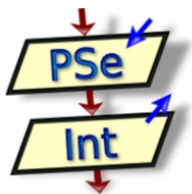


CURSO DE PROGRAMACIÓN FULL STACK

SUBPROGRAMAS CON PSEINT

FUNCIONES



EJERCICIOS DE APRENDIZAJE

Para cada uno de los siguientes ejercicios realizar el análisis del problema e indicar cuáles son los datos de entrada y cuáles son los datos de salida. Escribir luego el programa en PSeInt utilizando funciones.

Subprogramas: Funciones

1. Realizar una función que calcule la suma de dos números. En el algoritmo principal le pediremos al usuario los dos números para pasárselos a la función. Después la función calculará la suma y lo devolverá para imprimirlo en el algoritmo.
2. Realizar una función que valide si un numero es impar o no. Si es impar la función debe devolver un verdadero, si no es impar debe devolver falso. Nota: la función no debe tener mensajes que digan si es par o no, eso debe pasar en el Algoritmo.
3. Crea una función EsMultiplo que reciba los dos números pasados por el usuario, validando que el primer numero múltiplo del segundo y devuelva verdadero si el primer numero es múltiplo del segundo, sino es múltiplo que devuelva falso.
4. Realizar un programa que pida al usuario una frase y una letra a buscar en esa frase. La función debe devolver la cantidad de veces que encontró la letra. Nota: recordar el uso de la función Subcadena().
5. Realizar una función que reciba un numero ingresado por el usuario y averigüe si el numero es primo o no. Un número es primo cuando es divisible sólo por 1 y por sí mismo, por ejemplo: 2, 3, 5, 7, 11, 13, 17, etc. **Nota:** recordar el uso del MOD.

Recursión

6. Escribir una función recursiva que devuelva la suma de los primeros N enteros.

EJERCICIOS DE APRENDIZAJE EXTRA

Estos van a ser ejercicios para reforzar los conocimientos previamente vistos. Estos pueden realizarse cuando hayas terminado la guía y tengas una buena base sobre lo que venimos trabajando. Además, si ya terminaste la guía y te queda tiempo libre en las mesas, puedes continuar con estos ejercicios extra, recordando siempre que no es necesario que los termines para continuar con el tema siguiente. Por último, recordá que la prioridad es ayudar a los compañeros de la mesa y que cuando tengas que ayudar, lo más valioso es que puedas explicar el ejercicio con la intención de que tu compañero lo comprenda, y no sólo mostrarlo. ¡Muchas gracias!

Funciones

1. Realizar una función que calcule y retorne la suma de todos los divisores del número n distintos de n . El valor de n debe ser ingresado por el usuario.
2. Diseñar una función que reciba un número en forma de cadena y lo devuelva como número entero. El programa podrá recibir números de hasta 3 dígitos. Nota: no poner números con decimales
3. Crear una función llamada "Login", que recibe un nombre de usuario y una contraseña y que devuelve Verdadero si el nombre de usuario es "usuario1" y si la contraseña es "asdasd". Además la función calcula el número de intentos que se ha usado para loguearse, tenemos solo 3 intentos, si nos quedamos sin intentos la función devolverá Falso.
4. Los empleados de una fábrica trabajan en dos turnos: Diurno y Nocturno. Se desea calcular el jornal diario de acuerdo con las siguientes reglas:
 - a) La tarifa de las horas diurnas es de \$ 90
 - b) La tarifa de las horas nocturnas es de \$ 125
 - c) En caso de ser feriado, la tarifa se incrementa en un 10% si el turno es diurno y en un 15% si el turno es nocturno.

El programa debe solicitar la siguiente información al usuario: el nombre del trabajador, el día de la semana, el turno (diurno o nocturno) y la cantidad de horas trabajadas. Además, debemos preguntarle al usuario si el día de la semana (lunes, martes, miércoles, etc.) era festivo o no, para poder calcular el jornal diario. Utilice una función para realizar el cálculo.

5. Realizar una función que calcule la suma de los dígitos de un número.

Ejemplo: $25 = 2 + 5 = 7$

Nota: Para obtener el último número de un dígito de 2 cifras o más debemos pensar en el resto de una división entre 10. Recordar el uso de la función Mod y Trunc.

6. Realizar una función que reciba un numero ingresado por el usuario y averigüe si el número tiene todos sus dígitos impares (ejemplo: 333, 55, etc.). Para esto vamos a tener que separar el numero en partes (si es un numero de más de un dígito) y ver si cada numero es par o impar. **Nota:** recordar el uso de la función Mod y Trunc(). No podemos pasar el numero a cadena para realizar el ejercicio.
7. Realizar una función que permita obtener el término n de la sucesión de Fibonacci. La sucesión de Fibonacci es la sucesión de los siguientes números:

1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Donde cada uno de los números se calcula sumando los dos anteriores a él. Por ejemplo:

La sucesión del número 2 se calcula sumando (1+1)

Análogamente, la sucesión del número 3 es (1+2),

Y la del 5 es (2+3),

Y así sucesivamente...

La sucesión de Fibonacci se puede formalizar de acuerdo a la siguiente fórmula:

Fibonacci (n) = Fibonacci (n-1) + Fibonacci (n-2) para todo n > 1

Fibonacci (n) = 1 para todo n <= 1

Por lo tanto, si queremos calcular el término "n" debemos escribir una función que reciba como argumento el valor de "n" y que calcule la serie hasta llegar a ese valor.

Para conocer más acerca de la serie de Fibonacci consultar el siguiente link:
<https://quantdare.com/numeros-de-fibonacci/>

8. Realizar una función que reciba un numero ingresado por el usuario y averigüe si el numero es capicúa o no (Por ejemplo: 12321). **Nota:** recordar el uso del MOD y el Trunc. No podemos transformar el numero a cadena para realizar el ejercicio.
9. El número de combinaciones de m elementos tomados de n es:

$$\binom{m}{n} = \frac{m!}{n!(m-n)!}$$

Diseñar una función que permita calcular el número combinatorio de $\binom{m}{n}$

Nota: n debe ser mayor a 0 y menor que m.

Recursión

10. Crear un programa que calcule la suma de los enteros positivos pares desde N hasta 2. Chequear que si N es impar se muestre un mensaje de error.
11. Implemente de forma recursiva una función que le dé la vuelta a una cadena de caracteres. **NOTA:** Si la cadena es un palíndromo, la cadena y su inversa coincidirán.

12. Realice nuevamente un programa que calcule la función de Fibonacci pero esta vez de manera recursiva.
13. Escribir un programa que calcule el máximo común divisor (MCD) de dos enteros positivos. Si $M \geq N$ una función recursiva para MCD es:

$MCD = M$, si $N = 0$
 $MCD = MCD(N, M \bmod N)$, si $N \neq 0$

El programa le debe permitir al usuario ingresar los valores para M y N. Una función recursiva es entonces llamada para calcular el MCD. El programa debe imprimir el valor para el MCD.