# Assignment 5

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

In [1]:

```
numbers=input("").split(' ')
sumPrimeNumbers=0
for n in numbers:
    n=int(n)
    isPrime=True
    if n > 1:

        for i in range(2, n):
            if (n % i) == 0:
                isPrime=False
                break
        if isPrime:
            sumPrimeNumbers+=n


print(str(sumPrimeNumbers))
```

```
1 2 2 5 6 14 2 12 1 2 12 12 1
13
```

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

In [3]:

```
test_list = [15,12,10,8,19]

print ("Original list : " + str(test_list))

res = bool(lambda test_list: reduce(lambda i, j: j
                if
                i < j else 9999, test_list) != 9999)


print ("Is list strictly increasing  : " + str(res))
```

```
Original list : [15, 12, 10, 8, 19]
Is list strictly increasing  : True
```

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

In [4]:

```
def exp(l):
    diff=0
    flag=True
    for i in range(1,len(l)):
        diff1=l[i]-l[i-1]
        if diff>diff1:
            flag=False
            break
        diff=diff1

    return  flag
l1=input().split()
list1=list(map(lambda x:int(x),l1))
print('List is expanding: ',exp(list1))
```

```
20 25 30 35 40
List is expanding:  True
```

# Q4. Write a function to calculate all permutations of a given string. (Without using iterto

In [6]:

```
def permutation(str) :
    if len(str) == 1 :
        return [str]
    prmts = permutation(str[1:])
    fst = str[0]
    rslt = []
    for i in prmts :
        for j in range(len(i)+1) :
            rslt.append(i[:j] + fst + i[j:])
    return rslt


inp = str(input("Enter the String : "))
permutation(inp)
```

```
Enter the String : HARI
```

Out[6]:

```
['HARI',
 'AHRI',
 'ARHI',
 'ARIH',
 'HRAI',
 'RHAI',
 'RAHI',
 'RAIH',
 'HRIA',
 'RHIA',
 'RIHA',
 'RIAH',
 'HAIR',
 'AHIR',
 'AIHR',
 'AIRH',
 'HIAR',
 'IHAR',
 'IAHR',
 'IARH',
 'HIRA',
 'IHRA',
 'IRHA',
 'IRAH']
```

In [ ]:

```

```