(https://www.darshan.ac.in/)

Python Programming - 2101CS405

Lab - 7

Name:Parmar Himanshu

Roll No.:341 B3

Enrollment No.:22010101132

Functions

01) WAP to count simple interest using function.

In [2]:
```
def intrest(p,r,t):
    return (p*r*t)/100
p = float(input("Enter Principal:"))
r = float(input("Enter Rate:"))
t = float(input("Enter Time:"))
ans = intrest(p,r,t)
print("Simple Intrest:",ans)
```

```
Enter Principal:50000
Enter Rate:2
Enter Time:3
Simple Intrest: 3000.0
```

02) WAP that defines a function to add first n numbers.

In [3]:
```
def addNNumber(n):
    sum = 0
    for i in range(1,n+1):
        sum+=i
    return sum
n = int(input("Enter Number:"))
ans = addNNumber(n)
print(f"Sum {n} Number:",ans)
```

```
Enter Number:6
Sum 6 Number: 21
```

03) WAP to find maximum number from given two numbers using function.

In [9]:
```
def maxNumb(num1,num2):
    return num1 if num1>num2 else num2
num1 = int(input("Enter 1st Number:"))
num2 = int(input("Enter 2nd Number:"))
ans = maxNumb(num1,num2)
print("Maximum Number:",ans)
```

```
Enter 1st Number:50
Enter 2nd Number:1
Maximum Number: 50
```

04) WAP that defines a function which returns 1 if the number is prime otherwise return 0.

In [12]:
```
def isPrime(num):
    for i in range(2,int(num**(0.5))+1):
        if num%i==0:
            return 0
    else:
        return 1
num = int(input("Enter Number:"))
ans = isPrime(num)
print(ans)
```

```
Enter Number:9973
1
```

05) Write a function called primes that takes an integer value as an argument and returns a list of all prime numbers up to that number.

In [6]:
```
def isPrime(num):
    for i in range(2,int(num**(0.5))+1):
        if num%i==0:
            return False
    else:
        return True
list = []
def primes(n):
    if isPrime(n):
        list.append(n)

num = int(input("Enter Number:"))
for i in range(2,num+1):
    primes(i)
print(list)
print("Total Primes:",len(list))
```

```
Enter Number:1000
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 7
1, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 1
51, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229,
233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313,
317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409,
419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499,
503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601,
607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691,
701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797, 809,
811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907,
911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997]
Total Primes: 168
```

06) WAP to generate Fibonacci series of N given number using function name fibbo. (e.g. 0 1 1 2 3 5 8...)

In [29]:
```
def fibbo(n,list,i):
    if i>n:
        return
    list.append(list[i-1]+list[i-2])
    fibbo(n,list,i+1)

num = int(input("Enter Number:"))
list = [0,1]
fibbo(num,list,2)
print(list)
```

```
Enter Number:50
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2
584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 31781
1, 514229, 832040, 1346269, 2178309, 3524578, 5702887, 9227465, 14930352,
24157817, 39088169, 63245986, 102334155, 165580141, 267914296, 433494437,
701408733, 1134903170, 1836311903, 2971215073, 4807526976, 7778742049, 125
86269025]
```

07) WAP to find the factorial of a given number using recursion.

In [1]:
```
def factorial(n):
    if n==0:
        return 1
    return n * factorial(n-1)
num = int(input("Enter Number:"))
ans = factorial(num)
print(ans)
```

```
Enter Number:5
120
```

08) WAP to implement simple calculator using lamda function.

In [15]:
```
num1 = int(input("Enter 1st Number:"))
num2 = int(input("Enter 2nd Number:"))
sum = lambda num1,num2 : num1+num2
sub = lambda num1,num2 : num1-num2
mul = lambda num1,num2 : num1*num2
div = lambda num1,num2 : num1/num2
print(sum(num1,num2))
print(sub(num1,num2))
print(mul(num1,num2))
print(div(num1,num2))
```

```
Enter 1st Number:10
Enter 2nd Number:20
30
-10
200
0.5
```

09)Write a Python program that accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically

Sample Items : green-red-yellow-black-white

Expected Result : black-green-red-white-yellow

In [24]:
```
def sortword(list):
    temp = list.split("-")
    ans = sorted(temp)
    print("-".join(ans))
list = input("Enter hyphen-separated sequence of words:")
sortword(list)
```

```
Enter hyphen-separated sequence of words:green-red-yellow-black-white
black-green-red-white-yellow
```

10) Write a python program to implement all function arguments type

Positional arguments
Default argument
Keyword arguments (named arguments)
Arbitrary arguments (variable-length arguments args and kwargs)

In [26]:
```
# Positional arguments
def add(a,b):
    return a+b
print(add(10,20))
```

```
30
```

In [27]:
```
# Default argument
def add(a,b=10):
    return a+b
print(add(10,20))
print(add(10))
```

```
30
20
```

In [28]:
```
# Keyword arguments (named arguments)
def add(a,b):
    print("a",a)
    print("b",b)
print(add(b=10,a=20))
```

```
a 20
b 10
None
```

In [32]:
```
# Arbitrary arguments (variable-length arguments args and kwargs)
def add(*a):
    print(a)
add(10,20,30,40)

def add2(**a):
    print(a['A'])
    print(a['B'])
    print(a)
add2(A=10,B=30,C=40)
```

```
(10, 20, 30, 40)
10
30
{'A': 10, 'B': 30, 'C': 40}
```

01) WAP to calculate power of a number using recursion.

In [1]:
```
def power(base,exponnet):
    if exponnet==0:
        return 1
    return base * power(base,exponnet-1)
base = int(input("Enter Base:"))
exponnet = int(input("Enter exponnet:"))
print(power(base,exponnet))
```

```
Enter Base:2
Enter exponnet:3
8
```

02) WAP to count digits of a number using recursion.

In [35]:
```
def countdigit(n,count):
    if n==0:
        return count
    return countdigit(n//10,count+1)
num = int(input("Enter Number:"))
ans = countdigit(num,0)
print(ans)
```

```
Enter Number:1234567890
10
```

03) WAP to reverse an integer number using recursion.

In [46]:
```
def reverse(num,ans):
    if num<=0:
        return ans
    ans=(ans*10+(num%10))
    return reverse(num//10,ans)
num = int(input("Enter Number:"))
ans = reverse(num,0)
print(ans)
```

```
Enter Number:152
251
```

04) WAP to convert decimal number into binary using recursion.

In [4]:
```
def decimal_to_binary(decimal):
    if decimal == 0:
        return ""
    else:
        binary = decimal_to_binary(decimal // 2)
        return binary + str(decimal % 2)

decimal_number = int(input("Enter a decimal number: "))

binary_representation = decimal_to_binary(decimal_number)
print(f"binary: {binary_representation}")
```

```
Enter a decimal number: 7
binary: 111
```

In [ ]:
```
```