

El **objetivo** de esta práctica es familiarizarse con el uso de las funciones.

Realizar un programa en el lenguaje de programación C que muestre por consola un menú como el que se muestra a continuación:

1. Calcular el factorial de un número.
2. Calcular la n-ésima potencia de un número.
3. Escribir los n primeros números de la serie de fibonacci.
0. Salir.

El programa deberá contener al menos cuatro funciones: una para cada una de las tres opciones, así como otra que presente el menú por pantalla, solicite al usuario la opción que desea utilizar, y devuelva como resultado la opción seleccionada.

El usuario deberá seleccionar una de las opciones y, a continuación, se le solicitará introducir los datos necesarios. Por último, se mostrará por consola el resultado de la opción seleccionada.

- Antes de invocar las funciones, el programa deberá comprobar que los valores solicitados al usuario estén comprendidos entre 0 y 20.
- Para realizar la segunda función no se permite utilizar funciones de la librería *math.h*.
- Las dos primeras funciones deberán devolver un resultado de tipo *float*, y la tercera será de tipo *void*.
- La función que presenta el menú, será de tipo *char*, y deberá validar la respuesta del usuario.

**Solución:**

```
#include <stdio.h>
#define INICIO 0
#define FIN 20

char menu ()
{
    char opcion;
    printf("\n\nOpciones disponibles \n");
    printf(" 1. Calcular el factorial de un numero \n");
    printf(" 2. Calcular la n-esima potencia de un numero\n");
    printf(" 3. Escribir los n primeros numeros de la serie de fibonacci\n");
    printf("\n");
    printf(" 0. Salir\n\n");
    printf("Seleccione la opcion que desee probar: ");
    fflush(stdin);
    do
    {
        scanf("%c", &opcion);
        fflush(stdin);
        if ((opcion < '0') || (opcion > '3'))
            printf ("Opcion incorrecta, escriba un número entre 0 y 3: ");
    }
    while ((opcion < '0') || (opcion > '3'));
    return opcion;
}

int valorCorrecto (int n1, int n2)
{
    int num;
    printf("Debe estar comprendido entre %d y %d ", n1, n2);
    scanf("%d", &num);
    while (num<n1 || num>n2)
    {
        printf("\n");
        printf("El numero introducido debe estar entre %d y %d)\n", n1, n2);
        printf("Introduzca nuevamente un numero entre %d y %d: ", n1, n2);
        scanf("%d", &num);
    }
    return num;
}

float potencia (int base, int exponente)
{
    float resul = 1;
    int i;
    for (i = 1; i <= exponente; i++)
        resul = resul * base;
    return resul;
}

float factorial (int n)
{
    int i;
    float resul = 1;
    for (i = 1; i <=n; i++)
        resul = resul * i;
    return resul;
}
```

```
void fibonacci (int n)
{
    int i, ant=0, act = 1, valor = 1;
    if (n > 0)
        printf (" 0");
    for (i = 1; i < n; i++)
    {
        printf (" %d", valor);
        valor = ant + act;
        ant = act;
        act = valor;
    }
}

int main()
{
    int base, exponente, n, opcion;
    float resul;
    do
    {
        opcion = menu();
        switch (opcion)
        {
            case '1':
                printf("\nIntroduzca un numero: \n");
                n = valorCorrecto(INICIO, FIN);
                resul = factorial (n);
                printf ("El factorial de %d es: %e\n", n, resul);
                break;
            case '2':
                printf("\nIntroduzca la base: \n");
                base = valorCorrecto(INICIO, FIN);
                printf("\nIntroduzca el exponente: \n");
                exponente = valorCorrecto(INICIO, FIN);
                resul = potencia (base, exponente);
                printf ("La potencia %d-esima de %d es: %e\n", exponente, base, resul);
                break;
            case '3':
                printf("\nIntroduzca un numero: \n");
                n = valorCorrecto(INICIO, FIN);
                printf ("los %d primeros numeros de la serie de Fibonacci son:\n", n);
                fibonacci (n);
                break;
            case '0':
                printf ("Adios");
        }
        printf("\n");
    }
    while (opcion != '0');
    return 0;
}
```