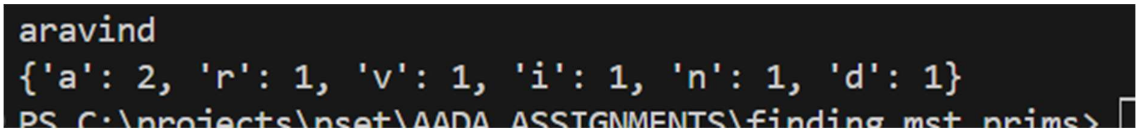
Q1 to count the number of strings in dictionary

Input

```
a=input()
t={}
for i in a:
    if i in t:
      t[i]=t[i]+1
    else:
      t[i]=1
print(t)
```

output



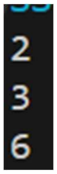
```
aravind
{'a': 2, 'r': 1, 'v': 1, 'i': 1, 'n': 1, 'd': 1}
PS C:\projects\pset\AADA ASSIGNMENTS\finding mst prims>
```

Q to print gcd and lcm

Input

```
a=int(input())
b=int(input())
def num(t,s):
   t=max(a,b)
   s=min(a,b)
   while(True):
      if((t%a==0) and (t%b==0)):
         l=t
         break
      t=t+1
   return l
s=num(a,b)
print(s)
```

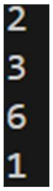
output



```
2
3
6
```

Printing gcd

```
import math
h=math.gcd(a,b)
print(h)
```



```
2
3
6
1
```

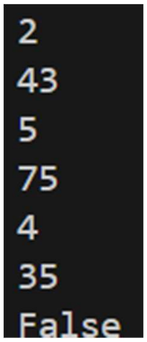
Q4 to find whther they are colliding or not

Input

```
x1=int(input())
y1=int(input())
r1=int(input())
x2=int(input())
y2=int(input())
r2=int(input())
t=(x1,y1,r1)
s=(x2,y2,r2)
import math
r=r1+r2
def colide(t,s,r):
    d=math.sqrt(pow(t[0]-s[0],2)+pow(t[1]-s[1],2))
    if(d<=r):
        return True
    else:
        return False
h=colide(t,s,r)
```

```
print(h)
```

output



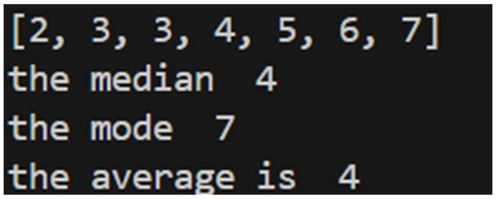
```
2
43
5
75
4
35
False
```

Q5 to print the mode ,median of the numbers

Input

```
l=[3,5,2,3,4,7,6]
l.sort()
t=len(l)
print(l)
k=0
if(t%2==0):
   k=(l[t//2]+l[(t//2)+1])/2
else:
   k=l[(t//2)]
print("the median ",k)
print("the mode ",l[t-1])
sum=0
for i in l:
   sum=sum+i
sum=sum//t
print("the average is ",sum)
```

output



```
[2, 3, 3, 4, 5, 6, 7]
the median  4
the mode  7
the average is  4
```

Q to sort using

A bubble sort b insertion c selection d ..

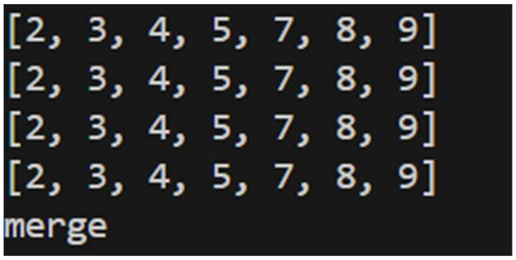
Input

```
arr=[2,5,4,8,7,3,9]
t=len(arr)

def fro(arr):
    n = len(arr)
    for i in range(n-1):
        for j in range(n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
    return arr
h=fro(arr)
print(h)

def mergeSort(arr):
    if len(arr) <= 1:
        return arr

    mid = len(arr) // 2
    leftHalf = arr[:mid]
    rightHalf = arr[mid:]

    sortedLeft = mergeSort(leftHalf)
    sortedRight = mergeSort(rightHalf)

    return merge(sortedLeft, sortedRight)

def merge(left, right):
    result = []
    i = j = 0

    while i < len(left) and j < len(right):
        if left[i] < right[j]:
            result.append(left[i])
            i += 1
        else:
            result.append(right[j])
            j += 1

    result.extend(left[i:])
    result.extend(right[j:])

    return result
j=mergeSort(arr)
print(j)


def selectionSort(array, size):

    for step in range(size):
        min_idx = step

        for i in range(step + 1, size):

            if array[i] < array[min_idx]:
                min_idx = i


        (array[step], array[min_idx]) = (array[min_idx], array[step])
selectionSort(arr,t)
```

```
print(arr)

def insertionSort(array):

    for step in range(1, len(array)):
        key = array[step]
        j = step - 1


        while j >= 0 and key < array[j]:
            array[j + 1] = array[j]
            j = j - 1


        array[j + 1] = key
insertionSort(arr)
print(arr)
def arg(ar):
    match ar:
        case 0:
            return "buble"
        case 1:
            return "merge"
        case 2:
            return "selection"
        case 3:
            return "insertion"
        case default:
            return "something"

b=arg(1)
```

```
print(b)
```

output

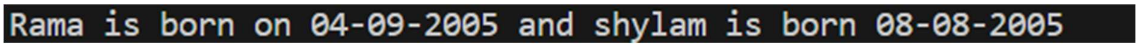


```
[2, 3, 4, 5, 7, 8, 9]
[2, 3, 4, 5, 7, 8, 9]
[2, 3, 4, 5, 7, 8, 9]
[2, 3, 4, 5, 7, 8, 9]
merge
```

Q2 to print the birthday

```
t={
   "rama":"04-09-2005",
   "shylam":"08-08-2005"
   }
print("Rama is born on",t["rama"],"and shylam is born",t["shylam"])
r=t["rama"].split(" ")
h=t["shylam"].split(" ")
```

output



```
Rama is born on 04-09-2005 and shylam is born 08-08-2005
```