

función que recibe un número y devuelve el factorial de dicho número acompañado de un mensaje de bienvenida.

1. (Nivel 1) Desarrolle una función que calcule el factorial de un número.

Input: Output: 6
---------------------

2. (Nivel 1) Elabore una función recursiva para calcular el factorial de un número.

función que recibe un número y devuelve el factorial de dicho número acompañado de un mensaje de bienvenida.

Input: Output: 1.5
-----------------------

2014 CERCAL



ingre  
prese de

Thomas

De acordo com a fórmula de cálculo da eficiência, a eficiência média dos

Irrese un i

1 1 \ T



• ENDORSE:

Lesde Z haben

compar,

para n=4 au

para

de

Topic

Index

Index

Index

400

13. (1 punto) Escribir un programa recursivo que pida un número  $n$  y devuelva los números pares entre 0 y  $n$ , separados por un espacio.

Ejemplo:

Se pide un número: 10

Se devuelven: 0 2 4 6 8 10

Indicaciones: el número es: 5

Los números pares son: 1 2 3 4 5

14. (1 punto) Diseñar un algoritmo que implemente la función recursiva que permita calcular cuántos pares hay en un número  $n$ . Considere que el cero es par.

Algunos ejemplos de ejecución de este programa serían:

Input: 10

Output: 5

Input: 5

Output: 3

Input: 0

Output: 1

15. (1 punto) Diseñar un programa que calcule el valor de la función recursiva:

$$H(n, d) = 1 + 1/d + H(n, d+1) + 2/d + 3/d + \dots + n/d$$

El programa debe pedir al usuario que se ingrese un número  $n$ , y un número  $d$ .

El programa debe calcular el valor de  $H(n, d)$  usando una función recursiva,

imprimir el resultado de  $H(n, d)$

Algunos ejemplos de ejecución de este programa serían:

Input: n: 3

Input: d: 2

Output: 3

Output: n: 5

Input: 0

Indicaciones: el programa debe seguir la siguiente estructura:

(Código)

- Debe calcular el valor de  $H(n, p)$  usando una función recursiva,
- Debe calcular el módulo de  $H(n, p)$

Algunos ejemplos de diálogo de este programa:

```
Input n: 5
Input p: 2
Output: 8
```

```
Input n:
Input p:
Output:
```

17. (20 puntos) Diseñar un programa que permita verificar si un número es divisible por 3. Para ello, se utilizará el operador módulo. El criterio de divisibilidad por 3 es el siguiente: un número es divisible por 3 si la suma de sus dígitos es divisible por 3 o múltiplo de 3. Es decir, si un número es divisible por 3, entonces la suma de sus dígitos también lo es. Escribir un programa que permita verificar si un número es divisible por 3. El programa debe solicitar al usuario un número, calcular la suma de sus dígitos y verificar si es divisible por 3. El programa debe imprimir el resultado de la verificación.

Describe un programa que realice lo siguiente:

- Solicitar al usuario ingresar un número.
- Obtener la suma de los dígitos del número.
- Verificar si la suma de los dígitos es divisible por 3 (utilizar el operador módulo %).
- Imprimir "El número SI es divisible por 3" cuando el número sea divisible por 3 y "El número NO es divisible por 3" en caso contrario.

Para obtener la suma de los dígitos, implementar la siguiente **función recursiva**:

- **suma\_digito**: recibe como parámetro un número y retorna la suma de los dígitos.

Algunos ejemplos de diálogo de este programa serían:

```
Ingrese un número: 14587629
La suma de los dígitos es: 42
El número SI es divisible por 3
```

```
Ingrese un número: 783475
```

```
El número NO es divisible por 3
```

18. (Nivel 3) Un número **palindromo** es un número que se lee de igual manera de izquierda a derecha, que de derecha a izquierda. La palabra "recrea" es un palindromo.

El número 12321 es un palindromo. El número 1234567890 no lo es.

- La diferencia entre el número y su inverso es igual al cero.

El número 12321 es un palindromo. El número 1234567890 no lo es.

El número 12321 es un palindromo. El número 1234567890 no lo es.

El número 12321 es un palindromo. El número 1234567890 no lo es.

El número 12321 es un palindromo. El número 1234567890 no lo es.

El número 12321 es un palindromo. El número 1234567890 no lo es.

El número 12321 es un palindromo. El número 1234567890 no lo es.

El número 12321 es un palindromo. El número 1234567890 no lo es.

19. (Nivel 3) Escribe un programa en Python que contenga una función **recursiva** que reciba una lista de números y retorne la sumatoria de sus elementos.

Por ejemplo:

```
Ingrese la lista: [3, 5, 0, 1]
La suma total es 9
```

```
Ingrese la lista: [1.2, 1.0, 1.0, 1.0, 1.0]
La suma total es 5.2
```

20. (Nivel 3) Diseñe e implemente una función recursiva para hallar el mayor elemento de una lista.

Plantilla del código:

```
def mayorListaRec(d):  
    #  
    # implemente funcion recursiva aqui  
    #  
  
lista = [5,4,1,8,6,2,3]  
print("Mayor:", mayorListaRec(lista))
```

Mayor: 8

21. (Nivel 3) Escribir un función recursiva que calcule el capital obtenido en un determinado mes sabiendo que el Banco brinda una tasa de interés del 12% por año (0.12 para la fórmula).

- La fórmula para conocer cuánto es el monto futuro es:  $C_f = C_i * (1 + T_a)^{(n/12)}$ , siendo  $T_a$  la tasa anual,  $C_f$  la cuota final,  $C_i$  la cuota inicial y  $n$  el número de meses.
- **Nota:** Truncar el resultado en tres decimales.

Un ejemplo de diálogo de este programa sería:

```
Ingrese el Capital Inicial: 10000  
Ingrese el mes en el que quiere saber su capital: 10  
El Capital en el mes 10 es: 11046.221
```

22. (Nivel 3) Diseñe e implemente una función recursiva que permita hallar el cociente (únicamente la parte entera) de dividir dos números por medio de restas. El encabezado de la función debe ser: `divide_restando(dividendo, divisor)`.

- En caso el segundo número sea 0, deberá mostrar un mensaje explicando que no se puede dividir entre cero.
- El programa no permitirá números mayores a 10500.

```
Ingrese el primer número: 200  
Ingrese el segundo número: 7  
Resultado: 28
```

23. (Nivel 4) Generar una lista de números enteros usando comprensión y que se encuentre el menor y el mayor elemento ingresado a la lista utilizando recursividad.

Tener en cuenta lo siguiente:

- Debe crear la función recursiva mayor.
- Debe crear la función recursiva menor.
- Pedir al usuario el número de elementos a ingresar.



- Pedir al usuario que ingrese cada elemento correlativamente.

Algunos ejemplos de diálogo de este programa serían:

```
Ingrese el número de elementos de la lista: 5
Ingrese elemento 1: 10
Ingrese elemento 2: 11
Ingrese elemento 3: 78
Ingrese elemento 4: 16
Ingrese elemento 5: 96
El menor elemento de la lista es 10
El mayor elemento de la lista es 96
```

```
Ingrese el número de elementos de la lista: 3
Ingrese elemento 1: 11523
Ingrese elemento 2: 10
Ingrese elemento 3: 14584
El menor elemento de la lista es 10
El mayor elemento de la lista es 14584
```

```
Ingrese el número de elementos de la lista: 10
Ingrese elemento 1: 1
Ingrese elemento 2: 9
Ingrese elemento 3: 8
Ingrese elemento 4: 7
Ingrese elemento 5: 5
Ingrese elemento 6: 6
Ingrese elemento 7: 15
Ingrese elemento 8: 4
Ingrese elemento 9: 12
Ingrese elemento 10: 14
El menor elemento de la lista es 1
El mayor elemento de la lista es 15
```

24. (Nivel 5) Implementar un algoritmo que permita calcular el Máximo Común Divisor) de dos números.

- Implementar un algoritmo que calcule el MCD.
- El usuario ingresa dos números enteros por teclado.
- Imprimir el resultado del MCD.

Algunos ejemplos de diálogo de este programa serían:

```
Input:
Por favor ingrese número A: 12
Por favor ingrese número B: 3
```

```
Output:  
El máximo común divisor es: 3
```

```
Input:  
Por favor ingrese número A: 169  
Por favor ingrese número B: 256  
Output:  
El máximo común divisor es: 1
```