1. **LIST COMPREHENSION PROGRAM**

**if** \_\_name\_\_ == '\_\_main\_\_':

    x = **int**(**input**())

    y = **int**(**input**())

    z = **int**(**input**())

    n = **int**(**input**())

    output=[]

    klm=[]

**for** X **in** **range**(x+1):

**for** Y **in** **range**(y+1):

**for** Z **in** **range**(z+1):

**if**(X+Y+Z!=n):

                    klm=[X,Y,Z]

                    output.append(klm)

**print**(output)

1. **FIND THE RUNNER-UP SCORE PROGRAM**

**if** \_\_name\_\_ == '\_\_main\_\_':

    n = **int**(**input**())

    arr = **map**(**int**, **input**().split())

    arr=**list**(arr)

    a=**max**(arr)

    b=-9999999

**for** i **in** **range**(0,n):

**if** arr[i]<a **and** arr[i] >b:

            b=arr[i]

**print**(b)

1. **NESTED-LISTS**

**if** \_\_name\_\_ == '\_\_main\_\_':

    scorelist=[]

**for** \_ **in** **range**(**int**(**input**())):

        name = **input**()

        score = **float**(**input**())

        scorelist.append([name,score])

secondhigh=**sorted**(**set**([score **for** name,score **in** scorelist]))[1]

**print**('\n'.join(**sorted**([name **for** name,score **in** scorelist **if** score==secondhigh])))

1. **FINDING PERCENTAGE PROGRAM**

**if** \_\_name\_\_ == '\_\_main\_\_':

    n = **int**(**input**())

    student\_marks = {}

**for** \_ **in** **range**(n):

        name, \*line = **input**().split()

        scores = **list**(**map**(**float**, line))

        student\_marks[name] = scores

    query\_name = **input**()

marks=**list**(student\_marks[query\_name])

percentage=**sum**(marks)/**len**(marks)

**print**("%.2f" % percentage)

1. **LIST OPERATIONS PROGRAM**

**if** \_\_name\_\_ == '\_\_main\_\_':

    N = **int**(**input**())

    final=[]

**for** i **in** **range**(0,N):

        a=**input**().split()

**if**(a[0]=="print"):

**print**(final)

**elif**(a[0]=="remove"):

            final.remove(**int**(a[1]))

**elif**(a[0]=="append"):

            final.append(**int**(a[1]))

**elif**(a[0]=="insert"):

            final.insert(**int** (a[1]),**int** (a[2]))

**elif**(a[0]=="sort"):

            final.sort()

**elif**(a[0]=="pop"):

            final.pop()

**else**:

            final.reverse()

1. **TUPLE HASH FUNCTION PROGRAM**

**if** \_\_name\_\_ == '\_\_main\_\_':

    n = **int**(**raw\_input**())

    integer\_list = **map**(**int**, **raw\_input**().split())

    a=**tuple**(integer\_list)

**print**(**hash**(a))

1. **SETS -AVERAGE PROGRAM**

**def** average(array):

*# your code goes here*

    array1=**sum**(**set**(array))

    length=**len**(**set**(array))

    average=array1/length

**return** average

**if** \_\_name\_\_ == '\_\_main\_\_':

    n = **int**(**input**())

    arr = **list**(**map**(**int**, **input**().split()))

    result = average(arr)

**print**(result)

1. **DISJOINT SETS PROGRAM**

k=**input**().split()

l=**int**(k[0])

m=**int**(k[1])

s=**list**()

count=0

s=**list**(**map**(**int**,**input**().strip().split()))

a=**set**(**map**(**int**,**input**().strip().split()))

b=**set**(**map**(**int**,**input**().strip().split()))

**for** i **in** s:

**if** i **in** a:

        count+=1

**if** i **in** b:

        count-=1

**print**(count)

1. **SYMMETRIC DIFFERENCE PROGRAM**

a=**int**(**input**())

aset=**set**(**map**(**int**,**input**().split()))

b=**int**(**input**())

bset=**set**(**map**(**int**,**input**().split()))

adif=aset.difference(bset)

bdif=bset.difference(aset)

symdif=adif.union(bdif)

**for** i **in** **sorted**(**list**(symdif)):

**print**(i)

1. **ADD OPERATION IN SETS PROGRAM**

n=**int**(**input**())

countries=**set**()

**for** i **in** **range**(n):

    countries.add(**input**())

**print**(**len**(countries))

1. **SETS DISCARD,POP&REMOVE OPERATIONS**

n = **int**(**input**())

s = **set**(**map**(**int**, **input**().split()))

a=**int**(**input**())

**for** i **in** **range**(a):

    k=**input**().split()

**if**(k[0]=="remove"):

        s.remove(**int**(k[1]))

**elif**(k[0]=="discard"):

        s.discard(**int**(k[1]))

**else**:

        s.pop()

**print**(**sum**(**list**(s)))

1. **UNION OPERATION PROGRAM**

A=**int**(**input**())

Aset=**set**(**map**(**int**,**input**().split()))

B=**int**(**input**())

Bset=**set**(**map**(**int**,**input**().split()))

cset=Aset.union(Bset)

**print**(**len**(cset))

1. **INTERSECTION OPERATION PROGRAM**

A=**int**(**input**())

Aset=**set**(**map**(**int**,**input**().split()))

B=**int**(**input**())

Bset=**set**(**map**(**int**,**input**().split()))

cset=Aset.union(Bset)

**print**(**len**(cset))

1. **DIFFERENCE OPERATION PROGRAM**

a=**int**(**input**())

s1=**set**(**input**().split())

b=**int**(**input**())

s2=**set**(**input**().split())

s3=s1.difference(s2)

**print**(**len**(s3))

1. **SYMMETRIC\_DIFFERENCE PROGRAM**

a=**int**(**input**())

s1=**set**(**input**().split())

b=**int**(**input**())

s2=**set**(**input**().split())

s3=s1.symmetric\_difference(s2)

**print**(**len**(s3))

1. **SET MUTATIONS PROGRAM**

len\_set=**int**(**input**())

s=**set**(**map**(**int**,**input**().split()))

op\_len=**int**(**input**())

**for** i **in** **range**(op\_len):

    operation=**input**().split()

**if** operation[0]=="intersection\_update":

        temp=**set**(**map**(**int**,**input**().split()))

        s.intersection\_update(temp)

**elif** operation[0]=="update":

         temp=**set**(**map**(**int**,**input**().split()))

         s.update(temp)

**elif** operation[0]=="symmetric\_difference\_update":

        temp=**set**(**map**(**int**,**input**().split()))

        s.symmetric\_difference\_update(temp)

**elif** operation[0]=="difference\_update":

        temp=**set**(**map**(**int**,**input**().split()))

        s.difference\_update(temp)

**else**:

**assert** false

**print**(**sum**(s))

1. **CAPTAIN’S ROOM PROGRAM**

N=**int**(**input**())

s=**map**(**int**,**input**().split())

s=**sorted**(s)

**for** i **in** **range**(**len**(s)):

**if**(i!=**len**(s)-1):

**if**(s[i]!=s[i-1] **and** s[i]!=s[i+1]):

**print**(s[i])

**break**

**else**:

**print**(s[i])

1. **SUBSET OPERATION PROGRAM**

num=**int**(**input**())

**for** i **in** **range**(num):

    a=**input**()

    k=**set**(**input**().split())

    b=**input**()

    l=**set**(**input**().split())

**print**(k.issubset(l))

1. **STRICT SUPERSET OPERATION PROGRAM**

a=**set**(**input**().split())

n=**int**(**input**())

check=**True**

**for** i **in** **range**(n):

    b=**set**(**input**().split())

**if** **not** b.issubset(a):

        check=**False**

**if**(**len**(b)>=**len**(a)):

        check=**False**

**print**(check)