

1. Estructuras de control selectivas

Al finalizar la presente sesión de laboratorio, el estudiante debe ser capaz de

- Modificar el orden de ejecución de las sentencias de un programa a través de sentencias **if**
- Utilizar la cláusula **else** de la sentencia **if**
- Resolver problemas implementando instrucciones **if**
- Manejar los operadores de asignación compuestos

1.1. Preliminares

Los operadores de relación son empleados generalmente para construir expresiones de relación usadas en sentencias selectivas e iterativas.

Las sentencias en C se ejecutan normalmente de arriba hacia abajo conforme aparecen en el código fuente. Una sentencia de control modifica el orden de ejecución de las sentencias de un programa.

1.2. La sentencia **if**

```
        if ( expre )  
{  
sentencia ;  
}
```

Si *expre* se evalúa como verdadera, *sentencia* es ejecutada; caso contrario, no es ejecutada. En cualquier caso, la ejecución continúa con lo que viene después de }.

N.B.: Ambas líneas: **if (*expre*)** y *sentencia* ; se consideran que constituyen la sentencia **if** completa. No son sentencias separadas. Una sentencia **if** puede controlar la ejecución de múltiples sentencias, a través del uso de una **sentencia compuesta** o **bloque**.

1.2.1. La cláusula **else**

Una sentencia **if** puede opcionalmente incluir una cláusula **else**.

```
        if ( expre )  
  
sentencia1 ;
```

else

sentencia2 ;

Si *expre* se evalúa como verdadera, *sentencia1* es ejecutada; caso contrario, *sentencia2*

es ejecutada. Tanto *sentencia1* como *sentencia2* pueden ser sentencias compuestas o bloques.

Sentencia if anidada...

1.3. Valores de Verdadero

Se ha visto hasta ahora que se evalúa a 0 para representar falso y a 1 para verdadero. No obstante, es importante tener en cuenta que cualquier valor numérico es interpretado ya sea bien como verdadero o bien como falso cuando se espera un valor lógico de él. La regla es que un valor de cero representa falso y cualquier valor diferente de cero representa verdadero.

Los operadores de asignación compuestos proveen un método abreviado para combinar una operación aritmética binaria con una operación de asignación.

Ejemplo 1.1. $x += 5$; //es equivalente a

$x = x + 5$;

$x /= 2$; //es equivalente a

$x = x / 2$;

Ejercicio 1.1. Se pide ingresar un número entero y se muestre si este es par o impar.

Ejercicio 1.2. Se pide ingresar un entero positivo y muestre si es múltiplo de nueve o no.

Ejercicio 1.3. Se debe simular a una calculadora elemental: se pide un número, un operador (+, -, *, /) y otro número y a continuación se debe imprimir el resultado de la operación correspondiente. Si en caso se ingresa la división por cero, se debe mostrar un mensaje de alerta.

Ejercicio 1.4. El pago que recibe un carpintero depende de sus años de experiencia y del tipo de producto que produce según la siguiente tabla:

Años de experiencia	S/. por unidad	
	sillas	mesas
0 - 5	20	30
6 - 20	35	60
21 - más	25	40

Además, el carpintero recibe una bonificación especial de acuerdo a la cantidad que produce según la siguiente tabla:

Total de unidades producidas	bonificación (%)
1 - 5	0
6 - 20	20
21 - más	50

Se debe mostrar el pago de un carpintero al ingresar los años (número entero) de experiencia de dicho carpintero y la cantidad de sillas y mesas que produce. Por ejemplo, cuando un carpintero de 8 años de experiencia produce 3 sillas y 4 mesas, recibe $1,2(3 \times 35 + 4 \times 60)$ Soles.

Ejercicio 1.5. El pago que recibe un técnico de computadoras depende de su categoría y del tipo de computadora que repara según la siguiente tabla:

Categoría	S/. por unidad	
	Desktop	Laptop
A	20	30
B	30	50
C	50	100

Además, el técnico ofrece descuentos según la cantidad de computadoras que repara según la siguiente tabla:

Total de unidades reparadas	Descuento (%)
1 - 2	0
3 - 5	10
6 - 9	20
10 - más	25

Se debe mostrar el pago de un técnico de computadoras y el descuento que da el mismo al ingresar su categoría y la cantidad de desktops y laptops que produce. Por ejemplo, cuando un técnico de computadoras de categoría B repara 2 desktops y 3 laptops, su pago es de $189 = 0,9(2 \times 30 + 3 \times 50)$ soles y el descuento fue de $21 = 0,1(2 \times 30 + 3 \times 50)$ soles.

Ejercicio 1.6. El bronce es una aleación metálica de cobre y estaño en la que el primero constituye su base y el segundo aparece en una proporción del 3 al 20 %. En la siguiente tabla se da a conocer tres tipos de bronce:

Tipo de bronce	Proporción (%)	
	Cobre	Estaño
A	80	20
B	90	10
C	95	5

El precio de una reliquia de bronce se incrementa en proporción a su precio actual según su antigüedad conforme a la siguiente tabla:

Años de antigüedad	Incremento (%)
0 - 5	0
6 - 20	50
21 - 50	100
51 - más	200

Si los precios de un kilogramo de cobre y estaño son de 500 y 200 soles, respectivamente, se debe mostrar el precio de una reliquia de bronce al ingresar su tipo, antigüedad en número entero de años y peso en kilogramos.

Ejercicio 1.7. Construir un programa que acepte un número en el rango de 1 a 99. Luego, muestre dicho número en romano. Por ejemplo:

Se ingresa : 68
Se obtiene : LXVIII

Ejercicio 1.8. Sobre el siguiente programa:

```
1  #include <stdio.h>
2
3  void main ()
4  {
5      if (1)
6          printf("uno\n");
7      else
8          if (2)
9              printf("dos\n");
10         else
11             printf("tres\n");
12 }
```

1. ¿Cuántas sentencias if hay?
2. ¿Qué es lo que se muestra en la pantalla?
3. Si en la línea 5 se cambia el 1 por el 0, ¿qué se muestra en la pantalla?
4. Si en las líneas 5 y 8 se cambian dichos números por el 0, ¿qué se muestra en la pantalla?

Ejercicio 1.9. se pida ingresar tres números enteros. Luego determine y muestre el mayor de ellos.

Ejercicio 1.10. se pida ingresar tres números enteros. Luego muestre dichos números ordenados de menor a mayor.

Ejercicio 1.11. se pida ingresar un año después de 1600 y muestre si es bisiesto o no.

Ejercicio 1.12. se pida ingresar un ángulo en sexagesimal y nos indique en qué cuadrante se encuentra dicho ángulo.

Ejercicio 1.13. Sobre el siguiente programa:

```
1      #include <stdio.h>
2
3      int main ( )
4      {
5          if (-2)
6              printf("uno\n");
7          else
8              if (3.14)
9                  printf("dos\n");
10             else
11                 printf("tres\n");
12         return -2;
13     }
```

1. ¿Cuántas sentencias if hay? ¿Por qué?
2. ¿Qué es lo que se muestra en la pantalla?
3. Si en la línea 5 se cambia el -2 por el 0, ¿qué se muestra en la pantalla?
4. Si en las líneas 5 y 8 se cambian dichos números por el 0, ¿qué se muestra en la pantalla?

Ejercicio 1.14. El plan postpago de un teléfono celular incluye 100 minutos y 500 mb de internet por 20 soles mensuales. Por cada minuto extra se cobra S/0.20 y por cada mb adicional S/0.05. Además, el plan incluye un pago por un seguro del equipo de S/1.50, y toda esta cuenta (incluyendo el pago del seguro) es sujeta al IGV que es del 18 por ciento. Escriba un programa que lea la cantidad de minutos y mb usados en un mes por un usuario. Luego, muestre la cuenta básica, la cuenta por minutos adicionales, la cuenta por mb adicionales, por el seguro, por el impuesto y la cuenta total a pagar. Todos los montos deben ser mostrados empleando 2 decimales.

Ejercicio 1.15. Sobre el siguiente programa:

```

1  #include <stdio.h>
2
3  int main ()
4  {
5      int a, b, c, aux;
6
7      printf (" Ingrese 3 nmeros: ");
8      scanf ("%d %d %d", &a, &b, &c);
9
10     if (a > b)
11     {
12         aux = a;
13         a = b;
14         b = aux;
15     }
16     if (c < a)
17         printf (" %d %d %d \n", c,a,b);
18     else if (b < c)
19         printf (" %d %d %d \n", a,b,c);
20     else
21         printf (" %d %d %d \n", a,c,b);
22     return 0;
23 }

```

- (a) ¿Qué pasaría si intentamos compilar sin la sentencia `#include`?
- (b) ¿Cuántas sentencias `if` hay? ¿Por qué?
- (c) ¿Qué es lo que se mostraría en la pantalla al ingresar los siguientes valores?
- a) 3 3 3
 - b) 1 2 3
 - c) 2 7 5
 - d) 8 4 2
- (d) ¿Cuál es el papel de la variable `aux`?