Objetivo de la práctica:
- Realizar programas con estructuras iterativas.

Conceptos básicos: ESTRUCTURAS ITERATIVAS

Sentencia while

```
while (expresión lógica)
{
    sentencia1;
    sentencia2;
}
```

Sentencia do..while

```
do
{
    sentencia1;
    sentencia2;
}
while (expresión lógica);
```

Sentencia for

```
for(inic; cond; increm)
{
    sentencia1;
    sentencia2;
}
```

COMENTARIOS DE INTERÉS

1. Las sentencias **while**() y **for**() no finalizan con '**;**', mientras que la sentencia **do..while**(); sí.

2. En el cuerpo de un bucle **for**, no debemos variar el valor de las variables que forman parte de la condición de fin, puesto que pueden producirse efectos inesperados.

3. ¿Cuándo utilizar un bucle u otro?
- La sentencia **while** suele utilizarse cuando no se conoce el número de iteraciones del bucle, pudiendo ser éste mayor o igual a 0.
- La sentencia **do..while** suele utilizarse cuando no se conoce el número de iteraciones del bucle, pudiendo ser éste mayor o igual a 1.
- La sentencia **for** suele utilizarse cuando se conoce exactamente el número de iteraciones del bucle.

4. Técnicas de control de los bucles:
- **Bucles controlados por indicadores (banderas):**
  Se utiliza una variable "bandera" de tipo bool, de cuyo valor depende la terminación del bucle.

```
bool continuar;

continuar = true;  // Inicializacion del indicador
while (continuar)
{
    ...
    if (condición para acabar)
        continuar = false;
    ...
}
```

- **Bucles controlados por centinela:**
  En un proceso de introducción de datos, el centinela es el valor cuya lectura como dato, indica la finalización del proceso.

```
suma = 0;
cout << "Introduce números a sumar, 0 para acabar";
cin >> num;
while (num != 0)
{
    suma = suma + num;
    cout << "Introduce números a sumar, 0 para acabar";
    cin >> num;
}
cout << suma;
```

PROBLEMAS

1. Escribe un programa que lea números enteros por teclado hasta que se introduzca un **valor "centinela".** A continuación, debe mostrarse por pantalla el número de valores introducidos (sin contar el valor centinela), el valor máximo y mínimo.

2. Escribe un programa que calcule la **potencia** de un número (entero) elevado a un exponente (entero) utilizando bucles (sin utilizar la función matemática que permite calcular la potencia directamente). Se debe introducir por teclado la base y el exponente de la potencia, y el programa debe mostrar el resultado de la operación.

3. Escribe un programa que calcule la serie:

$$\sum_{i=1}^{N}(i+k)$$ dados los enteros **N, k** leídos por teclado.

4. Escribe un programa que muestre el siguiente **menú** por pantalla:

```
Menú de opciones

1. Introducir teléfono
2. Mostrar información
3. Finalizar

Elige opción:
```

- Si se escoge la opción 1, debe leerse del teclado un número de teléfono compuesto por 9 dígitos (2 para el prefijo y 7 para el teléfono). Por ejemplo, 96-3112233. A continuación, debe aparecer de nuevo el menú de opciones.

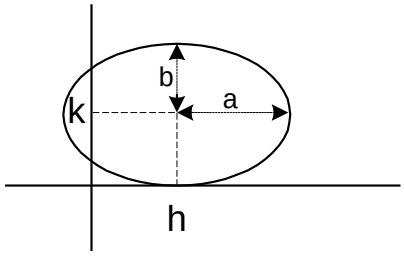
- Si se escoge la opción 2, debe aparecer en pantalla la siguiente información sobre el teléfono previamente almacenado:

```
La comunidad es: Valencia
Y el teléfono: (96) - 3112233
```

  Y a continuación, debe aparecer de nuevo el menú de opciones.
  Considera también teléfonos de Madrid (91) y Barcelona (93).

- Si se escoge la opción 3, el programa debe terminar.

5. Escribe un programa que dados los valores **h, k, a, b** (ver gráfico) introducidos por teclado, sea capaz de dibujar una elipse utilizando los métodos **polinomial** y **trigonométrico** (ver fórmulas).



Centro de la elipse = (h,k)
a = longitud eje mayor
b = longitud eje menor

a) Método **Polinomial :** Obtener los puntos (x,y) de la elipse, variamdo ***x*** entre (h-a) y (h+a)

$$\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} = 1 \xrightarrow{\text{despejando la y ...}} y = b\sqrt{1 - \frac{(x-h)^2}{a^2}} + k$$

b) Método **Trigonométrico:** Obtener los puntos (x,y) de la elipse, variamdo θ entre **0** y 2Π radianes.

$x = a \cdot \cos(\theta) + h$,
$y = b \cdot \text{sen}(\theta) + k$

NOTA:
Desde una aplicación de consola podemos dibujar con caracteres (ASCII) incluyendo una librería (no estándar) llamada **conio.** Para ello insertaremos **`#include <conio.c>`** al principio de nuestra práctica. En este ejercicio, básicamente necesitamos 2 funcionalidades: (a) definir el área o ventana de dibujo y (b) posicionar el puntero (salida estándar) en cualquier lugar de la misma. Para ello utilizaremos las siguientes funciones:

```
void window(int _left, int _top, int _right, int _bottom);
```

Fijamos el tamaño de nuestra ventana, de forma que nos quepa el dibujo (en caracteres).

```
void gotoxy(int coord_x, int coord_y);
```

Posiciona el puntero (la salida estándar) en las coordenadas x, y.