

UNIDAD 4: PROGRAMACIÓN ESTRUCTURADA Y MODULAR

Professor: José Ramón Simó Martínez

Ejercicios: Funciones

Nota

Los ejercicios están ordenados de menor a mayor dificultad en cada apartado.

FUNCIONES SIN DATOS DE ENTRADA NI SALIDA

(4.1) En Linux existe el comando “clear” que nos permite “limpiar” la consola cuando tenemos mucho texto que ya no es útil. Su equivalente en Windows es “cls”. Realmente lo que hace este comando es insertar muchas líneas en blanco de manera que parece que desaparecen el texto que había anteriormente.

Vamos a simular este comportamiento escribiendo un programa que borre la pantalla dibujando 25 líneas en blanco. Implementa y utiliza la función:

```
void borrarPantalla()
```

(4.2) Escribe un programa que dibuje un cuadrado formado por 3 filas y 3 columnas de asteriscos. Implementa y utiliza la función:

```
void dibujarCuadrado3x3()
```

FUNCIONES CON DATOS DE ENTRADA

(4.3) Modifica el programa 5.2 de manera que ahora podamos dibujar un rectángulo. Implementa y utiliza la función:

```
void dibujarRectangulo(int ancho, int alto)
```

(4.4) Escribe un programa donde el usuario introduce una letra y un número, y el programa debe mostrar esa letra tantas veces como indique ese número (en la misma línea). Implementa y utiliza la función:

```
void escribirRepetido(char letra, int nrepeticiones)
```

Ayuda: utiliza la función `charAt(0)` para obtener el carácter introducido por el usuario (ej. `sc.next().charAt(0)`);

(4.5) Escribe una nueva versión de la función `dibujarRectangulo` del ejercicio 5.3, que se apoye en la función `escribirRepetido` del ejercicio 5.4

FUNCIONES CON DATOS DE ENTRADA Y SALIDA

(4.6) Escribe un programa que calcule el cubo de un número real (float) introducido por el usuario. El resultado debe ser otro número real. Implementa y utiliza la función:

```
float cubo(float num)
```

(4.7) Escribe un programa indique si un estudiante está aprobado a partir de la nota que ha sacado en la evaluación y la nota mínima para aprobar. El programa pedirá la nota mínima para aprobar, luego irá pidiendo notas de cada estudiante y según el resultado de la función que se utilizará mostrará si está aprobado o no. El programa finaliza cuando se introduce de nota un -1. Implementa y utiliza la función:

```
boolean estaAprobado(int nota, int notaMinima)
```

PARÁMETROS DE ENTRADA AL PROGRAMA (LÍNEA DE COMANDOS)

(4.8) Escribe un programa llamado Calculadora que haga las siguientes operaciones aritméticas: suma(s), resta(r), multiplicación(m) y división (d). Al programa se le pasará por línea de comandos tres parámetros: dos números enteros y la operación a realizar. Por ejemplo, si ejecutamos “Calculadora 1 2 s”, hará la operación $1 + 2$, si ejecutamos “Calculadora 3 5 m” hará la operación $3 * 5$, etc. El programa mostrará el resultado de cada una de estas operaciones.

Ayuda:

FUNCIONES RECURSIVAS VS ITERATIVAS

(4.9) Suma recursiva

Escribe un programa que pida un número natural N (los números positivos) y de como resultado la suma de todos los números naturales hasta N. Implementa y utiliza la función iterativa y recursiva:

```
int suma(int n) // implementa versión iterativa
```

```
int sumaRec(int n) // implementa versión recursiva
```

(4.10) Potencia de un número

Escribe un programa que calcule el valor de elevar un número entero a otro número entero. Por ejemplo, 3 elevado a 4 = $3^4 = 3 \times 3 \times 3 \times 3 = 81$. Luego se mostrará el resultado devuelto por la función. Implementa y utiliza la función iterativa y la recursiva:

```
int potencia(int base, int exponente) // implementa versión iterativa.
```

```
int potenciaRec(int base, int exponente) // implementa versión recursiva.
```