AI Programming
[Day-6]

To understand function basics in Python

Program 1:[Function]Perfect number

Design and code a function viz., "perfect()" that determines if parameter number is a perfect number. Use this function in a program that determines and prints all the perfect numbers between 1 and N in a list.
[An integer number is said to be "perfect number" if its factors, including 1(but not the number itself), sum to the number. E.g., 6 is a perfect number because 6=1+2+3].

TC1
Input:
Enter the value of N: 1000
Output:
Perfect numbers are: [6, 28, 496]

TC2
Input:

Enter the value of N: 10000
Output:
Perfect numbers are: [6, 28, 496, 8128]

TC3
Input:
Enter the value of N: 50
Output:
Perfect numbers are: [6, 28]

Solution:

```
#1 [Function]Perfect number
def perfect(n):
 list1=[]
 for i in range(2,n):
  sum=0
  for j in range(1,i):
   if i%j==0:
    sum+=j
  if sum==i:
   list1.append(i)
 print("Perfect numbers are:",list1)
n=int(input("Enter the Value of N: "))
perfect(n)
```

Output:

```
Enter the Value of N: 1000
Perfect numbers are: [6, 28, 496]
```

Program 2: [Function]Even length words identification

A professor writes some sentences on whiteboard in a classroom and asks his students to

design a function in Python to extract even length words in a list from the given sentences.

Solution:

```
2 [Function]Even length words identification
def even_length(a):
 list1=[]
 for i in a.split():
  if len(i)%2==0:
   list1.append(i)
 print("Extracted even length words are: ",list1)
print("Enter Sentences:")
a=input()
even_length(a)
```

Output:

```
Enter Sentences:
this is a python function to extract
even length words
Extracted even length words are:
['this', 'is', 'python', 'function',
'to', 'even', 'length']
```

Program3: [Function]Square&amp;Cube

One of the parents, in a family, gives a number N and asks his child to extract list of those

numbers that are both squared and cube of some numbers but should be bounded by N. Also, find total such numbers.

TC1
Input:
Enter the number:
1000
Output:
Desired numbers are [1, 64, 729] and total count is 3

TC2
Input:
Enter the number:
5000
Output:
Desired numbers are [1, 64, 729, 4096] and total count is 4

TC3
Input:
Enter the number:
50000
Output:
Desired numbers are [1, 64, 729, 4096, 15625, 46656] and total count is 6

Solution:

```
#3. [Function]Square&Cube
def square_cube(num):
  a=1
  list1=[]
  while (a**6)<=num:
    list1.append(a**6)
    a+=1
  print("Desired numbers are %s and total
count is %s"%(list1,len(list1)))
print("Enter the number")
num=int(input())
square_cube(num)
```

Output:

```
Enter the number
1000
Desired numbers are [1, 64, 729] and
total count is 3
```

Program 4:[Function]Duplicacy removal

While distributing the sweets Rohan on his birthday found that at the end of the distribution he was running in short of sweets. Though, he got 1 sweet for every classmate. He then realized that somebody had taken the sweet more than once. He reported this issue to the class teacher. To avoid such things the teacher made a rule that once the student had taken the sweet his roll no will be recorded in a list. Thus, if at the end there is any duplicacy in the roll number in the list then 'False' is called out else 'True'. Write a function that does the same i.e returns 'True' if the list contains all unique numbers else returns 'False'.

Solution:

```
#4. [Function]Duplicacy removal
def duplicacy_removal():
  list1=[]
  num=int(input("Enter length of list: "))
  for i in range(num):
    val=int(input())
    list1.append(val)
  print(list1)
  list2=list(set(list1))
  if len(list2)==len(list1):
    print(True)
  else:
    print(False)
duplicacy_removal()
```

Output:

```
Enter length of list: 5
21
1
45
25
10
[21, 1, 45, 25, 10]
True
```

Program 5: [function]Passing a list to a function

Write a function which will receive a list of numeric values, changes the negative values to corresponding positive values (example: -5 will be changed to 5) and returns the changed list. After receiving the changed list, display the original list, the changed list along with the number of places where the changed list is different from the original list.

```
Input:
[1,2,-3,8,5,6,0,9]
Output:
[1, 2, -3, 8, 5, 6, 0, 9]
[1, 2, 3, 8, 5, 6, 0, 9]
1
```

```
Input:
[-3,2,3,4,5,-1,0,10,-20]
Output:
[-3, 2, 3, 4, 5, -1, 0, 10, -20]
[3, 2, 3, 4, 5, 1, 0, 10, 20]
3
```

Solution:

```python
#5. [function]Passing a list to a function
def function():
  list1=eval(input())
  val=0
  for i in range(len(list1)):
    if list1[i]<0:
      list1[i]=list1[i]*(-1)
      val+=1
  print(list1)
  print(val)
function()
```

Output:

```
8,9,7
(8, 9, 7)
0
```