User Defined Function

01) Write a function to calculate BMI given mass and height. (BMI = mass/h**2)

```
h=float(input("Enter your height in M:"))
w=float(input("Enter your weight in K6:"))
def calculateBMI(h,b):
    bmi = b / (h ** 2)
    return bmi
print(f"Your BMI is: {calculateBMI(h,w)}")
```

```
Your BMI is: 19.568236333796236
```

02) Write a function that add first n numbers.

```
n = int(input("Enter N:"))
def sumOfN(n):
    sum=0
    while n>0:
        sum = sum + n
        n = n - 1
    return sum
print(f"Sum of first n number is: {sumOfN(n)}")
```

```
Sum of first n number is: 465
```

03) Write a function that returns 1 if the given number is Prime or 0 otherwise.

```
def isprime(n):
    for i in range(2,n):
        if n % i == 0:
            return 0
        else:
            return 1
isprime(17)
```

```
1
```

04) Write a function that returns the list of Prime numbers between given two numbers.

```
def listOfPrime(n,m):
    primes = []
```

```
    for i in range(n, m + 1):
        for j in range(2, i):
            if i % j == 0:
                break
            else:
                primes.append(i)
    return primes
listOfPrime(1,10)
```

```
[3, 5, 5, 5, 7, 7, 7, 7, 7, 9]
```

05) Write a function that returns True if the given string is Palindrome or False otherwise.

```
def ispalindrome(s):
    if s == s[::-1]:
        return True
    else:
        return False
ispalindrome("abc")
```

```
False
```

06) Write a function that returns the sum of all the elements of the list.

```
def sumOfList(l):
    sum = 0
    for i in l:
        sum = sum+i
    return sum
sumOfList([5,4,3,2,1])
```

```
15
```

07) Write a function to calculate the sum of the first element of each tuples inside the list.

```
def sumOfFirstElement(l):
    sum = 0
    for i in l:
        sum = sum+i[0]
    return sum

sumOfFirstElement([(1,2,3),(4,5,6)])
```

```
5
```

08) Write a recursive function to find nth term of Fibonacci Series.

```
def fibonacci(n):
    if n <= 1:
        return n
    return fibonacci(n - 1) + fibonacci(n - 2)

print(fibonacci(10))
```

```
55
```

09) Write a function to get the name of the student based on the given rollno.

Example: Given dict1 = {101:'Ajay', 102:'Rahul', 103:'Jay', 104:'Pooja'} find name of student whose rollno = 103

```
def getname(d, rollno):
    return d.get(rollno)

getname({101:'Ajay', 102:'Rahul', 103:'Jay', 104:'Pooja'},103)
```

```
'Jay'
```

10) Write a function to get the sum of the scores ending with zero.

Example : scores = [200, 456, 300, 100, 234, 678]

Ans = 200 + 300 + 100 = 600

```
def sumOfScore(l):
    sum=0
    for i in l:
        if i % 10 == 0:
            sum+=i
    return sum

sumOfScore([200,456,300,100,234,678])
```

```
600
```

11) Write a function to invert a given Dictionary.

hint: keys to values & values to keys

Before : {'a': 10, 'b':20, 'c':30, 'd':40}

After : {10:'a', 20:'b', 30:'c', 40:'d'}

```
def invertDictionary(d):
    id = {}
```

```
    for k, v in d.items():
        id.update({v:k})
    return id

print(invertDictionary({'a': 10, 'b': 20, 'c': 30, 'd' : 40}))
```

```
{10: 'a', 20: 'b', 30: 'c', 40: 'd'}
```

12) Write a function to check whether the given string is Pangram or not.

hint: Pangram is a string containing all the characters a-z atlest once.

"the quick brown fox jumps over the lazy dog" is a Pangram string.

```

```

13) Write a function that returns the number of uppercase and lowercase letters in the given string.

example : Input : s1 = AbcDEfgh ,Ouptput : no_upper = 3, no_lower = 5

```
def countUpperAndLower(s):
    upper = 0
    lower = 0
    for i in s:
        if i.isupper():
            upper = upper+1
        elif i.islower():
            lower = lower+1
    return upper, lower

countUpperAndLower("AbcDEfgh")
```

```
(3, 5)
```

14) Write a lambda function to get smallest number from the given two numbers.

```
min = lambda a,b : a if a<b else b
min(53,10)
```

```
10
```

15) For the given list of names of students, extract the names having more that 7 characters. Use filter().

```
studentName = filter(lambda x : len(x) > 7,
["zxxczc","abcefghijk","lmnopwq"])
```

```
for i in studentName:
    print(i)

abcefghijk
```

16) For the given list of names of students, convert the first letter of all the names into uppercase. use map().

```
studentName = map(lambda x : x.capitalize() ,
["zxxczc","abcefghijk","lmnopwq"])
for i in studentName:
    print(i)

Zxxczc
Abcefghijk
Lmnopwq
```

17) Write udfs to call the functions with following types of arguments:

1. Positional Arguments
2. Keyword Arguments
3. Default Arguments
4. Variable Legngth Positional(*args) & variable length Keyword Arguments (**kwargs)
5. Keyword-Only & Positional Only Arguments

```
def positional(a,b):
    return a+b

print(positional(5,10))
#-------------------------
def keyword(a,b):
    return a+b

print(keyword(a=5,b=10))
#-------------------------
def default(a=0,b=0):
    return a+b

default()
#-------------------------
def variableLenPos(*args):
    sum=0
    for i in args:
        sum+=i
    return sum

print(variableLenPos(10,20,30))
#-------------------------
def lenKeyword(**kyword):
    return kyword
```

```
print(lenKeyword(a=10,b=20,c=30))
#-------------------------
def positionalOnly(a,b,/):
    return a+b

print(positionalOnly(10,20))
#-------------------------
def keywordOnly(*,a,b):
    return a+b

print(keywordOnly(a=10,b=20))
```

```
15
15
60
{'a': 10, 'b': 20, 'c': 30}
30
30
```