

## Ejercicios de funciones

### Ejercicio 1

Crea una función `EscribirCentrado`, que reciba como parámetro un texto y lo escriba centrado en pantalla (suponiendo una anchura de 80 columnas; pista: deberás escribir  $40 - \text{longitud}/2$  espacios antes del texto). Además subraya el mensaje utilizando el carácter `=`.

### Ejercicio 2

Crea un programa que pida dos números enteros al usuario y diga si alguno de ellos es múltiplo del otro. Crea una función `EsMultiplo` que reciba los dos números, y devuelve si el primero es múltiplo del segundo.

### Ejercicio 3

Crear una función que calcule la temperatura media de un día a partir de la temperatura máxima y mínima. Crear un programa principal, que utilizando la función anterior, vaya pidiendo la temperatura máxima y mínima de cada día y vaya mostrando la media. El programa pedirá el número de días que se van a introducir.

### Ejercicio 4

Crea una función `ConvertirEspaciado`, que reciba como parámetro un texto y devuelve una cadena con un espacio adicional tras cada letra. Por ejemplo, `"Hola, tú"` devolverá `"H o l a , t ú "`. Crea un programa principal donde se use dicha función.

### Ejercicio 5

Crea una función `calcularMaxMin` que recibe una arreglo con valores numéricos y devuelve el valor máximo y el mínimo. Crea un programa que pida números por teclado y muestre el máximo y el mínimo, utilizando la función anterior.

### Ejercicio 6

Diseñar una función que calcule el área y el perímetro de una circunferencia. Utiliza dicha función en un programa principal que lea el radio de una circunferencia y muestre su área y perímetro.

### Ejercicio 7

Crear una subrutina llamada `Login`, que recibe un nombre de usuario y una contraseña y te devuelve Verdadero si el nombre de usuario es `"usuario1"` y la contraseña es `"asdasd"`. Además

recibe el número de intentos que se ha intentado hacer login y si no se ha podido hacer login incrementa este valor.

Crear un programa principal donde se pida un nombre de usuario y una contraseña y se intente hacer login, solamente tenemos tres oportunidades para intentarlo.

#### Ejercicio 8

Crear una función recursiva que permita calcular el factorial de un número. Realiza un programa principal donde se lea un entero y se muestre el resultado del factorial.

#### Ejercicio 9

Crear una función que calcule el MCD de dos números por el método de Euclides. El método de Euclides es el siguiente:

- Se divide el número mayor entre el menor.
- Si la división es exacta, el divisor es el MCD.
- Si la división no es exacta, dividimos el divisor entre el resto obtenido y se continúa de esta forma hasta obtener una división exacta, siendo el último divisor el MCD.

Crear un programa principal que lea dos números enteros y muestre el MCD.

#### Ejercicio 10

Escribir dos funciones que permitan calcular:

- La cantidad de segundos en un tiempo dado en horas, minutos y segundos.
- La cantidad de horas, minutos y segundos de un tiempo dado en segundos.

Escribe un programa principal con un menú donde se pueda elegir la opción de convertir a segundos, convertir a horas, minutos y segundos o salir del programa.

#### Ejercicio 11

El día juliano correspondiente a una fecha es un número entero que indica los días que han transcurrido desde el 1 de enero del año indicado. Queremos crear un programa principal que al introducir una fecha nos diga el día juliano que corresponde. Para ello podemos hacer las siguientes subrutinas:

- LeerFecha: Nos permite leer por teclado una fecha (día, mes y año).
- DiasDelMes: Recibe un mes y un año y nos dice los días de ese mes en ese año.
- EsBisiesto: Recibe un año y nos dice si es bisiesto.
- Calcular\_Dia\_Juliano: recibe una fecha y nos devuelve el día juliano.

## Ejercicio 12

Vamos a mejorar el ejercicio anterior haciendo una función para validar la fecha. De tal forma que al leer una fecha se asegura que es válida.

## Ejercicio 13

Queremos crear un programa que trabaje con fracciones a/b. Para representar una fracción vamos a utilizar dos enteros: numerador y denominador.

Vamos a crear las siguientes funciones para trabajar con funciones:

- Leer\_fracción: La tarea de esta función es leer por teclado el numerador y el denominador. Cuando leas una fracción debes simplificarla.
- Escribir\_fracción: Esta función escribe en pantalla la fracción. Si el denominador es 1, se muestra sólo el numerador.
- Calcular\_mcd: Esta función recibe dos números y devuelve el máximo común divisor.
- Simplificar\_fracción: Esta función simplifica la fracción, para ello hay que dividir numerador y denominador por el MCD del numerador y denominador.
- Sumar\_fracciones: Función que recibe dos fracciones  $n_1/d_1$  y  $n_2/d_2$ , y calcula la suma de las dos fracciones. La suma de dos fracciones es otra fracción cuyo  $\text{numerador} = n_1*d_2 + d_1*n_2$  y  $\text{denominador} = d_1*d_2$ . Se debe simplificar la fracción resultado.
- Restar\_fracciones: Función que resta dos fracciones:  $\text{numerador} = n_1*d_2 - d_1*n_2$  y  $\text{denominador} = d_1*d_2$ . Se debe simplificar la fracción resultado.
- Multiplicar\_fracciones: Función que recibe dos fracciones y calcula el producto, para ello  $\text{numerador} = n_1*n_2$  y  $\text{denominador} = d_1*d_2$ . Se debe simplificar la fracción resultado.
- Dividir\_fracciones: Función que recibe dos fracciones y calcula el cociente, para ello  $\text{numerador} = n_1*d_2$  y  $\text{denominador} = d_1*n_2$ . Se debe simplificar la fracción resultado.

Crear un programa que utilizando las funciones anteriores muestre el siguiente menú:

1. Sumar dos fracciones: En esta opción se piden dos fracciones y se muestra el resultado.
2. Restar dos fracciones: En esta opción se piden dos fracciones y se muestra la resta.
3. Multiplicar dos fracciones: En esta opción se piden dos fracciones y se muestra el producto.
4. Dividir dos fracciones: En esta opción se piden dos fracciones y se muestra el cociente.
5. Salir: esta opción permite al usuario abandonar el menú y el programa principal.

#### Ejercicio 14

Vamos a crear un programa para trabajar con una pila. Una pila es una estructura de datos que nos permite guardar un conjunto de variables. La característica fundamental es que el último elemento que se añade al conjunto es el primero que se puede sacar.

Para representar una pila vamos a utilizar un arreglo (vector) de cadena de caracteres con tamaño 10, por lo tanto la pila no podrá tener más de 10 elementos.

Vamos a crear varias funciones para trabajar con la pila:

- InicializarPila: Como tenemos un arreglo de 10 elementos de cadenas tenemos que inicializarlo e introducir un carácter (por ejemplo un \* que indique que ese elemento del arreglo no corresponde con un dato de la pila. Esta función inicializa el vector con ese carácter.
- LongitudPila: Función que recibe una pila y devuelve el número de elementos que tiene.
- EstaVacíaPila: Función que recibe una pila y que devuelve si la pila está vacía, no tiene elementos.
- EstaLlenaPila: Función que recibe una pila y que devuelve si la pila está llena.
- AddPila: función que recibe una cadena de caracteres y una pila, y añade la cadena a la pila, si no está llena. si esta llena muestra un mensaje de error.
- SacarDeLaPila: Función que recibe una pila y devuelve el último elemento añadido y lo borra de la pila. Si la pila está vacía muestra un mensaje de error.
- EscribirPila: Función que recibe una pila y muestra en pantalla los elementos de la pila.

Realiza un programa principal que nos permita usar las funciones anterior, que nos muestre un menú, con las siguientes opciones:

1. Añadir elemento a la pila
2. Sacar elemento de la pila
3. Longitud de la pila
4. Mostrar pila
5. Salir

#### Ejercicio 15

Vamos a realizar un programa similar al anterior para trabajar con una cola. Una cola es una estructura de datos que nos permite guardar un conjunto de variables. La característica fundamental es que el primer elemento que se añade al conjunto es el primero que se puede sacar.

En realizada nos sirven todas las funciones del ejercicio anterior menos la función SacarDeLaCola que es la que tienes que modificar.