

In []:



In []:



```
#list comprehensions
```

In []:



```
x = int(input())
y = int(input())
z = int(input())
n = int(input())

output=[]
for a in range(x+1):
    for b in range(y+1):
        for c in range(z+1):
            if a+b+c==n:
                continue
            else:
                output.append([a,b,c])

print(output)
```

In []:



```
# Runner up score
```

In []:



```
n = int(input())
value = map(int,input().split())
print(sorted(list(set(value)))[-2])
```

In []:



```
# Finding the percentage
```

In []:



```
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 = input()
    mark=0
    for i in student_marks[query]:
        mark=mark+i
    average=mark/3
    print("%.2f"%average)
```

In []:



Nested lists

In []:



```
if __name__ == '__main__':
    result=[]
    score=[]
    for _ in range(int(input())):
        name = input()
        mark = float(input())
        result+=[[name,mark]]
        score+=[mark]
    b=sorted(list(set(score)))[1]
    for a,c in sorted(result):
        if c==b:
            print(a)
```

In [16]:



Lists

In []:



```
if __name__ == '__main__':
    N = int(input())
    command=[]
    for i in range(N):
        command.append(input().split())

    result=[]
    for i in range(N):
        if command[i][0]=='insert':
            result.insert(int(command[i][1]),int(command[i][2]))
        elif command[i][0]=='print':
            print(result)
        elif command[i][0]=='remove':
            result.remove(int(command[i][1]))
        elif command[i][0]=='append':
            result.append(int(command[i][1]))
        elif command[i][0]=='pop':
            result.pop()
        elif command[i][0]=='sort':
            result.sort()
        elif command[i][0]=='reverse':
            result.reverse()
```

In []:



```
# introduction to sets
```

In []:



```
def average(array):
    s=set(array)
    return sum(s)/len(s)

if __name__ == '__main__':
    n = int(input())
    arr = list(map(int, input().split()))
    result = average(arr)
    print(result)
```

In []:



```
# no idea
```

In []:



```
n,m=list(map(int, input().split()))
ns=list(map(int, input().split()))
h=set(map(int, input().split()))
s=set(map(int, input().split()))
res=0
for x in ns:
    if x in h:
        res+=1
    elif x in s:
        res-=1
print(res)
```

In []:



```
# Symmetric Difference
```

In []:



```
if __name__ == "__main__":
    M = int(input().strip())
    set_m = set(map(int, input().strip().split(' ')))

    N = int(input().strip())
    set_n = set(map(int, input().strip().split(' ')))

    for j in sorted(set_m ^ set_n):
        print(j)
```

In []:



```
# set.add()
```

In []:



```
s = set()
for i in range(int(input())):
    s.add(input())
print(len(s))
```

In []:



```
# set.discard().remove()
```

In []:



```
n = int(input())
s = set(map(int, input().split()))
for i in range(int(input())):
    s1 = input().split()
    if s1[0] == 'pop':
        s.pop()
    elif s1[0] == 'remove':
        s.remove(int(s1[1]))
    elif s1[0] == 'discard':
        s.discard(int(s1[1]))
print(sum(s))
```

In []:



```
# set.union() operation
```

In []:



```
num1 = int(input())
setA = set(map(int, input().split()))
num2 = int(input())
setB = set(map(int, input().split()))

print(len(setA.union(setB)))
```

In []:



```
# set.intersection() operation
```

In []:



```
N1 = int(input())
storage1 = set(input().split())

N2 = int(input())
storage2 = set(input().split())

storage3 = storage2.intersection(storage1)

print(len(storage3))
```

In []:



```
# set.difference() operation
```

In []:

```
N1 = int(input())
storage1 = set(input().split())

N2 = int(input())
storage2 = set(input().split())

storage3 = storage1.difference(storage2)

print(len(storage3))
```

In []:

```
# set.symmetric_difference() operation
```

In []:

```
_, a = input(), set(input().split())
_, b = input(), set(input().split())
print(len(a.symmetric_difference(b)))
```

In []:

```
# set mutations
```

In []:

```
_ = input()
a = set(int(u) for u in input().split(' '))

n = int(input())
for _ in range(n):
    op, _ = input().split(' ')
    b = set(int(u) for u in input().split(' '))
    if op == "update":
        a |= b
    elif op == "intersection_update":
        a &= b
    elif op == "difference_update":
        a -= b
    elif op == "symmetric_difference_update":
        a ^= b

print(sum(a))
```

In []:

```
# The captains room
```

In []:



```
N = int(input())

storage = map(int, input().split())
storage = sorted(storage)

for i in range(len(storage)):
    if(i != len(storage)-1):
        if(storage[i]!=storage[i-1] and storage[i]!=storage[i+1]):
            print(storage[i])
            break;
    else:
        print(storage[i])
```

In []:



```
# check subset
```

In []:



```
for i in range(int(input())):
    a = int(input()); A = set(input().split())
    b = int(input()); B = set(input().split())
    print(A.issubset(B))
```

In []:



```
# check strict superset
```

In []:



```
storage = set(input().split())
N = int(input())
output = True

for i in range(N):
    storage2 = set(input().split())
    if not storage2.issubset(storage):
        output = False
    if len(storage2) >= len(storage):
        output = False

print(output)
```

In []:

