

El **objetivo** de esta práctica es familiarizarse con el uso de algunas estructuras de control y saber decidir cuál se adapta mejor a las especificaciones del enunciado.

La numeración romana se desarrolló en la antigua Roma y se utilizó como sistema de numeración durante todo el imperio romano. Este sistema emplea algunas letras mayúsculas como símbolos para representar ciertos valores. Los números se escriben como una combinación de letras.

El sistema se basa en un sistema aditivo/sustractivo, cada signo representa un valor que se va sumando o restando al anterior.

Signo	Valor
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

Para convertir números romanos a la notación decimal actual se utilizan, entre otras, las siguientes reglas:

- Los números se leen de izquierda a derecha, empezando por los símbolos de mayor valor.
- Si un símbolo es igual o mayor que el siguiente se suma.
(p.e. XVI = 10 + 5 + 1 = 16)
- Si un símbolo es de menor valor que el siguiente se resta.
(p.e. XIV = 10 - 1 + 5 = 14)

Realizar un programa en lenguaje C, que lea desde el teclado un número representado en numeración romana, de forma correcta, y lo muestre en pantalla en formato decimal. El número se leerá como una secuencia de caracteres que termina en punto, se sabe que el número está libre de errores y todos los caracteres están escritos en mayúscula y de forma correcta.

a) Escribir el código a continuación

```
#include <stdio.h>
int main()
{
    int num = 0; // para almacenar el num. en decimal
    int vant=0, vact; /* para almacenar el valor decimal de los caracteres
                        actual y anterior y compararlos */
    char act; //para almacenar el caracter actual

    printf ("Escribe un número representado en numeración romana, terminado en punto:
\n");
    scanf ("%c", &act);

    while (act != '.')
    {
        switch (act)
        {
            case 'I': vact = 1; break;
            case 'V': vact = 5; break;
            case 'X': vact = 10; break;
            case 'L': vact = 50; break;
            case 'C': vact = 100; break;
            case 'D': vact = 500; break;
            case 'M': vact = 1000; break;
        }
        if (vant >= vact)
            num = num + vant;
        else
            num = num - vant;
        vant = vact;
        scanf ("%c", &act);
    }

    num = num + vant;
    printf ("\nEl valor decimal del número introducido es: %d", num);

    return 0;
}
```

- b) Probar el programa con los siguientes datos y copiar la pantalla de ejecución:

DCXXV

MCCLXII

MMCDXLVI

```
Escribe un n·mero representado en numeraci·n romana, terminado en punto:  
DCXXV.
```

```
El valor decimal del n·mero introducido es: 625  
Process returned 0 (0x0)   execution time : 35.469 s  
Press any key to continue.
```

```
Escribe un n·mero representado en numeraci·n romana, terminado en punto:  
MCCLXII.
```

```
El valor decimal del n·mero introducido es: 1262  
Process returned 0 (0x0)   execution time : 15.492 s  
Press any key to continue.
```

```
Escribe un n·mero representado en numeraci·n romana, terminado en punto:  
MMCDXLVI.
```

```
El valor decimal del n·mero introducido es: 2446  
Process returned 0 (0x0)   execution time : 16.591 s  
Press any key to continue.
```

- c) Modificar el programa anterior para que en vez de leer un único número romano, pregunte, si quiere seguir introduciendo números romanos mientras que la respuesta sea afirmativa (s).

```
#include <stdio.h>
int main()
{
    int num; // para almacenar el num. en decimal
    int vant, vact; /* para almacenar el valor decimal de los caracteres actual y
anterior y compararlos */
    char act, resp; //para almacenar el caracter actual y la respuesta
    do
    {
        num = 0; // inicializamos las variables para cada número
        vant = 0;
        printf ("\nEscribe un número en numeración romana, terminado en punto:\n");
        scanf ("%c", &act);
        while (act != '.')
        {
            switch (act)
            {
                case 'I':
                    vact = 1;
                    break;
                case 'V':
                    vact = 5;
                    break;
                case 'X':
                    vact = 10;
                    break;
                case 'L':
                    vact = 50;
                    break;
                case 'C':
                    vact = 100;
                    break;
                case 'D':
                    vact = 500;
                    break;
                case 'M':
                    vact = 1000;
                    break;
            }
            if (vant >= vact)
                num = num + vant;
            else
                num = num - vant;
            vant = vact;
            scanf ("%c", &act);
        }
        num = num + vant;
        printf ("\nEl valor decimal del número introducido es: %d", num);
        fflush(stdin);
        printf ("\n\nQuieres seguir introduciendo numeros (s/n): ");
        scanf ("%c", &resp);
        fflush(stdin);
    }
    while (resp == 's');
    return 0;
}
```

- d) Probar el programa con los datos anteriores y copiar la pantalla de ejecución:

```
Escribe un n-mero representado en numeraci3n romana, terminado en punto:  
DCXXV.  
El valor decimal del n-mero introducido es: 625  
Quieres seguir introduciendo numeros (s/n): s  
Escribe un n-mero representado en numeraci3n romana, terminado en punto:  
MCCLXII.  
El valor decimal del n-mero introducido es: 1262  
Quieres seguir introduciendo numeros (s/n): s  
Escribe un n-mero representado en numeraci3n romana, terminado en punto:  
MMCDXLVI.  
El valor decimal del n-mero introducido es: 2446  
Quieres seguir introduciendo numeros (s/n): n  
Process returned 0 (0x0)   execution time : 58.355 s  
Press any key to continue.
```

OPCIONAL:

- e) Modificar el programa anterior, sabiendo que cada número romano termina con un salto de línea, en lugar de con punto.

```
#include <stdio.h>
int main()
{
    int num; // para almacenar el num. en decimal
    int vant, vact; /* para almacenar el valor decimal de los caracteres actual y
anterior y compararlos */
    char act, resp; //para almacenar el caracter actual y la respuesta

    do
    {
        num = 0;
        vant = 0;
        printf ("\nEscribe un número representado en numeración romana, terminado en
punto:\n ");
        scanf ("%c", &act);

        while (act != '\n')
        {
            switch (act)
            {
                case 'I':
                    vact = 1;
                    break;
                case 'V':
                    vact = 5;
                    break;
                case 'X':
                    vact = 10;
                    break;
                case 'L':
                    vact = 50;
                    break;
                case 'C':
                    vact = 100;
                    break;
                case 'D':
                    vact = 500;
                    break;
                case 'M':
                    vact = 1000;
                    break;
            }
            if (vant >= vact)
                num = num + vant;
            else
                num = num - vant;
            vant = vact;
            scanf ("%c", &act);
        }

        num = num + vant;
        printf ("\nEl valor decimal del número introducido es: %d", num);
        fflush(stdin);
        printf ("\n¿Quieres seguir introduciendo numeros (s/n): ");
        scanf ("%c", &resp);
        fflush(stdin);
    }
    while (resp == 's');
    return 0;
}
```

- f) Modificar el programa anterior, para leer los números romanos de un fichero de texto, donde hay un número romano en cada línea.

```
#include <stdio.h>

int main()
{
    int num; // para almacenar el num. en decimal
    int fin, vant, vact; /* almacenar valor decimal de los caracteres actual y anterior
y compararlos */
    char act; //para almacenar el caracter actual
    FILE *f;
    f=fopen ("d:\\romanos.txt", "r");
    if (f==NULL)
        printf ("No encuentra el fichero");
    else
    {
        fin = fscanf (f,"%c", &act);
        while (fin!=EOF)
        {
            num = 0;
            vant = 0;
            while (act != '\n')
            {
                switch (act)
                {
                    case 'I':
                        vact = 1;
                        break;
                    case 'V':
                        vact = 5;
                        break;
                    case 'X':
                        vact = 10;
                        break;
                    case 'L':
                        vact = 50;
                        break;
                    case 'C':
                        vact = 100;
                        break;
                    case 'D':
                        vact = 500;
                        break;
                    case 'M':
                        vact = 1000;
                        break;
                }
                if (vant >= vact)
                    num = num + vant;
                else
                    num = num - vant;
                vant = vact;
                fin = fscanf (f,"%c", &act);
            }
            num = num + vant;
            printf ("\nEl valor decimal del número introducido es: %d", num);
            fin = fscanf (f,"%c", &act);
        }
        fclose (f);
    }
    return 0;
}
```

// Otra solución

```
#include <stdio.h>

int main()
{
    int num; // para almacenar el num. en decimal
    int vant, vact; /* para almacenar el valor decimal de los caracteres actual y
anterior y compararlos */
    char act; //para almacenar el caracter actual
    FILE *f;

    f=fopen ("romanos.txt", "r");

    if (f==NULL)
        printf ("No encuentra el fichero");
    else
    {
        fscanf (f,"%c", &act);
        while (!feof(f))
        {
            num = 0;
            vant = 0;
            while (act != '\n')
            {
                switch (act)
                {
                    case 'I':
                        vact = 1;
                        break;
                    case 'V':
                        vact = 5;
                        break;
                    case 'X':
                        vact = 10;
                        break;
                    case 'L':
                        vact = 50;
                        break;
                    case 'C':
                        vact = 100;
                        break;
                    case 'D':
                        vact = 500;
                        break;
                    case 'M':
                        vact = 1000;
                        break;
                }
                if (vant >= vact)
                    num = num + vant;
                else
                    num = num - vant;
                vant = vact;
                fscanf (f,"%c", &act);
            }

            num = num + vant;
            printf ("\nEl valor decimal del número introducido es: %d", num);
            fscanf (f,"%c", &act);
        }

    }

    return 0;
}
```