $\theta C = .$

$\theta C = \theta F$

## 4.3 Selectiva doble anidada

Es aquella estructura que dentro del alcance de una condición tiene otra condición; en otras palabras, en un *if - else* se encuentra otro *if - else*.

- Cada condición será evaluada en el orden en el que va apareciendo: si la condición 1 es verdadera se ejecuta la primera instrucción y ya no se revisan las demás condiciones, de lo contrario se evalúa la siguiente condición —la 2— y si ésta es verdadera realizará su instrucción y así sucesivamente. El *else* final que contiene la última instrucción será ejecutado si ninguna de las condiciones anteriores fue

verdadera. Esta estructura es más útil cuando se tienen tres o más opciones. Es posible utilizar tantos anidamientos como se requiera. Su sintaxis es la siguiente:

```
if (condición 1 )
    instrucción 1;
    else
    if (condición 2)
        instrucción 2;
        else
        instrucción 3;
```

#### EJEMPLO 4.5 Indicar si el número leído es positivo, negativo o cero

```
/*Positivo, negativo o cero*/
#include<stdio.h>
#include<conio.h>
main()
{
    int num;
    printf("Indica si el numero es positivo, negativo o cero\n");
    printf("Dame el numero\n");
    scanf("%d",&num);
    if(num==0)
        printf("Es cero\n");
    else
        if(num>0)
            printf("Es positivo\n");
        else
            printf("Es negativo\n");
    getch();
    return 0;
}
```

Una vez almacenado el número en la variable *num*, se pasa a la siguiente instrucción donde se evalúa si *num* es igual que cero y en caso de cumplirse la condición se imprimirá *Es cero*. Si se cumplió la primera condición se sale del anidamiento y el programa termina; de lo contrario, se evaluará la siguiente condición: si *num* es mayor que cero se imprimirá *Es positivo*, se sale del anidamiento y el programa

termina. De lo contrario, se ejecutará la última instrucción donde se visualizará en la pantalla *Es negativo*.

**EJEMPLO 4.6** Leer dos números y si son iguales multiplicarlos; si el primero es mayor que el segundo, que se resten; si el primero es menor que el segundo, que se sumen

```
#include <stdio.h>
#include <conio.h>
main()
{
    int num1,num2,resul;
    printf("Dame el primer ");
    scanf("%d",&num1);
    printf("Dame el segundo ");
    scanf("%d",&num2);
    if (num1==num2)
    {
        resul = num1 * num2;
        printf ("La multiplicacion de los numeros es %d",resul);
    }
    else
        if (num1 > num2)
        {
            resul = num1 - num2;
            printf ("La resta de los numeros es %d", resul);
        }
        else
        {
            resul = num1 + num2;

            printf ("La suma de los numeros es %d", resul);
        }
    getch();
    return 0;
}
```

Inicia pidiendo los dos valores, a continuación se compara si son iguales, si esto es verdadero, se realiza la multiplicación de los números y ya no se evalúan las

En las siguientes condiciones saliendo del anidamiento, si la condición 1 fue falsa se evalúa la condición 2 y se compara si el primer número es mayor que el segundo, si esto se cumple se restan los valores y se sale del anidamiento. En caso de que la condición 2 sea falsa, se ejecutan las instrucciones a continuación del último `else`.

## Ejercicios resueltos

### EJERCICIO 4.4 Indicar el tipo de triángulo introducido

#### Descripción

Introducir los tres lados del triángulo.

Comparar si los tres lados son iguales o si dos de ellos son iguales.

Imprimir el tipo de triángulo.

Variables		
Nombre	Tipo	Uso
<i>l1</i>	Entero	Lado 1 triángulo.
<i>l2</i>	Entero	Lado 2 triángulo.
<i>l3</i>	Entero	Lado 3 triángulo.

#### Codificación

```

/*Clasificación de triángulos*/
#include<stdio.h>
#include<conio.h>

main()
{
    int l1, l2, l3;
    clrscr();
    printf("Clasifica un triangulo\n\n");
    printf("Escriba los lados del triangulo: " );
    scanf("%d %d %d", &l1, &l2, &l3);
    printf("\n");
    if( l1 == l2 && l2 == l3 )
        printf( "El triangulo es Equilatero" );

```

```
else
    if( l2 == l3 || l2 == l1 || l3 == l1 )
        printf( "El triangulo es Isosceles" );
    else
        printf( "El triangulo es Escaleno" );

getch();
return 0;
}
```

### Explicación

Se introducen los valores de los tres lados del triángulo. Se compara si son iguales en cuyo caso se cumple la primera condición y se imprime *El triangulo es Equilatero*. En caso contrario, se compara si dos lados son iguales; si lo son, es verdadera la condición y se imprime *El triangulo es Isosceles*. Si la segunda condición no se cumple, se imprimirá *El triangulo es Escaleno*.

### Ejecución

1.  
*Clasifica un triangulo.*  
*Escribe los lados del triangulo 7 7 7.*  
*El triangulo es Equilatero.*
2.  
*Clasifica un triangulo.*  
*Escribe los lados del triangulo 5 3 9.*  
*El triangulo es Escaleno.*

4

## 4.4 Selectiva múltiple (switch - case)

Es aquella estructura que permite elegir entre dos o más opciones, *switch* evalúa la expresión que se encuentra dentro de los paréntesis y el resultado se compara con valores alternativos.

El *switch* en la expresión lleva implícito el operador igual (`==`), por lo que compara si la expresión es igual a alguna de las opciones. Por lo tanto no se puede comparar utilizando otro operador relacional.

El tipo de dato de la expresión sólo puede ser entero o carácter; por lo tanto, las opciones deberán coincidir con el tipo de dato de la expresión.

*switch* compara el valor de la expresión con cada una de las opciones en el orden en que se encuentran. Cada opción se representa con la palabra reservada *case*, por lo tanto habrá tantos *case* como opciones. Una vez que encuentra la igualdad de la expresión con una opción se realizarán las instrucciones que están a continuación del *case* hasta encontrar un *break*; si no encuentra ningún valor igual a la expresión, realizará la instrucción asignada al *default*, si éste existe.

Se deberá utilizar la palabra reservada *break* al término de cada *case* para interrumpir la estructura y no revisar las siguientes opciones.

Como se mencionó anteriormente, el *default* se ejecutará cuando la expresión no coincida con ninguna opción. Sin embargo, será decisión del programador incluirla o no en su programa, ya que éste es opcional. La sintaxis es:

```
switch (expresion)
{
    case 1: instruccion 1;
        break;
    case 2: instruccion 2;
        break;
    .
    .
    case n: instruccion n;
        break;
    default:
        instruccion n + 1;
}
```

#### EJEMPLO 4.7 Indicar un día de la semana y que el programa escriba el número de día que le corresponde

```
#include <stdio.h>
#include <conio.h>
main()
{
    char dia;
    printf("Elige un dia de la semana:\n Lunes = L\n Martes = M\n
    Miercoles = I\n Jueves = J\n Viernes = V\n Sabado
```



```

    = S\n Domingo = D");
scanf("%c",&dia);
switch(dia)
{
case 'L': printf("# 1");
    break;
case 'M': printf("# 2");
    break;
case 'I': printf("# 3");
    break;
case 'J': printf("# 4");
    break;
case 'V': printf("# 5");
    break;
case 'S': printf("# 6");
    break;
case 'D': printf("# 7");
    break;
}
getch();
return 0;
}

```

4

El usuario elige en un menú un nombre de día de la semana; automáticamente, aparece en la pantalla el número de día de la semana que le corresponde.

En este ejemplo, el tipo de dato de la expresión es carácter. A continuación se puede observar que la instrucción *default* es opcional, y en este ejemplo no se utilizó.

#### EJEMPLO 4.8 Realizar la operación que se elige del menú visualizado

```

/*Calculadora*/
#include<stdio.h>
#include<conio.h>
main()
{
int x,y,op;
float z;
printf("Calculadora basica elige la operacion a realizar\n");

```

(continúa)

(continuación)

```
printf("1=Suma, 2=Resta, 3=Multiplicacion, 4=Division\n");
scanf("%d",&op);
switch(op)
{
case 1:printf("Suma\n");
printf("Teclea el primer \n");
scanf("%i",&x);
printf("Teclea el segundo \n");
scanf("%i",&y);
z=x+y;
printf("La suma es%f",z);
break;

case 2:printf("Resta\n");
printf("Teclea el primer \n");
scanf("%i",&x);
printf("Teclea el segundo \n");
scanf("%i",&y);
z=x-y;
printf("la resta es%f",z);
break;

case 3:printf("Multiplicacion\n");
printf("Teclea el primer \n");
scanf("%i",&x);
printf("Teclea el segundo \n");
scanf("%i",&y);
z=x*y;
printf("La multiplicacion es%f",z);
break;

case 4:printf("Division\n");
printf("Teclea el primer \n");
scanf("%i",&x);
printf("Teclea el segundo \n");
scanf("%i",&y);
z=x/y;
printf("La division es%f",z);
break;
```

```

    }
    getch();
    return 0;
}

```

Este programa produce una calculadora básica y despliega cuatro opciones; la opción elegida por el usuario se almacena en la variable *op*, y será la expresión que el *switch* emplee para comparar con cada una de las opciones siguientes; cuando *op* encuentre una igualdad en alguna opción, se realizarán las instrucciones que se encuentran en el *case*.

Por ejemplo, si el usuario elige la opción 3, en la variable *op* se almacenará el 3 y empezará a compararse con cada uno de los *case*. Cuando se evalúe la opción 3, el usuario habrá elegido multiplicar y mostrará *Tecllea el primer*, después *Tecllea el segundo* y para finalizar aparecerá *La multiplicacion es....*

## Ejercicios resueltos

4

### EJERCICIO 4.5 Elegir una figura geométrica y calcular su área

#### Descripción

Elegir una figura geométrica a partir del menú.

Solicitar los valores.

Mostrar el área.

Variables		
Nombre	Tipo	Uso
<i>n1</i>	<i>Real</i>	<i>Operando 1.</i>
<i>n2</i>	<i>Real</i>	<i>Operando 2.</i>
<i>r</i>	<i>Real</i>	<i>Resultado.</i>
<i>op</i>	<i>Carácter</i>	<i>Opción operación.</i>

#### Codificación

```

/*CALCULADORA*/
#include<stdio.h>
#include<conio.h>

```

(continúa)

(continuación)

```
main()
{
    float n1, n2, r;
    char op;
    clrscr();
    printf( "Figuras geometricas\n\n" );
    printf( "Elija su figura ( 1. triangulo\n 2. rectangulo\n 3.
        cuadrado,\n 4. circulo ): \n\n" );
    scanf( "%c",&op);
    switch( op )
    {
        case '1': printf("\ndame la base y la altura");
            scanf("%f%f",&n1,&n2);
            printf("\n el area es %.2f",n1*n2/2);
            break;
        case '2': printf("\ndame la base y la altura");
            scanf("%f%f",&n1,&n2);
            printf("\n el area es %.2f",n1*n2);
            break;
        case '3': printf("\ndame el lado");
            scanf("%f",&n1);
            printf("\nel area es %.2f",n1*n1);
            break;
        case '4': printf("\ndame el radio");
            scanf("%f",&n1);
            printf("\nel area es %.2f",3.1416*n1*n1);
            break;
        default: printf( "Opcion incorrecta" );
    }
    getch();
    return 0;
}
```

### Explicación

El dato importante a conocer es el de la figura geométrica. El operador que se introduce en el ejemplo es 1, que corresponde al área del triángulo; a continuación, la variable selector se compara con la opción 1, al ser iguales, se solicita el valor de la base y la altura, después se muestra el resultado.

### Ejecución

Figuras geometricas

Elija su figura ( 1. triangulo

2. rectangulo

3. cuadrado

4. circulo ):

1

dame la base y la altura 2 3

el area es 3.00

### EJERCICIO 4.6 Imprimir el salario real de un trabajador

#### Descripción

Introducir las horas trabajadas.

Introducir el puesto que desempeña.

Indicar el salario real; si el trabajador gana más de \$8,000 a la semana reducir 20% de su salario.

4

Variables		
Nombre	Tipo	Uso
op	Carácter	Puesto.
sal	Real	Salario total.
sdes	Real	Salario descuento.
h	Entero	Número de horas.

Constantes		
Nombre	Tipo	Uso
D	Entero	Precio hora director.
G	Entero	Precio hora gerente.
S	Entero	Precio hora supervisor.
EV	Entero	Precio hora ejecutivo.
SEC	Entero	Precio hora secretaria.

(continúa)

(continuación)

**Codificación**

```
#include<stdio.h>
#include<conio.h>
#define D 600
#define G 450
#define S 300
#define EV 150
#define SEC 80

main(){
float sdes, sal;
int h;
char op;
clrscr();

printf( "NOMINA\n\n" );
printf( "Horas trabajadas en la semana: " );
scanf( "%d", &h );
printf( "Indica el puesto que desempeñas: " );
printf( "D = Director\n G = Gerente\n S = Supervisor\n E =
Ejecutivo de ventas\n T = Secretaria \n");
scanf( "%c", &op );
switch(op)
{
case 'D': sal = D * h;
break;
case 'G': sal = G * h;
break;
case 'S': sal = S * h;
break;
case 'E': sal = EV * h;
break;
case 'T': sal = SEC * h;
break;
}
if ( sal > 8000 )
{
sdes = sal * .8;
```

```

printf("Tu salario con el 20 por ciento de descuento es:
    $ %.2f ",sdes);
}
else
printf ("Tu salario es: $ %.2f",sal);
getch();
return 0;
}

```

### Explicación

En este programa se solicitan dos datos al usuario: la cantidad de horas trabajadas y el puesto. A continuación se calcula el salario bruto; y si éste es mayor que \$8,000 se hará un descuento del 20%. Finalmente se muestra el salario neto.

### Ejecución

1.

NOMINA

Horas trabajadas en la semana: 2

Indica el puesto que desempeñas:

D = Director

G = Gerente

S = Supervisor

E = Ejecutivo de ventas

T = Secretaria

D

Tu salario es: \$1200.00

2.

NOMINA

Horas trabajadas en la semana: 30

Indica el puesto que desempeñas:

D = Director

G = Gerente

S = Supervisor

E = Ejecutivo de ventas

T = Secretaria

S

Tu salario con el 20 por ciento de descuento es: \$7200.00

**EJERCICIO 4.7** Indicar el signo zodiacal a partir de una fecha**Descripción**

Introducir día y mes de nacimiento.

Mostrar el signo zodiacal que le corresponda.

Variables		
Nombre	Tipo	Uso
<i>dia</i>	Entero	Día de nacimiento.
<i>mes</i>	Entero	Mes de nacimiento.

**Codificación**

```

/*ANIO BISIESTO Y SIGNO ZODIACAL*/
#include<stdio.h>
#include<conio.h>

main(){
    int dia, mes;
    clrscr();
    printf( "Se mostrara tu signo zodiacal\n\n" );
    printf( "Cuando naciste? (dd/mm) " );
    scanf( "%d/%d", &dia, &mes );
    printf( "\nTu signo zodiacal es: " );
    switch( mes ){
        case 1: if( dia <= 20 ) printf( "Capricornio" );
                else printf( "Acuario" );
                break;
        case 2: if( dia <= 20 ) printf( "Acuario" );
                else printf( "Piscis" );
                break;
        case 3: if( dia <= 20 ) printf( "Piscis" );
                else printf( "Aries" );
                break;
        case 4: if( dia <= 20 ) printf( "Aries" );
    }
}

```



```
        else printf( "Tauro" );
        break;
    case 5: if( dia <= 20 ) printf( "Tauro" );
        else printf( "Geminis" );
        break;
    case 6: if( dia <= 20 ) printf( "Geminis" );
        else printf( "Cancer" );
        break;
    case 7: if( dia <= 20 ) printf( "Cancer" );
        else printf( "Leo" );
        break;
    case 8: if( dia <= 20 ) printf( "Leo" );
        else printf( "Virgo" );
        break;
    case 9: if( dia <= 20 ) printf( "Virgo" );
        else printf( "Libra" );
        break;
    case 10: if( dia <= 20 ) printf( "Libra" );
        else printf( "Escorpion" );
        break;
    case 11: if( dia <= 20 ) printf( "Escorpion" );
        else printf( "Sagitario" );
        break;
    case 12: if( dia <= 20 ) printf( "Sagitario" );
        else printf( "Capricornio" );
        break;
    }
    getch();
    return 0;
}
```

### Explicación

Se solicita *dia* y *mes* de nacimiento. En la variable *mes*, se almacena el número de mes y a continuación se buscará la igualdad en algún *case*, en caso de encontrarla se evalúa el día, y dependiendo del resultado se muestra el signo.

(continúa)

(continuación)

## Ejecución

Se mostrara tu signo zodiacal.

Cuando naciste? (dd/mm) 5 4

Tu signo zodiacal es Aries

## Resumen

Las estructuras de control selectivas son aquellas que evalúan una condición booleana y a partir del resultado permiten tomar decisiones entre una, dos o más opciones; a esto se le conoce como selección condicional. Existen tres tipos de estructuras selectivas: selección simple, selección doble y selección múltiple.

La estructura de control selectiva simple es aquella que después de evaluar una condición determina su valor, que es verdadero o falso, y sólo si el resultado de la condición es verdadero se realizará la instrucción o instrucciones definidas para la condición *if*.

La estructura selectiva doble es aquella que permite evaluar una condición booleana y elegir entre dos opciones. Si la condición es verdadera ejecutará las instrucciones a continuación del *if*, pero si es falsa se ejecutará la instrucción que se encuentra a continuación del *else*.

La estructura de control selectiva doble anidada es aquella estructura que dentro del alcance de una condición tiene otra condición; en otras palabras en un *if - else* se encuentra otro *if - else*. Esta estructura es útil cuando se tienen tres o más opciones. Es posible utilizar tantos anidamientos como se requiera.

La estructura de control selectiva múltiple es aquella que permite elegir entre dos o más opciones. *switch* evalúa la expresión que se encuentra dentro de los paréntesis y el resultado de ésta se compara con valores alternativos. El tipo de dato de la expresión sólo puede ser entero o carácter; por lo tanto, las opciones deberán coincidir con el tipo de dato de la expresión. Se deberá utilizar la palabra reservada *break* al término de cada *case* para interrumpir la estructura e impedir que las siguientes opciones e instrucciones sigan ejecutándose.

## Evaluación

### I. Conteste las siguientes preguntas.

1. ¿Cómo se define la estructura de control selección?
2. ¿Cuántas opciones de respuesta puede tener una condición booleana?
3. ¿Es posible escribir una selección dentro de otra?
4. ¿Es válido evaluar una condición de selección y si es falsa evaluar otra condición?
5. En el *switch*, ¿es válido escribir varias opciones *case* y al final de ese conjunto escribir *break*?
6. La instrucción *default*, ¿es indispensable en el *switch*?
7. Es la estructura selectiva que se utiliza para ejecutar una acción cuando la condición es verdadera y otra acción cuando la condición es falsa:
  - a) Selectiva simple
  - b) Selectiva doble anidada
  - c) Selectiva doble
  - d) Selectiva múltiple
8. Estructura de control que, si no encuentra ningún valor igual a la expresión, realiza la instrucción o instrucciones asignadas por *default* (si ésta existe).
  - a) Selectiva simple
  - b) Selectiva doble
  - c) Selectiva múltiple
  - d) Selectiva doble anidada
9. Estructura de control que, después de evaluar una condición, si su valor es verdadero realiza una o más acciones.
  - a) Selectiva simple
  - b) Selectiva doble
  - c) Selectiva múltiple
  - d) Selectiva doble anidada
10. Estructura de control que, dentro de una condición, tiene otra condición a evaluar. Se utiliza cuando hay tres o más opciones.
  - a) Selectiva simple
  - b) Selectiva doble
  - c) Selectiva múltiple
  - d) Selectiva doble anidada

### II. Escriba un programa que muestre lo siguiente.

Simule el tradicional juego de piedra, papel o tijera utilizando las estructuras de control selectivas. Incluya la librería *time* y *stdlib*, además del comando *srand*.

En este juego de azar participan dos personas buscando un ganador. Las reglas son las siguientes:

- Tijera corta papel gana
- Papel envuelve a piedra gana
- Piedra rompe tijera gana
- Con cualquier opción diferente pierde

## Ejercicios propuestos

### I. Codifique el siguiente programa.

1. Mostrar los signos zodiacales, validando que el mes introducido esté entre 1 y 12, y los días entre 1 y 31, según el mes.

### II. Diga cuál es el resultado de la ejecución de los siguientes programas.

#### 1.

```
/*Compra de camisas*/
#include <stdio.h>
#include <conio.h>

main()
{
    int num_camisas;
    float prec,tot_comp,tot_pag;
    printf ("CAMISAS POLO\nIngresar el precio de la camisa ");
    scanf ("%f",&prec);
    printf ("Ingresar el numero de camisas a comprar ");
    scanf ("%d",&num_camisas);
    tot_comp=num_camisas*prec;
    if (num_camisas>=3)
    {
        tot_pag=tot_comp-tot_comp*.2;
    }
    else
    {
        tot_pag=tot_comp-tot_comp*.1;
    }
}
```

```
}  
printf ("El total a pagar es de $%f",tot_pag);  
getch();  
return 0;  
}
```

2.

```
/*Descuento en el almacén*/  
#include <stdio.h>  
#include <conio.h>  
  
main()  
{  
float compra,desc,tot_pag;  
printf ("LINDA ALMACEN\nIngresa el total de tu compra ");  
scanf ("%f",&compra);  
if (compra>1000)  
{  
desc=compra*.2;  
}  
else desc=0;  
tot_pag=compra-desc;  
printf ("Total a pagar:\n$%f\nGRACIAS POR TU  
PREFERENCIA",tot_pag);  
getch();  
return 0;  
}
```

3.

```
/*Capital con intereses*/  
#include <stdio.h>  
#include <conio.h>  
  
main()  
{  
float p_int,cap,tot_i,capf;  
char op;  
printf ("BANCO NACIONAL\nIngresa tu capital ");
```

```

scanf ("%f",&cap);
printf ("Ingresa la tasa de intereses ");
scanf ("%f",&p_int);
tot_i=cap*p_int;
if (tot_i>7000)
{
    printf ("Deseas reinvertir tu capital?\nS=Si\nN=No ");
    scanf ("%s",&op);
    switch (op)
    {
        case 'S':capf=cap+tot_i;
            printf ("TU INVERSION SERA DE $%f\nGRACIAS POR TU
PREFERENCIA",capf);
            break;
        case 'N':printf ("TU CAPITAL ESTA SEGURO\nGRACIAS POR TU
PREFERENCIA");
            }
    }
else printf ("TU CAPITAL ESTA SEGURO\nGRACIAS POR TU PREFERENCIA");
getch();
return 0;
}

```

### III. Complete los siguientes programas.

1.

```

/*Hospital*/
#include <stdio.h>
#include <conio.h>
#define e1 25
#define e2 16
#define e3 20
#define e4 32

main()
{
    int tipoenf,edad,dias,costot;
    printf ("HOSPITAL CERCA DEL CIELO\nIndica tu

```

```

enfermedad\n1=Intoxicacion\n2=Gripa\n3=Alergia\n4=Diabetes ");
scanf ("___",&tipoenf);
printf ("Indica cuantos dias estuviste hospitalizado ");
scanf ("%d",&dias);
printf ("Indica tu edad ");
scanf ("%d",&edad);
switch (____)
{
    case 1:costot=dias*e1;
        break;
    case 2:costot=dias*___;
        break;
    case 3:costot=dias*e3;
        ____;
    case 4:costot=dias*e4;
}
if ((edad>=14)__(edad<=22))
{
    costot=costot*1.1;
}
printf ("El costo de tu tratamiento y estancia es de $ %d",costot);
getch();
return 0;
}

```

2.

```

/*Llantas marca X*/
#include <stdio.h>
#include <conio.h>

main()
{
    ____ n_ll;
    float tot;
    printf ("Llantas X\n_____ ");
    scanf ("%d",&n_ll);
    if (n_ll<5)
    {

```

```

tot=n_ll*300;
printf ("El precio de cada llanta es de $300.00\nTotal a
pagar:\n$_____\nGRACIAS POR TU PREFERENCIA",tot);
}
if ((n_ll>=5)&&(____))
{
tot=n_ll*250;
printf ("El precio de cada llanta es de $250.00\nTotal a
pagar:\n$%f\nGRACIAS POR TU PREFERENCIA",tot);
}
if (n_ll>10)
{
tot=_____;
printf ("El precio de cada llanta es de $200.00\nTotal a
pagar:\n$%f\nGRACIAS POR TU PREFERENCIA",tot);
}
getch();
return 0;
}

```

## 3.

```

/*Horas extras*/
#include <stdio.h>
#include <conio.h>

main()
{
int ht;
float pph,tp,he,pe,pd,pt;
printf ("EMPRESA PATITO\nIngresa el pago por hora ");
scanf ("%f",____);
printf ("Ingresa el numero de horas trabajadas ");
scanf ("%d",____);
if (ht<=40)
{
tp=ht*pph;
}
}

```



```
he=ht-40;
if (he<=8)
{
pe=he*pph*2;
}
else
{
pd=8*pph*2;
pt=(he-8)*____*3;
pe=pd+pt;
}

printf ("El total a pagar en el sueldo es:\n$%f",tp);
getch();
return 0;
}
```

