Task 5

Registration ID : SIRSS1038

Leela Satya Kartheek Raja

Q1. Given a list of integers, write a function to return the sum of all prime numbers in that list.

In [2]:

```
import math
def prime(n):
    if n == 0 or n == 1:
        return False
    else:
        for i in range(2, int(n**0.5)+1):
            if n%i == 0:
                return False
                break
        else:
            return True

def sum_prime(ls):
    sum_of_primes=0
    for i in ls:
        if prime(i):
            print(i)
            sum_of_primes += i
    return sum_of_primes
```

In [4]:

```
sum_prime([0,1,2,3,4,5,10,11])
```

```
2
3
5
11
```

Out[4]:

```
21
```

Q2. Given a list of integers, write a function to check whether the list is strictly increasing or not.

In [5]:

```
def increasing(ls):
    for i in range(1,len(ls)):
        if ls[i] < ls[i-1]:
            return False
    return True
```

In [6]:

```
increasing([0,-1,2,3,4,5,7])
```

Out[6]:

```
False
```

Q3. Write a function to check whether a given list is expanding or not (the difference between adjacent elements should keep on increasing).

In [8]:

```
def expanding(l):
    d1 = abs(l[1] - l[0])
    d2 = abs(l[2] - l[1])

    if d1 >= d2:
        return False

    else:
        for i in range(3,len(l)):
            d3 = abs(l[i] - l[i-1])
            if d3 <= d2:
                return False
            d2 = d3
        else:
            return True
```

In [9]:

```
expanding([1,3,7,2,9])
```

Out[9]:

```
True
```

Q4. Write a function to calculate all permutations of a given string. (Without using itertools)

In [12]:

```
def toString(lst):
    return ''.join(lst)
def permute(a, l, r):
    if l==r:
        print(toString(a))
    else:
        for i in range(l,r+1):
            a[l], a[i] = a[i], a[l]
            permute(a, l+1, r)
            a[l], a[i] = a[i], a[l]

if __name__=='__main__':
    string = input("Enter the word : ")
    n = len(string)
    a = list(string)
    print('Permutations of given string  '+string+' are :')
    permute(a, 0, n-1)
```

```
Enter the word : abc
Permutations of given string  abc are :
abc
acb
bac
bca
cba
cab
```

In [ ]:

```
```