

Práctica 2: Funciones avanzadas

Nota: Ninguna función mostrará nada por pantalla a no ser que se diga lo contrario.

1. Realiza un programa que nos pida número enteros hasta que se introduzca el 0, diciéndonos, para cada número introducido si es primo o no. Hay que recordar que un número es primo si es divisible por si mismo y por 1. El 1 no es primo por convenio. Se debe crear una función que pasándole un número entero devuelva si es primo o no.
2. 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 muestre por pantalla la letra asociada. Para ello se deberá crear una función a la que se le pase el número y devuelva la letra.

Ejemplo: para el DNI 56321122 el NIF es 'X'.

3. Realiza un programa que permita comprobar si una terna de valores enteros (3 valores) se ajusta a la ecuación de Pitágoras: $x^2 + y^2 = z^2$. El programa solicita al usuario los valores x, y, z. Se deberá crear una función a la que se le pase x, y, z y devuelva si son iguales o no.
Por ejemplo: $3^2 + 4^2 = 5^2$.
4. Escribe un programa que imprima las tablas de multiplicar del 1 al 10. Implementa una función que reciba un número entero como parámetro e imprima su tabla de multiplicar.
5. Escribe un programa que muestre un menú con 2 opciones: "1.Circunferencia" y "2.Área".

En ambas se le pedirá al usuario que introduzca un radio y luego se le mostrará el cálculo

oportuno. Implementa las funciones:

```
int menu() // Muestra el menú y devuelve el número elegido
double pideRadio() // Pide que se introduzca el radio y lo devuelve
double circunferencia(double r) // Calcula la circunferencia y la devuelve
double area(double r) // Calcula el área y la devuelve
```

Modifica el programa añadiendo otra opción llamada “Volumen”, permitiendo que el usuario también pueda solicitar el cálculo del volumen. Añade la función:

```
double volumen(double r) // Calcula el volumen y lo devuelve
```

Modifica el programa añadiendo otra opción llamada “Todas” en la que se pida el radio una sola vez y se muestren los tres cálculos posibles (circunferencia, área y volumen).

Modifica el programa anterior de modo que el proceso se repita una y otra vez (mostrar menú -> realizar el cálculo -> volver a mostrar menú). Añade una opción más llamada “Salir” que terminará el programa si es elegida.

Recursividad

6. Dado un String, que se pedirá por pantalla, encuentra la primera letra mayúscula y **muéstrala por pantalla**. Realiza este programa inicialmente de forma recursiva, y posteriormente de forma iterativa.
7. Dado un array de enteros, imprime por pantalla un triángulo con la suma de cada nivel, subiendo hacia arriba. El primer nivel de números será el que se pase por parámetro. A partir de ese primer nivel (nivel base) de abajo del todo, los superiores irán conteniendo un elemento menos a medida que va formándose el triángulo con la suma de los elementos nexos del nivel anterior. Esta tarea se tendrá que realizar mediante recursividad y también mostrará por pantalla el triángulo.

Ejemplo:

Input:

A = {1, 2, 3, 4, 5}

Output:

[48]

[20, 28]

[8, 12, 16]

[3, 5, 7, 9]

[1, 2, 3, 4, 5]

Explicación:

[48]

[20, 28] --> ($20 + 28 = 48$)

[8, 12, 16] --> ($8 + 12 = 20$, $12 + 16 = 28$)

[3, 5, 7, 9] --> ($3 + 5 = 8$, $5 + 7 = 12$, $7 + 9 = 16$)

[1, 2, 3, 4, 5] --> ($1 + 2 = 3$, $2 + 3 = 5$, $3 + 4 = 7$, $4 + 5 = 9$)

8. Realiza un programa que encuentre el mínimo y máximo elemento de un array mediante dos funciones (una para cada uno). Este programa tendrá que obtener la solución mediante recursión y lo mostrará por pantalla.