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

# Python Programming - 2101CS405

# Lab - 7

**Name : Viral chauhan**

**Enrollment : 22010101027**

**Roll No. : 184  Batch : A4**

## Functions

## 01) WAP to count simple interest using function.

In [1]:
```
def simpleInterest(amount,rateOfInterest,time):
    return ((amount*rateOfInterest*time)/100)

p = float(input("Enter principle amount : "))
r = float(input("Enter rate of interest : "))
n = float(input("Enter time in year : "))
print("Your simple interest is ",simpleInterest(p,r,n))
```

```
Enter principle amount : 100
Enter rate of interest : 2
Enter time in year : 1
Your simple interest is  2.0
```

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

In [2]:
```
def sumOfFirstNNumbers(n):
    return ((n/2)*(n+1))
n = int(input("Enter a number : "))
print(f"sum of first {n} number is {sumOfFirstNNumbers(n)}")
```

```
Enter a number : 5
sum of first 5 number is 15.0
```

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

In [4]:
```
def maximum(n1,n2):
    return n1 if n1>n2 else n2
n1 = int(input("Enter a first number : "))
n2 = int(input("Enter a second number : "))

print(f"maximum of two numbers  {n1} and {n2} is {maximum(n1,n2)}")
```

```
Enter a first number : 5
Enter a second number : 4
maximum of two numbers  5 and 4 is 5
```

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

In [13]:
```
def isPrime(n):
    for i in range(2,int(n**0.5 + 1 )):
        if(n%i == 0):
            return 0

    return 1
n = int(input("Enter a number : "))
print(isPrime(n))
```

```
Enter a number : 4
0
```

## 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 [14]:
```
n = int(input("Enter a number : "))
primeList = []
for i in range(1,n):
    if(isPrime(i)):
        primeList.append(i)
print(primeList)
```

```
Enter a number : 100
[1, 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]
```

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

In [19]:
```
def fibbo(a1,a2,n):
    fibboSeries = []
    fibboSeries.append(a1)
    fibboSeries.append(a2)
    for i in range(a1, n-2 if a1==0 else n-1):
        fibboSeries.append(a1+a2)
        a2 = a1 + a2
        a1 = a2 - a1
    return fibboSeries
n = int(input("Enter a number : "))
print(fibbo(0,1,n))
```

```
Enter a number : 5
[0, 1, 1, 2, 3]
```

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

In [21]:
```
def factorial(n):
    if(n<=1):
        return 1
    return (n* factorial(n-1))
n = int(input("Enter a number : "))
print(factorial(n))
```

```
Enter a number : 5
120
```

## 08) WAP to implement simple calculator using lamda function.

In [27]:
```
def myCal(n1,n2,ch):
    match ch :
        case "+":
            return (lambda n1,n2: n1+n2)(n1,n2)
        case "-":
            return (lambda n1,n2: n1-n2)(n1,n2)
        case "*":
            return (lambda n1,n2: n1*n2)(n1,n2)
        case "/":
            return (lambda n1,n2: n1/n2)(n1,n2)
        case "%":
            return (lambda n1,n2: n1%n2)(n1,n2)
    return "invalid operater"
n1 = int(input("Enter a first number : "))
n2 = int(input("Enter a second number : "))

ch = input("Enter operation you want to perform (+,-,*,/,% for modulo)")
print(myCal(n1,n2,ch))
```

```
Enter a first number : 5
Enter a second number : 4
Enter operation you want to perform (+,-,*,/,% for modulo)*
20
```

## 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 [29]:
```
s = "green-red-yellow-black-white"
mystr = s.split("-")
mystr.sort()
s = ("-").join(mystr)
print(s)
```

```
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 [31]:
```
def myFun(n1,n2=1):
    print(n1*n2)
def myFun1(*n):
    print(n)
```

In [33]:
```
myFun(n2=5,n1=4) # keyword
myFun(4) # defual
myFun(11,2) # positional
myFun1(4,5,6,7,8,9,0) # arbitary
```

```
20
4
22
(4, 5, 6, 7, 8, 9, 0)
```

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

In [34]:
```
def myRecPorwer(base,power):
    if(power <=0):
        return 1
    else:
        return base * myRecPorwer(base,power-1)

b = int(input("Enter a base number : "))
p = int(input("Enter a power number : "))
print(myRecPorwer(b,p))
```

```
Enter a base number : 2
Enter a power number : 3
8
```

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

In [ ]:

## 03) WAP to reverse an integer number using recursion.

In [ ]:

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

In [ ]: