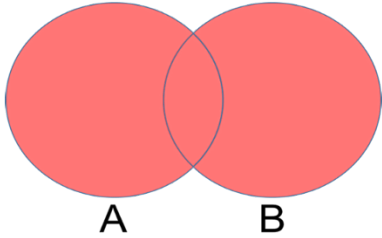
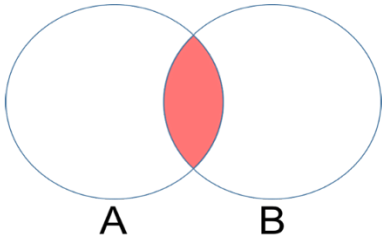
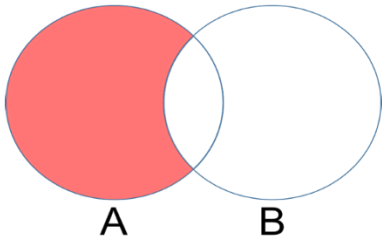
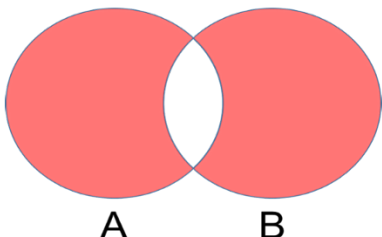


Set Operation	Venn Diagram	Interpretation
Union		$A \cup B$, is the set of all values that are a member of A , or B , or both.
Intersection		$A \cap B$, is the set of all values that are members of both A and B .
Difference		$A \setminus B$, is the set of all values of A that are not members of B
Symmetric Difference		$A \triangle B$, is the set of all values which are in one of the sets, but not both.

union(*others)

set | other | ...

Return a new set with elements from the set and all others.

```
>>> A={1,2,3,4,5}
>>> B={1,2,3,6,7}
>>> C=A.union(B)
>>> print(A,B,C,sep="\n")
{1, 2, 3, 4, 5}
{1, 2, 3, 6, 7}
```

```

{1, 2, 3, 4, 5, 6, 7}
>>> s1={1,2,3}
>>> s2={4,5,6}
>>> s3={7,8,9}
>>> s4=s1|s2 |s3
>>> print(s1,s2,s3,s4,sep="\n")
{1, 2, 3}
{4, 5, 6}
{8, 9, 7}
{1, 2, 3, 4, 5, 6, 7, 8, 9}
>>>

```

<https://www.hackerrank.com/challenges/py-set-union/problem?isFullScreen=false>

```

n=int(input())
eng=set(map(int,input().split(" ")[:n]))
b=int(input())
fr=set(map(int,input().split(" ")[:b]))
print(len(eng.union(fr)))

```

intersection(*others)

set & other & ...

Return a new set with elements common to the set and all others.

```

>>> set3=set1.intersection(set2)
>>> print(set1,set2,set3,sep="\n")
{1, 2, 3, 4, 5}
{1, 2, 3, 6, 7}
{1, 2, 3}
>>> java_students={'naresh','suresh','kishore','ramesh'}
>>> python_students={'kishore','rajesh','kiran','naresh'}
>>> java_python_students=java_students.intersection(python_students)
>>> print(java_students,python_students,java_python_students,sep="\n")
{'kishore', 'naresh', 'ramesh', 'suresh'}
{'kiran', 'rajesh', 'kishore', 'naresh'}
{'kishore', 'naresh'}
>>> s1={1,2,3,4}
>>> s2={1,2,4,5,6}
>>> s3={4,5,6,7,8}

```

```
>>> s4=s1&s2&s3
>>> print(s1,s2,s3,s4,sep="\n")
{1, 2, 3, 4}
{1, 2, 4, 5, 6}
{4, 5, 6, 7, 8}
{4}
>>>
```

difference(*others)

set - other - ...

Return a new set with elements in the set that are not in the others.

```
>>> A={1,2,3,4,5}
>>> B={1,2,3,6,7}
>>> C=A.difference(B)
>>> print(A,B,C,sep="\n")
{1, 2, 3, 4, 5}
{1, 2, 3, 6, 7}
{4, 5}
>>> D=A-B
>>> print(A,B,D,sep="\n")
{1, 2, 3, 4, 5}
{1, 2, 3, 6, 7}
{4, 5}
>>>
```

<https://www.hackerrank.com/challenges/py-set-difference-operation/problem?isFullScreen=false>

```
# Enter your code here. Read input from STDIN. Print output to STDOUT
n=int(input())
english=set(map(int,input().split(" ")[:n]))
m=int(input())
french=set(map(int,input().split(" ")[:m]))
only_english=english-french
print(len(only_english))
```

symmetric_difference(other)

set ^ other

Return a new set with elements in either the set or *other* but not both.

```

>>> A={1,2,3}
>>> B={1,2,4}
>>> C=A.symmetric_difference(B)
>>> print(A,B,C,sep="\n")
{1, 2, 3}
{1, 2, 4}
{3, 4}
>>>

```

<https://www.hackerrank.com/challenges/symmetric-difference/problem?isFullScreen=false>

```

m=int(input())
A=set(map(int,input().split(" ")[:m]))
n=int(input())
B=set(map(int,input().split(" ")[:n]))
C=A^B
l=list(C)
l.sort()
for value in l:
    print(value)

```

Mutable set operators

update(others*)**

set |= other | ...

Update the set, adding elements from all others.

```

>>> set1={1,2,3,4,5}
>>> set2={6,7,8,9,10}
>>> set1.update(set2)
>>> print(set1,set2,sep="\n")
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
{6, 7, 8, 9, 10}
>>>

```

intersection_update(*others)

set &= other & ...

Update the set, keeping only elements found in it and all others.

```
>>> set1={1,2,3,4,5}
>>> set2={1,2,3,6,7}
>>> set1.intersection_update(set2)
>>> print(set1)
{1, 2, 3}
>>>
```

difference_update(*others)

set -= other | ...

Update the set, removing elements found in others.

```
>>> python_students={'naresh','suresh','kishore','ramesh'}
>>> java_students={'kishore','ramesh','rajesh','kiran'}
>>> python_students.difference_update(java_students)
>>> print(python_students)
{'naresh', 'suresh'}
>>> print(java_students)
{'kiran', 'rajesh', 'kishore', 'ramesh'}
>>>
```

symmetric_difference_update(other)

set ^= other

Update the set, keeping only elements found in either set, but not in both.

```
>>> set1={1,2,3,4,5}
>>> set2={1,2,3,6,7}
>>> set1^=set2
>>> print(set1)
{4, 5, 6, 7}
>>>
```

<https://www.hackerrank.com/challenges/py-set-mutations/problem?isFullScreen=false>

```
m=int(input())
A=set(map(int,input().split(" ")[:m]))
n=int(input())
for i in range(n):
    cmd=input().split(" ")
    B=set(map(int,input().split():int(cmd[1])))
    if cmd[0]=="update":
        A.update(B)
    elif cmd[0]=="difference_update":
        A.difference_update(B)
    elif cmd[0]=="intersection_update":
        A.intersection_update(B)
    elif cmd[0]=="symmetric_difference_update":
        A.symmetric_difference_update(B)

print(sum(A))
```