Darshan UNIVERSITY

योगः कर्मसु कौशलम्

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

Python Programming - 2101CS405

Lab - 7

Name : Krish GOHEL

ROLLNO : 108

ENROLLMENT NO : 22010101060

Functions

01) WAP to count simple interest using function.

In [2]:

```
def interest(p,n,r):
    return p*n*r/100

ans = interest(10000,3,5)
print(ans)
```

```
1500.0
```

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

In [7]:

```
A = int(input('A : '))

def addn(n):
    temp = 0
    for i in range(n+1):
         temp += i
    return temp

ans = addn(100)
print(ans)
```

```
A : 100
5050
```

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

In [8]:

```
A = int(input('A:'))
B = int(input('B:'))

def maxN(A,B):
    if A>B:
        return A
    else:
        return B

ans = maxN(A,B)
print(ans)
```

```
452
652
652
```

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

In [12]:

```
n = int(input('n:'))

def prime(n):
    flag = 0
    for i in range(2,int(n/2+1)):
        if(n%i == 0):
            flag = 1
    if(flag == 1):
        return 'given number is not prime'
    else:
        return 'given number is prime.'

ans = prime(n)
print(ans)
```

```
n:5
given number is prime.
```

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 [20]:

```
n = int(input('n:'))

def primes(n):
    flag = 0
    a = []
    for i in range(1,n):
        for j in range(2,i//2+1):
            if(i%j == 0):
                break
        else:
            a.append(i)
    return a

ans = primes(n)

print(ans)
```

```
n:10
[1, 2, 3, 5, 7]
```

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

In [39]:

```
n = int(input('n:'))

def fibbo(n):
    if(n <= 1):
        return n
    else:
        return (fibbo(n-1)+fibbo(n-2))


for i in range(n):
    print(fibbo(i))
```

```
n:10
0
1
1
2
3
5
8
13
21
34
```

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

In [26]:

```
n = int(input('n:'))

def fact(n):
    if(n == 0):
        return 1
    elif(n>-1):
        return n*fact(n-1)

ans = fact(n)
print(ans)
```

```
n:5
120
```

In [ ]:

```
```

08) WAP to implement simple calculator using lamda function.

In [52]:
```
ch = (input('choice : '))
n1 = int(input('n1 :'))
n2 = int(input('n2 :'))

print((lambda a1,a2 : n1+n2 if ch == "+" else n1-n2 if ch == '-' else n1*n2 if
```

```
choice : +
n1 :10
n2 :20
30
```

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 [6]:
```
A = input("A : ")
B = []
B = A.split("-")
C = ""

B.sort()
print(B)

for i in B:
    C += i+"-"

print(C)
```

```
A : green-red-yellow-black-white
['black', 'green', 'red', 'white', 'yellow']
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 [36]:

```
# Positional argument
def demo(x,y):
    print(x,',',y)

demo(10,20)
```

```
10 , 20
```

In [41]:

```
# default argument
def demo(x,y=20):
    print(x,',',y)

demo(10)
```

```
10 , 20
```

In [45]:

```
# keyword argumnet
def demo(n1,n2):
    print(n1,',',n2)

demo(10,20)
demo(n2=10,n1=20)
```

```
10 , 20
20 , 10
```

In [47]:

```
# arbitury argument
def demo(n1,*n2):
    print(n1,',',n2)

demo(10,20,30,40,50)
```

```
10 , (20, 30, 40, 50)
```

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

In [56]:

```
b = float(input('b:'))
p = int(input('p:'))

def power(b,p):
    if(p <=0):
        return 1
    else:
        return b*power(b,p-1)

ans = power(b,p)
print(ans)
```

```
b:2.0
p:3
8.0
```

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

In [4]:

```
b = int(input("digit : "))

def digitcount(b,count):
    count = count
    if(b == 0):
        return count
    elif(b>0):
        count += 1
        return digitcount((int)(b/10),count)

ans = digitcount(b,0)
print("ans : ",ans)
```

```
digit : 45768
ans :  5
```

03) WAP to reverse an integer number using recursion.

In [11]:

```
n = int(input("n : "))

def reverseint(n,r = 0):
    if(n == 0):
        return r
    elif(n>0):
        r = r*10 + n%10
        return reverseint((int)(n/10),r)

ans = reverseint(n)
print("reverse = ",ans)
```

```
n : 6527
reverse =  7256
```

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

In [23]:

```
n = int(input("n : "))
A = 0

def dtob(n,a = []):
    if(n == 0):
        return a
    elif(n>0):
        a.append(n%2)
        return dtob((int)(n/2))

ans = dtob(n)
for i in ans[::-1]:
    A = A*10 + i
print(A)
```

```
n : 4
100
```

In [ ]: