

CAPÍTULO 4

Estructuras de control selectivas (if, if-else, switch)

Introducción

Los programas definidos hasta este punto se ejecutan de modo secuencial. La ejecución comienza con la primera sentencia de la función y prosigue hasta la última sentencia, cada una de las cuales se ejecuta una sola vez. Para la resolución de problemas de tipo general se necesita la capacidad de controlar cuáles son las sentencias que se ejecutan y en qué momentos. Las *estructuras o construcciones de control* controlan la secuencia o flujo de ejecución de las sentencias. Las estructuras de control se dividen en tres grandes categorías en función del flujo de ejecución: *secuencia*, *selección* y *repetición*.

Este capítulo considera las *estructuras selectivas o condicionales* —sentencias `if` y `switch`— que controlan si una sentencia o lista de sentencias se ejecutan en función del cumplimiento o no de una condición. Para soportar estas construcciones, C++ tiene el tipo lógico `bool`.

4.1. Estructuras de control

Las **estructuras de control** controlan el flujo de ejecución de un programa o función. Las instrucciones o sentencias se organizan en tres tipos de estructuras de control que sirven para controlar el flujo de la ejecución: *secuencia*, *selección (decisión)* y *repetición*. Una **sentencia compuesta** es un conjunto de sentencias encerradas entre llaves (`{` y `}`) que se utiliza para especificar un flujo secuencial.

4.2. La sentencia `if`

En C++, la estructura de control principal de selección es una sentencia `if`. La sentencia `if` tiene dos alternativas o formatos posibles. El formato más sencillo tiene la sintaxis siguiente:

`if (Expresión) Acción`

↓

Expresión lógica que determina si la acción se ha de ejecutar

↘

Acción se ejecuta si la expresión lógica es verdadera

La sentencia *if* funciona de la siguiente manera. Si *Expresión* es verdadera, se ejecuta *Acción*; en caso contrario no se ejecuta *Acción*.

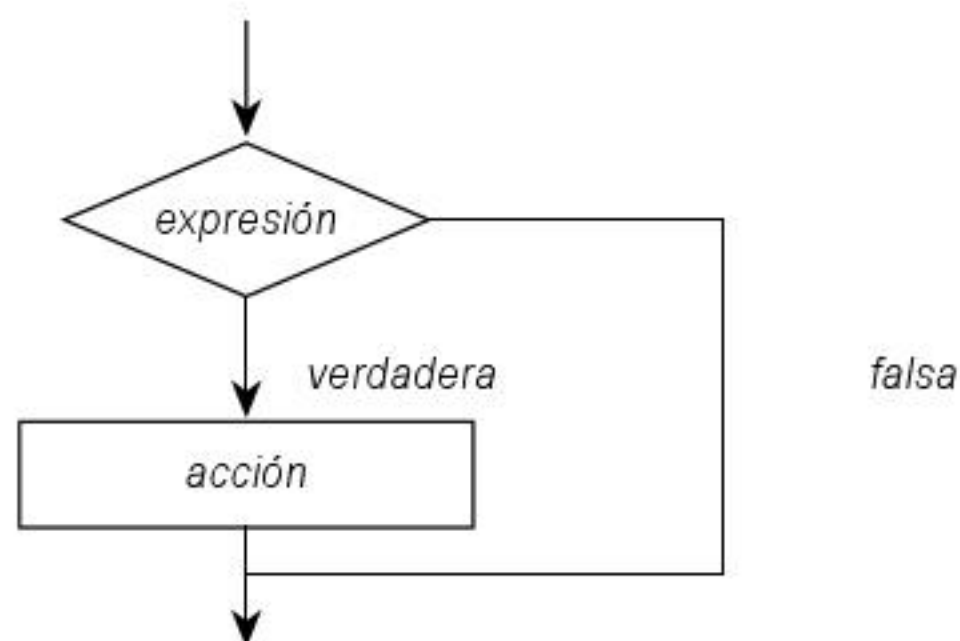


Figura 4.1. Diagrama de flujo de una sentencia básica *if* 3e2ff75c30d222a35aca2773f3e6e40d
ebrary

EJEMPLO 4.1. Prueba de divisibilidad de dos números enteros.

```
#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    int numero1, numero2;

    cout << "Introduzca dos enteros:";
    cin >> numero1 >> numero2;
    if (numero1 % numero2 == 0)
        cout << numero1 << " es divisible por " << numero2 << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Resultados de ejecución del programa anterior

3e2ff75c30d222a35aca2773f3e6e40d
ebrary Introduzca dos enteros: 25 5
25 es divisible por 5.

EJEMPLO 4.2. Decidir si un número es mayor que 10.

```
#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    float numero;

    cout << "Introduzca un número :";
    cin >> numero;
    // comparar número con diez
    if (numero > 10)
        cout << numero << "es mayor que 10" << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

EJEMPLO 4.3. Leer tres números enteros y visualizar el mayor.

Se realiza mediante un *algoritmo voraz*, de tal manera, que el mayor de un solo número es siempre el propio número. Si ya se tiene el mayor de una lista de números, y si a esa lista se le añade un nuevo número entonces el mayor o bien es el que ya teníamos, o bien es el nuevo.

```
#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    int n1, n2, n3, mayor;

    cout << " introduzca tres numeros ";
    cin >> n1 >> n2 >> n3;
    mayor = n1; // candidato a mayor
    if (mayor < n2) // nuevo mayor
        mayor = n2;
    if (mayor < n3) // nuevo mayor
        mayor = n3;
    cout << " el mayor es : " << mayor << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Resultados de ejecución del programa anterior:

```
introduzca tres numeros 2 7 4
el mayor es :7
Presione una tecla para continuar . . .
```

EJEMPLO 4.4. Lee un dato real y visualiza su valor absoluto.

```
include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    float Dato;

    cout << "Introduzca un numero: ";
    cin >> Dato;
    if (Dato < 0)
        Dato = - Dato; //Cambio de signo
    cout << " Valor absoluto siempre positivo " << Dato << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

4.3. Sentencia if de dos alternativas: if-else

El formato de la sentencia if-else tiene la siguiente sintaxis:

```
if (expresión) Acción1 else Acción2
```


Cuando se ejecuta la sentencia `if-else`, se evalúa *Expresión*. Si *Expresión* es verdadera, se ejecuta *Acción₁* y en caso contrario se ejecuta *Acción₂*.

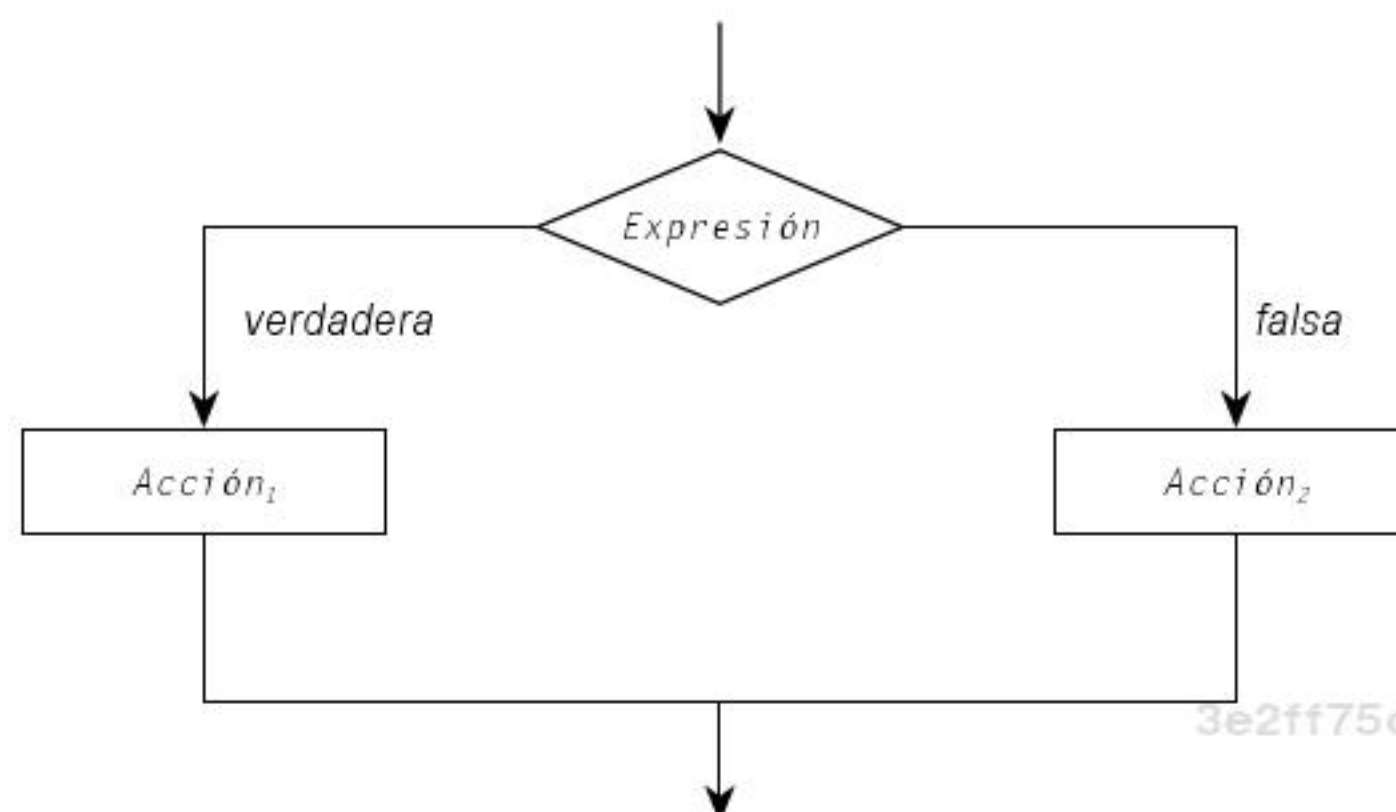
3e2ff75c30d222a35aca2773f3e6e40d
ebrary

Figura 4.2. Diagrama de flujo de la representación de una sentencia `if-else`

EJEMPLO 4.5. Leer una nota, y visualizar *baja* si es menor que 100 y *alta* en otro caso.

```

#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    int nota;

    cout << " dame nota: ";
    cin >> nota;
    if (nota < 100)
        cout << " Baja ";
    else
        cout << "Alta";
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

EJEMPLO 4.6. Leer el salario y los impuestos. Visualizar el salario neto.

```

#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    float Salario_bruto, Impuestos, Salario_netto;

    cout << " introduzca salario bruto e impuestos ";
    cin >> Salario_bruto >> Impuestos;           //lectura de datos
    if (Salario_bruto > 600)                       // restar los impuestos
        Salario_netto = Salario_bruto - Impuestos;
    else                                           //no restar impuestos

```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

```

        Salario_netto = Salario_bruto;
        cout << Salario_netto << Salario_bruto;           // visualización
        system("PAUSE");
        return EXIT_SUCCESS;
    }

```

4.4. Sentencias if-else anidadas

Una sentencia `if` es anidada cuando la sentencia de la rama *verdadera* o la rama *falsa*, es a su vez es una sentencia `if`. Una sentencia `if` anidada se puede utilizar para implementar decisiones con varias alternativas o multi-alternativas.

Sintaxis

```

if (condición1)
    sentencia1;
else if (condición2)
    sentencia2;
    .
    .
    .
else if (condiciónn)
    sentencian;
else
    sentencia;

```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

EJEMPLO 4.7. Leer la calificación (nota) en una variable real, y mediante `if` anidados escribir el resultado:

Menor que 0 o mayor que 10

0 a < 5.0

5 a < 6.5

6.5 a < 8.5

8.5 a < 10

10 a < 10.5

< 0 o > 10

Error en nota

Suspenso

Aprobado

Notable

Sobresaliente

Matrícula de honor

Error en nota

```

#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    float nota;
    cout << "dame nota: ";
    cin >> nota;
    if(( nota < 0.0 ) || ( nota > 10 ))
        cout << "Error en nota ";
    else if ( nota < 5.0 )
        cout << "Suspenso";
    else if( nota < 6.5 )
        cout << "Aprobado";
    else if ( nota < 8.5 )
        cout << "Notable";
}

```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

```
else if ( nota < 10)
    cout << "Sobresaliente";
else
    cout << "Matricula de Honor";
system("PAUSE");
return EXIT_SUCCESS;
}
```

4.5. Sentencia de control switch

La sentencia `switch` es una sentencia C++ que se utiliza para hacer una selección entre múltiples alternativas.

Sintaxis

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

```
switch (selector)
{
    case etiqueta1 : sentencias1;
    case etiqueta2 : sentencias2;
    .
    .
    .
    case etiquetan : sentenciasn;
    default:      sentencias;           // opcional
}
```

La expresión *selector* debe ser un tipo ordinal (`int`, `char`, `bool` pero no `float` o `string`). Cada *etiqueta* es un valor único, constante, y cada etiqueta debe tener un valor diferente de los otros. La expresión de control o *selector* se evalúa. Si su valor es igual a una de las etiquetas *case* —por ejemplo, *etiqueta_i*— entonces la ejecución comenzará con la primera sentencia de la secuencia *secuencia_i* y continuará hasta que se encuentra el final de la sentencia de control `switch`, o hasta encontrar la sentencia `break`.

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

EJEMPLO 4.8. Sentencia `switch` para informar sobre la lectura de una opción dentro de un rango.

```
#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    int opcion;

    cout << "introduzca opcion entre 0 y 3:";
    cin >> opcion;
    switch (opcion)
    {
        case 0:
            cout << "Cero!" << endl;
            break;
        case 1:
            cout << "Uno!" << endl;
            break;
```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

```

        case 2:
            cout << "Dos!" << endl;
            break;
        case 3:
            cout << "Tres!" << endl;
            break;
        default:
            cout << "Fuera de rango" << endl;
    }
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

EJEMPLO 4.9. *Sentencia switch con caracteres.*

```

#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    char nota;

    cout << "Introduzca calificación (S, A, B, N, E) :";
    cin >> nota;
    switch (nota)
    {
        case 'E': cout << "Sobresaliente.";
                  break;
        case 'N': cout << "Notable.";
                  break;
        case 'B': cout << "Bien.";
                  break;
        case 'A': cout << "Aprobado.";
                  break;
        case 'S': cout << "Suspens.";
                  break;
        default:
            cout << "no es posible esta nota";
    }
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary3e2ff75c30d222a35aca2773f3e6e40d
ebrary

4.6. Expresiones condicionales: el operador ?:

Una expresión condicional tiene el formato $C ? A : B$ y es realmente una operación ternaria (tres operandos) en la que C , A y B son los tres operandos y $?:$ es el operador.

Sintaxis

$\text{condición} ? \text{expresión}_1 : \text{expresión}_2$

 condición $\text{expresión}_1 \mid \text{expresión}_2$

es una expresión lógica

son expresiones compatibles de tipos

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

Se evalúa *condición*, si el valor de *condición* es verdadera (distinto de cero) entonces se devuelve como resultado el valor de *expresión₁*; si el valor de *condición* es falsa (cero) se devuelve como resultado el valor de *expresión₂*.

EJEMPLO 4.10. *Sentencia ?: para decidir el orden de dos números.*

```
#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    int n1, n2;

    cout << " introduzca dos numeros ";
    cin >> n1 >> n2 ;
    n1 > n2 ? cout << n1 << " > " << n2
            : cout << n1 << " <= " << n2;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

```
#include <cstdlib>
#include <iostream>
using namespace std;
```

EJEMPLO 4.11. *Escribe el mayor de dos números usando ?:*

```
#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    int n1, n2, mayor;

    cout << " introduzca dos numeros ";
    cin >> n1 >> n2 ;
    mayor = n1 > n2 ? n1 : n2;
    cout << " el mayor es: " << mayor << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Resultado de ejecución

```
introduzca dos numeros 5 7
el mayor es: 7
Presione una tecla para continuar . . .
```

4.7. Evaluación en cortocircuito de expresiones lógicas

La *evaluación en cortocircuito* de una expresión lógica significa que se puede detener la evaluación de una expresión lógica tan pronto como su valor pueda ser determinado con absoluta certeza. C++ realiza la evaluación en cortocircuito con los operadores && y ||, de modo que evalúa primero la expresión más a la izquierda, de las dos expresiones unidas por && o bien

por `||`. Si de esta evaluación se deduce la información suficiente para determinar el valor final de la expresión (independiente del valor de la segunda expresión), el compilador de C++ no evalúa la segunda expresión. Esta característica permite, en general, disminuir el tiempo de ejecución.

EJERCICIOS

4.1. *¿Qué errores de sintaxis tiene la siguiente sentencia?*

```
if x > 25.0
    y = x
else
    y = z;
```

4.2. *¿Qué valor se asigna a `consumo` en la sentencia `if` siguiente si `velocidad` es 120?*

```
if (velocidad > 80)
    consumo = 10.00;
else if (velocidad > 100)
    consumo = 12.00;
else if (velocidad > 120)
    consumo = 15.00;
```

4.3. *¿Qué salida producirá el código siguiente, cuando se inserta en un programa completo?*

```
int primera_opcion = 1;
switch (primera_opcion + 1)
{
    case 1:
        cout << "Cordero asado\n";
        break;
    case 2:
        cout << "Chuleta lechal\n";
        break;
    case 3:
        cout << "Chuletón\n";
    case 4:
        cout << "Postre de pastel\n";
        break;
    default:
        cout << "Buen apetito\n";
}
```

4.4. *¿Qué salida producirá el siguiente código, cuando se inserta en un programa completo?*

```
int x = 2;

cout << "Arranque\n";
if (x <= 3)
    if (x != 0)
        cout << "Hola desde el segundo if.\n";
    else
        cout << "Hola desde el else.\n";
```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary3e2ff75c30d222a35aca2773f3e6e40d
ebrary3e2ff75c30d222a35aca2773f3e6e40d
ebrary

```
cout << "Fin\n";
cout << "Arranque de nuevo\n";
if (x > 3)
    if (x != 0)
        cout << "Hola desde el segundo if.\n";
    else
        cout << "Hola desde el else.\n";
cout << "De nuevo fin\n";
```

4.5. Escribir una sentencia if-else que visualice la palabra Alta si el valor de la variable nota es mayor que 100 y Baja si el valor de esa nota es menor que 100.

4.6. ¿Cuál es la salida de este segmento de programa?

```
int x = 1;

cout << x << endl;
{
    cout << x << endl;
    int x = 2;
    cout << x << endl;
    {
        cout << x << endl;
        int x = 3;
        cout << x << endl;
    }
    cout << x << endl;
}
```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

4.7. Escribir una sentencia if-else que clasifique un entero x en una de las siguientes categorías y escriba un mensaje adecuado:

$x < 0$ o bien $0 \leq x \leq 100$ o bien $x > 100$

4.8. Escribir un programa que determine si un año es bisiesto. Un año es bisiesto si es múltiplo de 4 (por ejemplo 1984). Sin embargo, los años múltiplos de 100 sólo son bisiestos cuando a la vez son múltiplos de 400 (por ejemplo, 1800 no es bisiesto, mientras que 2000 sí lo es).

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

PROBLEMAS

- 4.1. Escribir un programa que introduzca el número de un mes (1 a 12) y el año y visualice el número de días de ese mes.
- 4.2. Cuatro enteros entre 0 y 100 representan las puntuaciones de un estudiante de un curso de informática. Escribir un programa para encontrar la media de estas puntuaciones y visualizar una tabla de notas de acuerdo al siguiente cuadro:

Media	Puntuación
[90-100]	A
[80-90)	B
[70-80)	C
[60-70)	D
[0-60)	E

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

- 4.3. Se desea calcular el salario neto semanal de los trabajadores de una empresa de acuerdo a las siguientes normas:

Horas semanales trabajadas ≤ 38 , a una tasa dada.

Horas extras (38 o más), a una tasa 50 por 100 superior a la ordinaria.

Impuestos 0 por 100, si el salario bruto es menor o igual a 300 euros.

Impuestos 10 por 100, si el salario bruto es mayor de 300 euros.

- 4.4. Escribir un programa que lea dos números enteros y visualice el menor de los dos.
- 4.5. Escribir y comprobar un programa que resuelva la ecuación cuadrática ($ax^2 + bx + c = 0$).
- 4.6. Escribir un programa que lea tres enteros y emita un mensaje que indique si están o no en orden numérico.
- 4.7. Escribir un programa que lea los valores de tres lados posibles de un triángulo a , b y c , y calcule en el caso de que formen un triángulo su área y su perímetro, sabiendo que su área viene dada por la siguiente expresión:

$$\text{Área} = \sqrt{p(p-a)(p-b)(p-c)}$$

donde p es el semiperímetro del triángulo $p = (a + b + c)/2$

- 4.8. Escribir y ejecutar un programa que simule un calculador simple. Lee dos enteros y un carácter. Si el carácter es un $+$, se visualiza la suma; si es un $-$, se visualiza la diferencia; si es un $*$, se visualiza el producto; si es un $/$, se visualiza el cociente; y si es un $\%$ se imprime el resto.
- 4.9. Escribir un programa que calcule los ángulos agudos de un triángulo rectángulo a partir de las longitudes de los catetos.

SOLUCIÓN DE LOS EJERCICIOS

- 4.1. La expresión correcta debe ser la siguiente:

```
if(x > 25.0)
    y = x;
else
    y = z;
```

Por tanto, le falta los paréntesis en la expresión lógica y un punto y coma después de la sentencia de asignación $y = x$.

- 4.2. Si velocidad toma el valor de 120 entonces necesariamente consumo debe tomar el valor de 10.00, ya que se evalúa la primera condición y es cierta por lo que se ejecuta la sentencia `consumo = 10.00;`.
- 4.3. Aparece escrito Chuleta lechal. Si a primera opción se le asignara el valor de 2 entonces aparece escrito Chuletón y en la siguiente línea Postre de pastel, ya que case 3: no lleva la orden `break`.
- 4.4. La salida del programa es:

```
Arranque
Hola desde el segundo if.
Fin
Arranque de nuevo
De nuevo fin
```


4.5. *Una codificación es la siguiente:*

```
#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    int nota;

    cout << " dame nota: ";
    cin >> nota;
    if (nota < 100)
        cout << " Baja ";
    else if (nota > 100)
        cout << "Alta";
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary**4.6.** *La salida del programa es:***4.7.** *Una codificación es la siguiente:*

```
#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    int x;

    cout << " dato ";
    cin >> x;
    if (x < 0)
        cout << "es negativo\n";
    else if (x <= 100)
        cout << "0 <= x = %d <= 100";
    else
        cout << " > 100";
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary3e2ff75c30d222a35aca2773f3e6e40d
ebrary

- 4.8.** *La variable booleana `bisiesto` se pone a `true` si el año es bisiesto. Esto ocurre cuando es divisible el año por 400, o es divisible por 4 y no por 100.*

```
#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    int x;
    bool bisiesto;

    cout << " introduzca año entero ";
    cin >> x;
    if (x % 400 == 0)
        bisiesto = true;
    else if (x % 100 == 0)
        bisiesto = false;
    else
        bisiesto =(x % 4 == 0);
    if (bisiesto)
        cout << x << " es bisiesto\n";
    else
        cout << x << " no es un año bisiesto\n";
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

SOLUCIÓN DE LOS PROBLEMAS

3e2ff75c30d222a35aca2773f3e6e40d

ebrary

- 4.1.** *Para resolver el problema, se ha de tener en cuenta que el mes 2 corresponde a febrero que puede tener 29 o 28 días dependiendo de si es o no bisiesto el año correspondiente. De esta forma, además de leer el mes, se lee el año, y se decide si el año es bisiesto de acuerdo con lo indicado en el Ejercicio resuelto 4.8 para saber si el mes de febrero tiene 28 o 29 días. El resto de los meses tiene 31 días excepto abril, junio, septiembre y noviembre que corresponden a los meses 4, 6, 9 y 11.*

```
#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    int mes, ano;

    bool bisiesto;
    cout << " introduzca mes entre 1 y 12 ":
    cin >> mes;
    cout << " introduzca año entero ";
    cin >> ano;
```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

```

if (x % 400 == 0)
    bisiesto = true;
else if (ano % 100 == 0)
    bisiesto = false;
else
    bisiesto = (ano % 4 == 0);
if (mes == 2)
    if(bisiesto)
        cout << " tiene 29 dias\n";
    else
        cout << " tiene 28 dias\n";
    else
        if((mes == 4) || (mes == 6) || (mes == 9) || (mes == 11))
            cout << " tiene 30 dias \n";
        else
            cout <<" tiene 31 dias \n";
system("PAUSE");
return EXIT_SUCCESS;
}

```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

- 4.2.** *El programa que se escribe, lee las cuatro notas enteras, calcula la media real, y escribe la media obtenida y su puntuación de acuerdo con la tabla indicada usando if anidados.*

```

#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    int nota1, nota2, nota3, nota4;
    float media;

    cout << "Dame nota 1 ";
    cin >> nota1;
    cout << "Dame nota 2 ";
    cin >> nota2;
    cout << "Dame nota 3 ";
    cin >> nota3;
    cout << "Dame nota 4 ";
    cin >> nota4;
    media = (float)(nota1 + nota2 + nota3 + nota4) / (float)4;
    if(( media < 0) || ( media > 100 ))
        cout << "fuera de rango ";
    else if( media >= 90)
        cout << " media = " << media << " A";
    else if(media >= 80)
        cout << "media = " << media << " B";
    else if(media >= 70)
        cout << "media = " << media << " C";
    else if(media >= 60)
        cout << "media = " << media << " D";
}

```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

3e2ff75c30d222a35aca2773f3e6e40d
ebrary


```

else
    cout << "media = " << media << "E";
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

- 4.3.** *Se escribe un programa que lee las Horas, la Tasa, y calcula las horas extras, así como el SalarioBruto y el SalarioNeto de acuerdo con la especificación, visualizando los resultados.*

```

#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    float Horas, Extras, Tasa, SalarioBruto, SalarioNeto;

    cout << " dame Horas\n";
    cin >> Horas;
    if ( Horas <= 38 )
        Extras = 0;
    else
    {
        Horas = 38;
        Extras = Horas - 38;
    }
    cout << "introduzca Tasa\n";
    cin >> Tasa;
    SalarioBruto = Horas * Tasa + Extras * Tasa * 1.5;
    if (SalarioBruto < 50000.0)
        SalarioNeto = SalarioBruto;
    else
        SalarioNeto = SalarioBruto * 0.9;
    cout << " Salario bruto " << SalarioBruto << endl;
    cout << " Salario neto " << SalarioNeto << endl ;
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

Un resultado de ejecución es el siguiente:

```

dame Horas
45
introduzca Tasa
25
Salario bruto 950
Salario neto 855
Presione una tecla para continuar . . .

```

- 4.4.** *Se solicitan los dos números. Si numero1 es menor que numero2, la condición es “verdadera” (true); en caso contrario la condición es “falsa” (false). De este modo se visualiza numero1 cuando es menor que numero2.*

```

#include <cstdlib>
#include <iostream>

```

```
using namespace std;
int main(int argc, char *argv[])
{
    int numero1, numero2;

    cout << "Introduzca dos enteros:";
    cin >> numero1 >> numero2;
    if (numero1 < numero2)
        cout << numero1 << endl;
    else
        cout << numero2 << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

3e2ff75c30d222a35aca2773f3e6e40d

ebrary

4.5. Para resolver el problema se ha tenido en cuenta que:

- Si $a \neq 0$ se presentan tres casos: el primero con dos soluciones dadas por la fórmula que da la solución de la ecuación de segundo grado cuando el discriminante $d = b^2 - 4ac$ es positivo $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$. El segundo con una solución dada por la fórmula cuando el discriminante es cero $x = \frac{-b}{2a}$. El tercero con dos soluciones complejas, dadas por la fórmula $\frac{-b}{2a} \pm i \frac{\sqrt{b^2 - 4ac}}{2a}$ cuando el discriminante es negativo.
- Si $a = 0$ se presentan a su vez otros tres casos: el primero es cuando $b \neq 0$ cuya solución es $x = -\frac{c}{b}$. El segundo es cuando $b = 0$ y $c = 0$, que es evidentemente una identidad. El tercero cuando $b = 0$ y $c \neq 0$ que no puede tener solución.

```
#include <cstdlib>
#include <iostream>
#include <math.h>
using namespace std;

int main(int argc, char *argv[])
{
    float a, b, c, d, x1, x2;

    cout << "introduzca los tres coeficientes\n";
    cin >> a, b, c;
    if (a != 0)
    {
        d = b * b - 4 * a * c;
        if (d > 0)
        {
            cout << "dos soluciones reales y distintas\n";
            x1 = (-b + sqrt(d)) / (2 * a);
            x2 = (-b - sqrt(d)) / (2 * a);
            cout << " x1= " << x1 << " x2= " << x2 << endl;
        }
        else if (d == 0)
        {
            cout << "dos soluciones reales e iguales\n";
            x1 = (-b) / (2 * a);
            cout << " x= " << x1;
        }
    }
}
```

3e2ff75c30d222a35aca2773f3e6e40d

ebrary

3e2ff75c30d222a35aca2773f3e6e40d

ebrary

```

else
{
    cout << " no tiene solucion real\n";
    cout << " tiene dos soluciones complejas \n";
    x1 = -b / (2 * a);
    x2 = sqrt(-d) / (2 * a);
    cout << " primera solucion\n";
    cout << " parte real "<< x1 << endl;
    cout << " parte imaginaria "<< x2 << endl;
    cout << " segunda solucion\n";
    cout << " parte real "<< x1 << endl;
    cout << " parte imaginaria "<< -x2 << endl;
}
}
else if (b != 0)
    cout << " una solucion simple= " << -c / b;
else if (c == 0)
    cout << " se introdujo la identidad 0 = 0\n";
else
    cout << " sin solucion\n";
system("PAUSE");
return EXIT_SUCCESS;
}

```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

4.6. La codificación usa la sentencia **?:** y además la sentencia de selección **if**.

```

#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    int x, y, z;
    bool ordenados;
    cin >> x >> y >> z;
    ordenados = x >= y ? true : false;
    ordenados = ordenados && (y >= z ? true : false);
    if (ordenados)
        cout << " estan ordenados " << endl;
    else
        cout << " no ordenados " << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

4.7. Para que el triángulo exista debe cumplirse que los lados sean todos positivos y , además, que la suma de dos lados cualesquiera sea mayor que el otro lado. El programa obtenido comprueba que los datos leídos cumplen las condiciones, y escribe en caso de formar un triángulo su área y su perímetro.

```

#include <cstdlib>
#include <iostream>
#include <math.h> // contiene la función pow
using namespace std;

```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary


```

int main(int argc, char *argv[])
{
    float a, b, c, p, area;

    cout << "Introduzca el valor de los tres lados";
    cin >> a >> b >> c;
    if ((a <= 0) || (b <= 0) || (c <= 0) ||
        ((a + b) < c) || ((a + c) < b) || ((b + c) < a))
        cout << " Los lados no dan un triángulo \n";
    else
    {
        p =(a + b + c)/ 2;
        area = pow(p * ( p - a ) * ( p - b ) * ( p - c ) , 0.5);
        cout << "la solucion es\n";
        cout << "area = " << area << endl;
        cout << " perimetro = " << p * 2 << endl;
    }
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

4.8. *El programa lee los dos operadores, el operando y visualiza la salida de acuerdo con lo solicitado.*

```

#include <cstdlib>
#include <iostream>
#include <math.h>
using namespace std;

int main(int argc, char *argv[])
{
    int operand1, operando2;
    char operador;

    cout << " Introduzca dos numeros enteros ";
    cin >> operand1 >> operando2;
    cout << " Introduzca operador + - * / % ";
    cin >> operador;
    switch(operador)
    {
        case '+': cout << operand1 + operando2;
                  break;
        case '-': cout << operand1 - operando2;
                  break;
        case '*': cout << operand1 * operando2;
                  break;
        case '/': cout << operand1 / operando2;
                  break;
        case '%': cout << operand1 % operando2;
                  break;
        default: cout << " fuera de rando";
    }

    system("PAUSE");
    return EXIT_SUCCESS;
}

```

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

3e2ff75c30d222a35aca2773f3e6e40d
ebrary

- 4.9.** Se calcula la hipotenusa por la fórmula del teorema de pitágoras, y se obtiene el ángulo mediante la función inversa del seno que es `asin()` que se encuentra en `math.h`. Además, se convierte el valor devuelto por la función arco seno a grados (la función arco seno da su resultado en radianes).

```
#include <cstdlib>
#include <iostream>
#include <math.h>
#define pi 3.141592
using namespace std;

int main(int argc, char *argv[])
{
    float a, b, h, angulo;

    cout << "Introduce los lados ";
    cin >> a >> b;
    if ((a <= 0) || (b <= 0))
        cout << " no solucion\n";
    else
    {
        h = sqrt(a * a + b * b);
        angulo = 180 / pi * asin(a / h); // ángulo en grados
        cout << " hipotenusa = " << h << endl;
        cout << "          angulo = " << angulo << endl;
        cout << "otro angulo = " << 90 - angulo << endl;
    }
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Un resultado de ejecución es:

```
Introduce los lados 5 6
hipotenusa = 7.81025
          angulo = 39.8056
otro angulo = 50.1944
```

EJERCICIOS PROPUESTOS

- 4.1.** Explique las diferencias entre las sentencias de la columna de la izquierda y de la columna de la derecha. Para cada una de ellas deducir el valor final de `x`, si el valor inicial de `x` es 0.

```
if (x >= 0)
    x = x+1;
else if (x >= 1);
    x = x+2;
```

```
if (x >= 0)
    x = x+1;
if (x >= 1)
    x = x+2;
```

- 4.2.** ¿Qué hay de incorrecto en el siguiente código?

```
if (x = 0) cout << x << " = 0\n";
else cout << x << " != 0\n";
```

- 4.3.** ¿Cuál es el error del siguiente código?

```
if (x < y < z) cout << x << "<" << y << "<"
               << z << endl;
```

4.4. ¿Cuál es el error de este código?

```
cout << "Introduzca n:";
cin >> n;
if (n < 0)
    cout << "Este número es negativo. Pruebe
        de nuevo .\n";
```

```
cin >> n;
else
    cout << "conforme. n= " << n << endl;
```

4.5. Determinar si el carácter asociado a un código introducido por teclado corresponde a un carácter alfabético, dígito, de puntuación, especial o no imprimible.**PROBLEMAS PROPUESTOS****4.1.** El domingo de Pascua es el primer domingo después de la primera luna llena posterior al equinoccio de primavera, y se determina mediante el siguiente cálculo:
$$A = \text{año} \bmod 19$$

$$B = \text{año} \bmod 4$$

$$C = \text{año} \bmod 7$$

$$D = (19 * A + 24) \bmod 30$$

$$E = (2 * B + 4 * C + 6 * D + 5) \bmod 7$$

$$N = (22 + D + E)$$

donde N indica el número de día del mes de marzo (si N es igual o menor que 30) o abril (si es mayor que 31). Construir un programa que determine fechas de domingos de Pascua.

4.2. Construir un programa que indique si un número introducido por teclado es positivo, igual a cero, o negativo.**4.3.** Se quiere calcular la edad de un individuo, para ello se va a tener como entrada dos fechas en el formato *día* (1 a 31), *mes* (1 a 12) y *año* (entero de cuatro dígitos), correspondientes a la fecha de nacimiento y la fecha actual, respectivamente. Escribir un programa que calcule y visualice la edad del individuo. Si es la fecha de un bebé (menos de un año de edad), la edad se debe dar en

meses y días; en caso contrario, la edad se calculará en años.

4.4. Se desea leer las edades de tres de los hijos de un matrimonio y escribir la edad mayor, la menor y la media de las tres edades.**4.5.** Escribir un programa que acepte fechas escritas de modo usual y las visualice como tres números. Por ejemplo, la entrada 15, Febrero 1989 producirá la salida 15 02 1989.**4.6.** Escribir un programa que acepte un número de tres dígitos escrito en palabras y, a continuación, los visualice como un valor de tipo entero. La entrada se termina con un punto. por ejemplo, la entrada *doscientos veinticinco*, producirá la salida 225.**4.7.** Se desea redondear un entero positivo N a la centena más próxima y visualizar la salida. Para ello la entrada de datos debe ser los cuatro dígitos A, B, C, D , del entero N . Por ejemplo, si A es 2, B es 3, C es 6 y D es 2, entonces N será 2362 y el resultado redondeado será 2400. Si N es 2342, el resultado será 2300, y si $N = 2962$, entonces el número será 3000. Diseñar el programa correspondiente.