
Guía de Ejercicios N°6 - Funciones

#	Ejercicio
---	-----------

- | | |
|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | <p>Hacer una función llamada EsPar que determine si un número es par o no. La función debe recibir un número entero por valor y devolver <i>true</i> si es par o <i>false</i> si no lo es. La función no debe mostrar nada por pantalla.</p> <p>Hacer un programa para ingresar un número y, utilizando EsPar, emita luego un cartel indicando si el número ingresado es par o no es par.</p> |
| 2 | <p>Hacer una función llamada CalcularMaximo que determine el máximo entre dos números. La función debe recibir dos números enteros por valor y devolver el valor más grande. Si los números son iguales debe devolver cualquiera de los dos. La función no debe mostrar nada por pantalla.</p> <p>Hacer un programa para ingresar dos números y, utilizando CalcularMaximo, emita luego el número mayor de ambos.</p> |
| 3 | <p>Hacer una función llamada EsPrimo que determine si un número es primo o no. La función debe recibir el número y devolver <i>true</i> si es primo o <i>false</i> si no lo es. La función no debe mostrar nada por pantalla.</p> <p>Hacer un programa para ingresar un número y, utilizando EsPrimo, emita luego un cartel indicando si el número ingresado es primo o no es primo.</p> |
| 4 | <p>Hacer una función llamada EsPrimoSophieGermain que reciba un número entero y determine si el mismo es un número primo de Sophie Germain. Debe devolver verdadero si lo es y falso si no lo es.</p> <p>NOTA: En teoría de números, se dice que un número natural es un número primo de Sophie Germain, si el número n es primo y $2*n+1$ también lo es.</p> |

Ejemplo:

El número 2 es número primo de Sophie Germain porque:

2 es primo

$2*2+1 \rightarrow 5$ es primo.

Hacer un programa para ingresar un número y, utilizando

EsPrimoSophieGermain, emita luego un cartel indicando si el número ingresado es primo Sophie Germain o no lo es.

- 5 Hacer una función llamada **Redondear** que reciba como parámetro un número *float* y devuelva un número entero con el redondeo del mismo.

Por ejemplo:

Si recibe 7.78, debe devolver 8.

Si recibe 7.48, debe devolver 7.

Si recibe 7.5, debe devolver 8.

Hacer un programa para ingresar un número y, utilizando **Redondear**, emita luego un cartel indicando el número redondeado.

- 6 Hacer una función que reciba un número entero por valor llamado *día* y un *string* llamado *nombre* por referencia y le asigne el nombre correspondiente según el número de día. Siendo 0 \rightarrow Domingo y 6 \rightarrow Sábado.

- 7 Escribir una función **CalcularMaximoAbsoluto** que reciba dos números y retorne el máximo absoluto de ambos. Por ejemplo el máximo absoluto de los números -40 y 20 es 40.

Hacer un programa para ingresar dos números y, utilizando

CalcularMaximoAbsoluto, emita luego el número mayor absoluto de ambos.

- 8 Escribir una función que reciba el valor de un año y retorne 1 si el mismo es bisiesto y 0 si es un año no bisiesto. Recordar que son años bisiestos los divisibles por 4, excepto los divisibles por 100, pero dentro de este grupo se incluyen los divisibles por 400. Por ejemplo 1992 fue bisiesto por ser divisible

por 4. El año 1900 no fue bisiesto por ser divisible por 100, aun siendo divisible por 4. Y el año 2000 fue bisiesto por ser divisible por 400, aun siendo divisible por 100.

Hacer un programa para ingresar una lista de 10 valores de años y contar cuantos son bisiestos. Utilizar la función solicitada.

- 9 Escribir una función que reciba un número y retorne 1 si el número recibido es perfecto y 0 si no es perfecto.

Hacer un programa para que, dada una lista de números que finaliza con cero, informe cuántos de ellos eran perfectos. Utilizar la función solicitada.

- 10 Hacer una función que reciba un código de naipes (del 1 al 40) y determine el número y el palo de la baraja española de 40 cartas (sin los 8, 9 y comodines del mazo). La función debe recibir por referencia el número de naipes y el nombre del palo (para ser completados por la función) y por valor el código de naipes.

Tener en cuenta que:

Los códigos de naipes de espada van del 1 al 10, basto del 11 al 20, copa del 21 al 30 y oro del 31 al 40. Por ejemplo, naipes con ID #10 es el 12 de espadas.

- 11 Hacer una función llamada **contarDigitos** que reciba por valor un número entero y determine y devuelva la cantidad de dígitos del número. Por ejemplo, si se recibe el número 840 debe devolver 3.

Hacer un programa que, a partir de un número que ingresa el usuario, informe por pantalla la cantidad de dígitos del número ingresado.

- 12 Escribir una función de nombre **validarFecha** que reciba 3 valores correspondientes al día, mes y año, y devuelva 1 si los valores recibidos corresponden a una fecha correcta o 0 si no es correcta. Por ejemplo, si la función recibe 30, 2, 2000 deberá devolver 0; y si recibe 12, 2, 1990, deberá devolver 1.

- 13 Escribir una función llamada **calcularPotencia** para que, dados dos números enteros, calcule y devuelva la potencia del primero a la del segundo. Tener en cuenta las siguientes posibilidades:

`calcularPotencia(2, 3) → 8`

`calcularPotencia(2, 0) → 1`

`calcularPotencia(2, -3) → 0,125`

- 14 Hacer una función llamada **esNumeroArmstrong** que reciba un número entero y devuelva *true* si el número enviado es un Número Armstrong y *false* si no lo es.

NOTA: Un número N es un número Armstrong si la suma de sus cifras elevadas a la cantidad de cifras del número da como resultado N.

Por ejemplo:

371 tiene 3 cifras.

Luego:

$$3^3 + 7^3 + 1^3 \rightarrow 371$$

$$27 + 343 + 1 \rightarrow 371$$