

_9generalas/basicfuncmath.py

```
1 # Sumador de lista
2 def sumar_lista(lista):
3     suma = 0
4     for numero in lista:
5         suma += numero
6     return suma
7
8 print(sumar_lista([1, 2, 3, 4, 5])) # Salida: 15
9
10 # Contador de números pares e impares
11 def evens_and_odds(n):
12     evens = sum(1 for i in range(n + 1) if i % 2 == 0)
13     odds = n - evens
14     return f"The number of odds are {odds}.\nThe number of evens are {evens}."
15
16 resultado = evens_and_odds(14)
17 print(resultado) # Salida: The number of odds are 7. The number of evens are 8.
18
19 # Generador de números pares
20 def pair_generator(limit):
21     my_list = []
22     for number in range(1, limit):
23         my_list.append(number * 2)
24     return my_list
25
26 print(pair_generator(10)) # Salida: [2, 4, 6, 8, 10, 12, 14, 16, 18]
27
28 # Factorial usando math
29 import math
30 print(math.factorial(5)) # Salida: 120
31
32 # Factorial usando recursión
33 def factorial(n):
34     if isinstance(n, int) and n >= 0:
35         if n == 0 or n == 1:
36             return 1
37         else:
38             return n * factorial(n - 1)
39     else:
40         return "Input must be a non-negative integer"
41
42 print(factorial(5)) # Salida: 120
43
44 # Factorial usando bucle
45 def factorial_iterativo(n):
46     if isinstance(n, int) and n >= 0:
47         resultado = 1
48         for i in range(1, n + 1):
49             resultado *= i
50         return resultado
51     else:
```

```
52         return "Input must be a non-negative integer"
53
54 print(factorial_iterativo(5)) # Salida: 120
55
56 # Potencia usando recursión
57 def potencia(base, exponente):
58     if exponente == 0:
59         return 1
60     else:
61         return base * potencia(base, exponente - 1)
62
63 print(potencia(2, 3)) # Salida: 8
64
65 # Suma de elementos en una lista usando recursión
66 def suma_lista(lista):
67     if len(lista) == 0:
68         return 0
69     else:
70         return lista[0] + suma_lista(lista[1:])
71
72 print(suma_lista([1, 2, 3, 4, 5])) # Salida: 15
73
74 # Número triangular usando recursión
75 def numero_triangular(n):
76     if n == 1:
77         return 1
78     else:
79         return n + numero_triangular(n - 1)
80
81 print(numero_triangular(4)) # Salida: 10
82
83 # Imprimir números pares en orden descendente usando recursión
84 def num_pares(num):
85     if num < 2:
86         return
87     print(num)
88     num_pares(num - 2)
89
90 num_pares(14)
91 # Salida: 14 12 10 8 6 4 2
92
93 # Imprimir números pares en orden descendente usando bucle
94 def numeros_pares(num):
95     for i in range(num, 0, -2):
96         print(i)
97
98 numeros_pares(14)
99 # Salida: 14 12 10 8 6 4 2
100
101 # Suma de dígitos de un número usando recursión
102 def suma_digitos(n):
103     if n == 0:
104         return 0
105     else:
```

```
106         return n % 10 + suma_digitos(n // 10)
107
108 print(suma_digitos(1234)) # Salida: 10
109
110 # Comprobación de número primo
111 def es_primo(n):
112     if n < 2:
113         return False
114     for i in range(2, int(n**0.5) + 1):
115         if n % i == 0:
116             return False
117     return True
118
119 print(es_primo(11)) # Salida: True
120 print(es_primo(4)) # Salida: False
121
122 # Generador de números primos hasta un límite
123 def generar_primos(limite):
124     primos = []
125     for num in range(2, limite + 1):
126         if es_primo(num):
127             primos.append(num)
128     return primos
129
130 print(generar_primos(10)) # Salida: [2, 3, 5, 7]
131
132 # Función principal para ejecutar todas las funciones de prueba
133 def main():
134     print("Sumador de lista:", sumar_lista([1, 2, 3, 4, 5])) # Salida: 15
135     print("Contador de números pares e impares:", evens_and_odds(14)) # Salida: The number
of odds are 7. The number of evens are 8.
136     print("Generador de números pares:", pair_generator(10)) # Salida: [2, 4, 6, 8, 10, 12,
14, 16, 18]
137     print("Factorial usando math:", math.factorial(5)) # Salida: 120
138     print("Factorial usando recursión:", factorial(5)) # Salida: 120
139     print("Factorial usando bucle:", factorial_iterativo(5)) # Salida: 120
140     print("Potencia usando recursión:", potencia(2, 3)) # Salida: 8
141     print("Suma de elementos en una lista usando recursión:", suma_lista([1, 2, 3, 4, 5]))
# Salida: 15
142     print("Número triangular usando recursión:", numero_triangular(4)) # Salida: 10
143     num_pares(14) # Salida: 14 12 10 8 6 4 2
144     numeros_pares(14) # Salida: 14 12 10 8 6 4 2
145     print("Suma de dígitos de un número usando recursión:", suma_digitos(1234)) # Salida:
10
146     print("Comprobación de número primo (11):", es_primo(11)) # Salida: True
147     print("Comprobación de número primo (4):", es_primo(4)) # Salida: False
148     print("Generador de números primos hasta un límite:", generar_primos(10)) # Salida: [2,
3, 5, 7]
149
150 if __name__ == "__main__":
151     main()
152
```