



## Python Programming - 2101CS405

### Lab - 7

#### ▼ Functions

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

```
def simpleIntrest(a,r,t):
    intrest=amount*rate*time//100
    return intrest
amount=float(input("Enter Ptincipal amount : "))
rate=float(input("ENter intrest rate : "))
time=float(input("Enter a time in year : "))
print(f"simple intrest : {simpleIntrest(a=amount,r=rate,t=time)}")

Enter Ptincipal amount : 1000
ENter intrest rate : 2
Enter a time in year : 1
simple intrest : 20.0
```

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

```
def addition(n):
    sum=0
    for i in range(n):
        sum+=i
    return sum
num1=int(input("Enter a last number,n : "))
print(f"Sum of 1 to n number is : {addition(num1+1)}")

Enter a last number,n : 10
Sum of 1 to n number is : 55
```

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

```
max1=lambda a,b : a if a>b else b
num1=int(input("Enter a 1st number : "))
num2=int(input("ENter 2nd number : "))
print(f"max number : {max1(num1,num2)}")

Enter a 1st number : 12
ENter 2nd number : 11
max number : 12
```

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

```
def primeNumber(n):
    for i in range(2,int(n**0.5)+1):
        if(n%i==0):
            return False
    else:
        return True
n=int(input("Enter a number : "))
primeNumber(n)

Enter a number : 1
True
```

- ▼ 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.

```
# def primeNum(n):
#     for j in range(2,n+1):
#         for i in range(2,int(j**0.5)+1):
#             if(j%i==0):
#                 break;
#         else:
#             primeNum.append(j)
def primeNumber(n):
    for i in range(2,int(n**0.5)+1):
        if(n%i==0):
            return False
    else:
        return True
n=int(input("Enter a Number : "))
[i for i in range(2,n+1) if primeNumber(i)]

Enter a Number : 100
[2,
3,
5,
7,
11,
13,
17,
19,
23,
29,
31,
37,
41,
43,
47,
53,
59,
61,
67,
71,
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...)

```
def fibbo(n):
    list_fibonacci=[i for i in range(n)]
    for i in range(2,n):
        list_fibonacci[i]=list_fibonacci[i-1]+list_fibonacci[i-2]
    return list_fibonacci
number=int(input("Enter a number : "))
fibbo(number)

Enter a number : 8
[0, 1, 1, 2, 3, 5, 8, 13]
```

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

```
factorial=lambda n:1 if(n in [0,1]) else n*factorial(n-1)
print(factorial(int(input("Enter a number:"))))
```

```
Enter a number:4
24
```

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

```
num1=int(input("Enter a number1 : "))
num2=int(input("Enter a number2 : "))
calc=lambda a,b,op:a+b if op=='+' else a-b if op=='-' else a*b if op=='*' else round(a/b,2)
print(f'{num1} + {num2} : {calc(num1,num2,"+")}')
print(f'{num1} - {num2} : {calc(num1,num2,"-")}')
print(f'{num1} * {num2} : {calc(num1,num2,"*")}')
print(f'{num1} / {num2} : {calc(num1,num2,"/")}')

```

```
Enter a number1 : 12
Enter a number2 : 2
12 + 2 : 14
12 - 2 : 10
12 * 2 : 24
12 / 2 : 6.0
```

#### 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

```
sampleItems = "green-red-yellow-black-white"
list1=sampleItems.split("-")
list1.sort()
print("-".join(list1))
```

```
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)

```
a = int(input("Enter Number : "))
b = int(input("Enter Number : "))
positionalArguments = lambda a,b : a+b
print("Positional arguments ",positionalArguments(a,b))
```

```
a = int(input("Enter Number : "))
defaultArgument = lambda a,b=10 : a+b
print("Default argument ",defaultArgument(a))
```

```
Enter Number : 12
Enter Number : 12
Positional arguments 24
Enter Number : 10
Default argument 20
```

```
a = int(input("Enter Number : "))
b = int(input("Enter Number : "))
def keywordArguments(a,b):
    return a+b
print("Keyword arguments (named arguments) ",keywordArguments(b=a,a=b))
```

```
a = int(input("Enter Number : "))
def arbitraryArguments(a,*b):
    sums = a
    for i in b:
        sums+=i
    return sums
print("Keyword arguments (named arguments) ",arbitraryArguments(a,5,10,12,121))
```

```

Enter Number : 12
Enter Number : 12
Keyword arguments (named arguments) 24
Enter Number : 10
Keyword arguments (named arguments) 158

```

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

```

power=lambda x,y:1 if y==0 else x*power(x,y-1)
exponent=int(input("Enter a exponent : "))
pow=int(input("Enter a power : "))
print(f"{exponent}^{pow} : {power(exponent,pow)}")

```

```

Enter a exponent : 13
Enter a power : 2
13^2 : 169

```

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

```

digitSum=lambda n: 0 if n==0 else (n%10)+digitSum(n//10)
num1=int(input("Enter a number : "))
print(f"sum of digits of {num1} : {digitSum(num1)}")

```

```

Enter a number : 123456789
sum of digits of 123456789 : 45

```

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

```

digit=lambda n,r: r if n==0 else digit(n//10,(r*10)+(n%10))
num=int(input("ENter a number : "))
res=0
print(digit(num,res))

```

```

ENter a number : 123
321

```

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

```

def decimalToBinary(n):
    if n==0:
        return 0
    else:
        return n%2+10*(decimalToBinary(n//2))

```

```

n = int(input("Enter Number : "))
ans = decimalToBinary(n)
print("Binary : ",ans)

```

```

Enter Number : 8
Binary : 1000

```

✓ 5s completed at 4:40 AM

● ×