

## PRÁCTICA INSTRUCCIONES DE SELECCIÓN

---

### Objetivos

- a) *El alumno será capaz de realizar programas que involucren la toma de decisiones y éstas llevarán a realizar diferentes procesos con el fin de alcanzar la solución de una tarea.*
- b) *El alumno conocerá la utilidad de la sentencia if como una alternativa doble que controla la ejecución de dos conjuntos de instrucciones.*
- c) *El alumno controlará la ejecución de varios conjuntos de instrucciones mediante el valor final de una expresión.*

*Al final de esta práctica el alumno podrá:*

1. *Realizar exitosamente programas que involucren la toma de decisiones.*

### Antecedentes

1. Conocer y manejar los diversos tipos de datos.
2. Saber realizar un programa básico en C.

### Introducción

En C existen tres tipos básicos de instrucciones capaces de controlar el flujo de ejecución de un programa: Las sentencias *if-else*, *switch-case* y la expresión condicional *?:*

### Sentencia if-else

La forma más sencilla de este tipo de sentencia es un *if* simple, y tiene la siguiente sintaxis:

```
if (expresión)
    Sentencia;
```

En la instrucción anterior, si la expresión es cierta se ejecuta la sentencia, en caso contrario se omite. En ambos casos, el flujo del programa continúa inmediatamente después de la sentencia. Debe tomarse en cuenta que en C una expresión es cierta si su valor es diferente de cero y es falsa si su valor es cero.

Muy frecuentemente se requiere ejecutar una sentencia diferente cuando la expresión es falsa. Para este caso particular se emplea la palabra reservada *else* como se muestra enseguida:

## PRÁCTICA INSTRUCCIONES DE SELECCIÓN

---

```
if (expresión)
    Sentencia1;
else
    Sentencia2;
```

En esta instrucción, si la expresión es cierta, se ejecuta la *Sentencia1*, en caso contrario, se ejecuta la *Sentencia2*; en ambos casos el flujo del programa continúa inmediatamente después de la segunda sentencia. Si se desea ejecutar un conjunto de proposiciones y no sólo una, es necesario colocarlas entre llaves, como se muestra a continuación:

```
if (Expresión)
{
    Sentencia_1.0;
    Sentencia_1.1;
}
else
{
    Sentencia_2.0;
    Sentencia_2.1;
}
```

Las sentencias *if-else* pueden anidarse cuantas veces sea necesario, tanto en la rama en donde la expresión lógica es verdadera como en la que es falsa. Esto permite construir expresiones de múltiples alternativas como la que se muestra a continuación:

```
if (Expresión1)
    Sentencia_1;
else if (Expresión2)
    Sentencia_2;
else if (Expresión3)
    Sentencia_3;
else if (Expresión4)
    Sentencia_4;
else
    Sentencia_5;
```

Ejemplo 1: Resolución de una ecuación de segundo grado

## PRÁCTICA INSTRUCCIONES DE SELECCIÓN

---

Una ecuación de segundo grado es de la forma  $ax^2 + bx + c = 0$ , para obtener su solución es necesario basarse en la fórmula  $x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ .

Con base en el valor arrojado por el discriminante  $d = b^2 - 4ac$  se sabe cómo son las dos raíces que resuelven la ecuación de segundo grado.

Se observan tres casos distintos:

- Si  $d > 0$  las dos raíces son reales y distintas:  $x_1 = \frac{-b + \sqrt{d}}{2a}$  y  $x_2 = \frac{-b - \sqrt{d}}{2a}$ .
- Si  $d = 0$  las dos raíces son reales e iguales:  $x_1 = x_2 = \frac{-b}{2a}$ .
- Finalmente, si  $d < 0$  las dos raíces son complejas conjugadas. Donde las partes real e imaginaria son :  $real = \frac{-b}{2a}$  e  $imaginaria = \frac{\sqrt{-d}}{2a}$ .

```
/*Ejemplo 1. Programa que resuelve una ecuación de segundo grado*/
#include <stdio.h>
#include <math.h>
int main(void)
{
    double a, b, c;
    double d, x1, x2, real, imaginaria;
    printf("Programa que resuelve una ecuación de segundo grado\n");
    printf("Escribe el valor del coeficiente A\n");
    scanf("%lf", &a);
    printf("Escribe el valor del coeficiente B\n");
    scanf("%lf", &b);
    printf("Escribe el valor del coeficiente C\n");
    scanf("%lf", &c);
    d = b*b - 4.0*a*c; /*Se calcula el discriminante b^2-4ac */
    if (d > 0.0) /*El discriminante es positivo*/
    {
        x1 = (-b + sqrt(d)) / (2.0*a); /*Se calculan las dos raíces, X1 y X2*/
        x2 = (-b - sqrt(d)) / (2.0*a);
        printf("\nExisten dos raíces reales: \nX1= %5.3lf \nX2= %5.3lf\n", x1, x2);
    }
    else if (d < 0.0) /*El discriminante es negativo*/
    {
        real = -b / (2.0*a);
        imaginaria = sqrt(-d) / (2.0*a);
        printf("\nExisten dos raíces complejas:\n");
    }
}
```

## PRÁCTICA INSTRUCCIONES DE SELECCIÓN

---

```
printf("X1= %5.3lf+%5.3lfi \nX2= %5.3lf%5.3lfi\n",real, imaginaria, real, -imaginaria);
}
else
{
    x1 = -b/(2.0*a); /*El discriminante es cero*/
    printf("\nLas dos raíces son iguales \nX1=X2= %5.3lf\n", x1);
}
}
```

### Sentencia switch-case

Este tipo de sentencia permite seleccionar múltiples alternativas dando mayor claridad a las posibles ramificaciones que la sentencia *if-else*, especialmente cuando existe un número considerable de decisiones. Sin embargo, se paga un precio por ello; se puede probar sólo una expresión entera. La sintaxis general de *switch-case* es la siguiente:

```
switch (Expresión entera)
{
    case valor_1:
        Sentencia 1.0;
        Sentencia 1.1;
        break;
    case valor_2:
        Sentencia 2.0;
        Sentencia 2.1;
        break;
    case valor_3:
        Sentencia 3.0;
        Sentencia 3.1;
        break;
    default: /* opcional*/
        Sentencia 4.0;
        Sentencia 4.1;
}
```

La sentencia inicia en la palabra reservada *switch*, colocando entre paréntesis la expresión que será evaluada. Esta expresión necesariamente tiene que ser de tipo `int` o `char` y el resultado será comparado con valores alternativos que siguen a cada instrucción *case*. Esta palabra reservada identifica los diferentes valores que serán comparados con el valor de la expresión entera en *switch*, cada valor debe ser una cantidad entera constante diferente, y debe culminar con el operador `:` (dos puntos). Una vez que la expresión colocada entre paréntesis coincide con uno de los valores, se ejecutan las sentencias correspondientes hasta encontrar la sentencia *break*, la cual indica el fin de dichas

## PRÁCTICA INSTRUCCIONES DE SELECCIÓN

---

sentencias. Posteriormente, el flujo del programa continúa inmediatamente después de la cerradura de las llaves del *switch*. Es importante mencionar que si se omite la instrucción *break*, se ejecutarán también todas las sentencias pertenecientes a la instrucción *case* que se encuentre enseguida.

La palabra *default* es otra palabra reservada que pertenece al *switch*, esta sentencia es opcional y con ella se está indicando que si ningún valor coincidió con el valor de la expresión de prueba se deberán ejecutar las sentencias que se encuentren inmediatamente debajo de *default*. Obsérvese que también los dos puntos acompañan a *default*.

Ejemplo 2. Realizar una operación sobre dos números.

Suponer que se lee un carácter y dos números enteros, si el carácter leído es cualquiera de los siguientes operadores aritméticos +, -, \*, /, se calcula la operación correspondiente con los dos números, si es cualquier otro, se muestra un mensaje de error.

Observar que se tienen cuatro posibles opciones de operación, sin embargo, sólo una de ellas puede realizarse a la vez, cuando el carácter es distinto a los cuatro mencionados, se ejecuta algo distinto a una operación aritmética.

/\*Ejemplo 2. Programa que realiza una operación sobre dos números.\*/

```
#include <stdio.h>
int main(void)
{
    char c;
    int a,b;
    printf("Dame un carácter\n");
    scanf("%c",&c);
    printf("Dame dos números enteros separados por comas\n");
    scanf("%d, %d",&a,&b);
    switch (c)
    {
        case '+':
            printf("Resultado de la suma %d",a+b);
            break;
        case '-':
            printf("Resultado de la resta %d",a-b);
            break;
        case '*':
            printf("Resultado de la multiplicación %d",a*b);
            break;
        case '/':
            printf("Resultado de la división %d",a/b);
            break;
        default:
```

## PRÁCTICA INSTRUCCIONES DE SELECCIÓN

---

```
        printf("error");
    }
}
```

### Ejemplo 3. Menú

Es posible que en ocasiones se tenga un menú, en este caso, el usuario puede seleccionar la opción que desee dentro de las posibles presentadas.

Para realizar un menú debe plantearse primero el orden como serán listadas las opciones y la forma en que serán seleccionadas -mediante letras, símbolos, números-. Debe considerarse que aunque existan  $n$  opciones sólo se ejecutará una de ellas -la previamente seleccionada-, siempre debe tomarse en cuenta el caso en que el usuario oprima alguna tecla errónea.

/\*Ejemplo 3. Programa que muestra un menú y según la opción seleccionada realiza el cálculo correspondiente\*/

```
#include <stdio.h>
#include <math.h>
int main(void)
{
    int x;
    double num;
    printf("\n 1. El triple");
    printf("\n 2. El cuadrado");
    printf("\n 3. Logaritmo natural");
    printf("\n 4. Logaritmo decimal");
    printf("\n 5. Seno");
    printf("\n 6. Coseno");
    printf("\n\n Escriba el número de la opción que desee: ");
    scanf("%d",&x);
    printf("\n\nEscribe el número: ");
    scanf("%lf",&num);

    switch(x)
    {
        case 1:
            printf("\nEl triple de %lf es %lf",num,3*num);
            break;
        case 2:
            printf("\nEl cuadrado de %lf es %lf",num,num*num);
            break;
        case 3:
            printf("\nEl logaritmo neperiano de %lf es %lf",num,log(num));
            break;
```

## PRÁCTICA INSTRUCCIONES DE SELECCIÓN

---

```
case 4:
    printf("\nEl logaritmo decimal de %lf es %lf",num,log10(num));
    break;
case 5:
    printf("\nEl seno de %lf es %lf",num,sin(num));
    break;
case 6:
    printf("\nEl coseno de %lf es %lf",num,cos(num));
    break;
default:
    printf("\n\nEsto no es ninguna opción ");
    break;
}
}
```

### Expresión condicional ?:

Otra forma muy utilizada para controlar el flujo de ejecución de un programa es mediante el operador **?:**. Ésta es una alternativa a la instrucción *if-else* y tiene la siguiente sintaxis:

Expresión de prueba ? Expresión\_1 : Expresión\_2;

Si la expresión de prueba es verdadera, es decir, distinta de cero, se evalúa la primera expresión y el resultado permanece como efecto de toda la expresión condicional. En el caso contrario, cuando la expresión de prueba es falsa, es decir, si su valor es cero, se evalúa la segunda expresión y el resultado permanece como efecto de toda expresión condicional.

Por ejemplo, la siguiente expresión asigna a la variable *max* el valor máximo de dos cantidades.

```
float a = 4.1, b = 12.3, max;
```

```
max = (a > b) ? a : b;
```

Observar que lo anterior es equivalente al código que sigue:

```
if (a > b)
    max = a;
else
    max = b;
```

### Errores más comunes que se pueden cometer:

---

Elaborada por:

M.C. Cintia Quezada Reyes [cintia@fi-b.unam.mx](mailto:cintia@fi-b.unam.mx)

M.C. Vicente Fuentes Gea [vfg@servidor.unam.mx](mailto:vfg@servidor.unam.mx)

Ing. Alejandra Vargas E. De los M. [alejandrav@correo.unam.mx](mailto:alejandrav@correo.unam.mx)

## PRÁCTICA INSTRUCCIONES DE SELECCIÓN

---

1. Colocar al finar del paréntesis de la expresión del *if* un punto y coma.
2. Colocar un *else* asociado a un *if* aun cuando no es necesario.
3. Olvidar encerrar entre llaves el conjunto de instrucciones correspondientes al *if* o al *else*.

### Ejercicios propuestos

**Nota: Realizar el algoritmo antes de elaborar el programa**

1. Modificar el ejemplo 2 de la práctica, para que se visualice en pantalla que no se puede realizar esta operación porque el divisor es igual a cero
2. Escribir un programa que lea la calificación de un examen por teclado y devuelva la calificación no numérica correspondiente. La calificación podrá ser: No aprobado (0-4.99), Aprobado (5-6.99), Notable (7-8.99), Sobresaliente (9-9.99) o Excelente (10). Realizar este ejercicio utilizando la sentencia de control switch.
3. Repetir el ejercicio anterior pero utilizando la sentencia de control if-else.
4. Mejorar el ejercicio anterior de modo que si el usuario introduce un valor menor que cero o un valor mayor que 10 se muestre por pantalla un mensaje de error.
5. Escribir un programa que lea cuatro números cualesquiera y determine cuál es el mayor. También deberá considerar el caso en el que los números sean iguales.
6. Escribir un programa que lea tres números diferentes cualesquiera y muestre por pantalla el mayor y el menor.
7. Escribir un programa que lea cuatro números enteros y determine cuál es el menor. También debe considerar el caso en el que los números sean iguales.
8. Escribir un programa que lea cinco números cualesquiera y emita un mensaje indicando si están o no ordenados en orden creciente.
9. Escribir un programa que permita introducir por teclado tres números enteros y responda si los números son iguales.
10. Escribir un programa que permita introducir por teclado tres letras y responda si existen al menos dos letras iguales.
11. Realizar un programa que aconseje qué ropa se debe poner el usuario en función de los datos de temperatura y humedad introducidos por teclado. La tabla de valores que debe seguir el programa para aconsejar, es la siguiente:

Temperatura	Humedad	Ropa
25	90	Pantalón largo y camisa
10	50	Pantalón de pana y chaleco
2	45	Pantalón, chaleco y abrigo
30	85	Pantalón corto y camiseta

Realizar este ejercicio utilizando la sentencia de control switch.

12. Repetir el ejercicio anterior utilizando la sentencia de control if –else
13. Realizar un programa en el que el usuario introduce un número entero y el programa responde si el número es par o impar.



**PRÁCTICA**  
**INSTRUCCIONES DE SELECCIÓN**

---

14. Realizar un programa que pida cinco números y muestre por pantalla el siguiente menú:
1. Suma
  2. Resta
  3. Multiplicación
  4. División
- El usuario debe elegir una opción y el programa mostrar el resultado de la operación realizada.
15. Mejorar el programa anterior para los casos en los que el usuario elige una opción distinta de las 4 posibles (por ejemplo: “Opción no válida”).
16. Plantear un programa que sirva para calcular el cateto de un triángulo rectángulo a partir del otro cateto y la hipotenusa, considerar que:  
El programa pide el valor de la hipotenusa y el valor de un cateto.  
Si el cateto es mayor que la hipotenusa, el programa muestra un mensaje de error, en caso contrario el programa da como resultado el valor del otro cateto.  
Recordar que  $c^2 = a^2 + b^2$ .
17. Realizar un programa que sirva para calcular el área de un triángulo  $A_t = \frac{b * h}{2}$  o el área de un rectángulo  $A_r = b * h$  o el área de un círculo  $A_c = \pi r^2$ .
18. Realizar un programa que obtenga el complemento a dos de un número binario de 5 bits.
19. En un supermercado se va a poner en marcha la “Promoción Bolitas”, la cual consiste en que al llegar a la caja el cliente y proceder a efectuar el pago correspondiente a sus compras, se le invita a sacar una bolita al azar de una caja virtual y dependiendo del color de la bolita obtendrá un descuento aplicable al total de su cuenta.

Color	Descuento
Negro	10 %
Verde	25 %
Amarillo	50 %
Azul	75 %
Rojo	100%

Realizar el programa con las siguientes características:

- El cliente debe sacar una bolita de la caja (totalmente al azar).
- El sistema debe mostrarle al cliente la bolita e indicarle el porcentaje del descuento obtenido.
- El sistema le solicita a la cajera teclear el importe de la cuenta total del cliente.
- El sistema debe aplicar el descuento correspondiente a la cuenta del cliente e indicar cuál es la cantidad a pagar.

**PRÁCTICA**  
**INSTRUCCIONES DE SELECCIÓN**

---

20. Escribir un programa que permita jugar a doble o nada, en donde el jugador apuesta una cantidad y tira una moneda. Si sale cara obtiene el doble de la cantidad apostada. Si sale cruz la pierde toda.
21. Escribir un programa que pida un número entero y determine si es múltiplo de 2 y de 5.
22. Escribir un programa que, dado el nombre o número del mes, y la información de si el año es bisiesto, se muestre en la pantalla el número de días del mes.
23. Diseñar un programa que permita realizar diferentes tipos de conversiones de monedas. El usuario debe seleccionar un tipo de conversión desde el menú principal.  
Por ejemplo:
  1. Dólares a pesos
  2. Pesos a dólares
  3. Pesos a euros
  4. Euros a pesos
  5. Dólares a euros
  6. Euros a dólares
24. Realizar un programa donde el usuario introduzca un número y se compare con el que genere el sistema al azar, debe indicarse si ambos números son iguales o no.

9`dfcZgcf`dcXfz`XjgY< Uf`gi`g`dfcd]cg`Y`YfW]cgzgjYa dfY`mWUbXc`WVfUdcf`Vta d`Ytc`Y`cV`Yhj`c`XY`UdfzVWV'