Objetivo de la práctica:
- Aprender el uso de estructuras condicionales.

*NOTA 1: Durante la práctica todos los ejercicios deberán ser guardados **temporalmente** en el directorio c:\tmp. Una vez finalizada la misma y transferidos los ficheros a un dispositivo de almacenamiento externo o enviados por correo, se deberá eliminar dicho directorio.*

CONCEPTOS BÁSICOS: Sentencias IF y SWITCH

Sentencia if

```
//Una alternativa
   if (a != 0)
       resultado = a / b;
//Dos alternativas
   if (a >= 0)
       cout << "positivo";
   else
       cout << "negativo";

//Multiples alternativas (if anidados)
   if (x < 0)
   {
       cout << "negativo";
       abs_x = -x;
   }
   else
       if (x == 0)
       {
           cout << "cero";
           abs_x = 0;
       }
       else
       {
           cout << "positivo";
           abs_x = x;
       }
```

Sentencia switch

```
switch (sig_car)
{
    case 'a':
        cout << "sobresaliente";
        break;

    case 'b':
        cout << "notable ";
        break ;

    case 'c':
        cout << "aprobado";
        break;

    case 'd':
        cout << "suspenso";
        break;

    default://otros casos
        cout << "no valido";
} // fin de switch
```

ERRORES FRECUENTES DE PROGRAMACIÓN

1. Uno de los errores más comunes en una sentencia **`if`** es utilizar operador de asignación '=' en lugar de un operador de igualdad "==".
2. En una sentencia **`if`** anidado cada cláusula **`else`** se corresponde con la **`if`** precedente más cercana. Por ejemplo, en el segmento de programación siguiente:

```
if (a > 0)
if (b > 0)
c = a + b;
else
c = a * b * c;
d = a * b;
```

¿Cuál es la sentencia **`if`** asociada a **`else`**?

El sistema más fácil para evitar errores es el sangrado, con lo que ya se aprecia que la cláusula **`else`** se corresponde a la sentencia que contiene condición: **`b > 0`**

```
if (a > 0)
    if (b > 0)
        c = a + b;
    else
        c = a * b * c;
d = a * b;
```

Una de las razones de utilizar la Guía de Estilo es la claridad de los programas y evitar errores innecesarios.

3. Será necesario utilizar corchetes para una sentencia **`if`** en caso de que haya más de una instrucción en dicho bloque:

```
if (a > 0)
{
    b = b / a;
    c = c – b;
}
```

4. El selector de una sentencia **`switch`** debe ser de tipo entero o compatible entero (ordinal). Así las constantes reales, como 2.4, -4.5, 3.1416, no pueden ser utilizadas en el selector.

5. Cuando se utiliza una sentencia **`switch`**, asegúrese que el selector de **`switch`** y las etiquetas **`case`** son del mismo tipo (`int`, `char` o `bool`, pero no `float`). Si el selector se evalúa a un valor no listado en ninguna de las etiquetas **`case`**, la sentencia **`switch`** no gestionará ninguna acción; por esta causa se suele poner una etiqueta **`default`** para resolver este problema.

SOBRE LA PRÁCTICA ANTERIOR:

Prueba el siguiente programa; si te parece que no funciona correctamente modifícalo hasta que funcione como debería.

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

int main()
{
    short c1;
    short c2 = 10;
    flota c3;

    c3 = c1 / c2;
    cout << c1 << " entre " << c2 << "=" << c3 << endl

    system("pause");
    return 0;
}
```

PROBLEMAS:

1. Escribe un programa que pida un número y nos clasifique éste según sea:

- Cero, mayor o menor que cero.
- Par o impar.
- Múltiplo de 5 o no múltiplo de 5.
- Múltiplo de 10 o no múltiplo de 10.

Por ejemplo para el número 25, deberá mostrarse por pantalla:

- El número introducido es mayor que cero.
- El número es impar.
- El número es múltiplo de 5.
- El número no es múltiplo de 10.

2. Una empresa de teléfonos móviles aplica una tarifa de 0.09 €/minuto (+IVA) a las llamadas que realizan sus abonados, pero teniendo en cuenta que el primer minuto siempre se cobra completo y que a partir del siguiente minuto se tarifa por segundos. Se pide, escribir un programa que dada como entrada la duración en segundos de una llamada indique el coste final de la misma (Impuestos incluidos).

3. Conociendo el siguiente método de evaluación de una asignatura ficticia:

- Un 50 % de su nota final corresponde a la nota del examen realizado.
- Un 30 % de su nota final corresponde a la nota obtenida en las prácticas realizadas.
- Un 20 % de su nota final corresponde a la nota de la actitud en clase.

Realiza un programa que pida las 3 notas, realice el cálculo oportuno y muestre por pantalla la nota final de la asignatura.
Tendremos también que tener en cuenta las siguientes restricciones:

- Para poder aprobar, la nota del examen debe ser como mínimo 4. Es decir que automáticamente si la nota del examen es inferior a 4 no calcularemos la nota final y mostraremos por pantalla un mensaje para indicar que no se ha llegado a 4 en el examen.

- De igual manera, para poder aprobar, tanto la nota de las prácticas realizadas, como la nota de la actitud en clase deberá ser como mínimo 5. Por tanto si alguna de estas notas es inferior a 5 no se calculará la nota media final y se mostrará un mensaje por pantalla indicando que no se ha superado el 5 en alguna de estas dos notas.

4. Amplía el ejercicio anterior de manera que después de calcular la nota media, la clasifique según el rango de notas siguiente y muestre la calificación final por pantalla:

- de 0 a 5 (sin incluir el 5): Insuficiente
- de 5 a 7 (sin incluir el 7): Aprobado
- de 7 a 9 (sin incluir el 9): Notable
- de 9 a 10 (sin incluir el 10): Sobresaliente
- 10: Matrícula de Honor

5. El NIF (o letra asociada a un DNI) se obtiene de la siguiente manera:
Se divide el número de DNI entre 23 y el resto es codificado por una letra según la siguiente equivalencia:

0: "T", 1: "R", 2: "W", 3: "A", 4: "G", 5: "M", 6: "Y", 7: "F", 8: "P", 9: "D",10: "X", 11: "B", 12: "N", 13: "J", 14: "Z", 15: "S", 16: "Q", 17: "V", 18: "H", 19: "L", 20: "C", 21: "K", 22: "E".

Escribe un programa que pida el DNI y ofrezca como resultado letra asociada.
Ejemplo: para el DNI 56321122 el NIF es 'X'.

6. Realiza un programa que pida un número de 5 cifras y nos diga si es o no capicúa.

Ejemplos de números de 5 cifras capicúas:
21012, 00100, 94549…

7. Asistimos a un parque de atracciones que cobra a sus clientes en función del tiempo de permanencia en el mismo. Concretamente los precios por minuto según el tiempo de permanencia son los siguientes:

Tiempo de permanencia	Precio en céntimos por minuto
Menos de una hora	8 céntimos / minuto
Entre una y dos horas	6 céntimos / minuto
Entre dos de cuatro horas	4 céntimos / minuto
Más de cuatro horas	12 euros

Haz un programa que realice los siguientes pasos:

- Pide la hora de entrada (hora y minutos) y la hora de salida (hora y minutos),
- Comprueba que la hora de salida es superior a la hora de entrada, en caso de no ser así mostrará un mensaje por pantalla indicando el error en las fechas introducidas.
- Realiza el cálculo del importe a cobrar.
- Informa por pantalla del total de minutos de permanencia en el parque y del importe total de pago desglosado en euros y céntimos de euro.

Para la introducción de las horas, utilizaremos el formato 24 h (es decir de 0 a 24 h.).