Set .discard(), .remove() & .pop()

HackerRank

Problem Statement

.remove(x)

This operation removes element x from set.
If element x is not in the set, it raises a `KeyError`.
$.remove(x)$ operation returns `None`.

Example

```
>>> s = set([1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> s.remove(5)
>>> print s
set([1, 2, 3, 4, 6, 7, 8, 9])
>>> print s.remove(4)
None
>>> print s
set([1, 2, 3, 6, 7, 8, 9])
>>> s.remove(0)
KeyError: 0
```

.discard(x)

This operation also removes element x from set.
But if element x is not in the set, it **does not** raises a KeyError.
$.discard(x)$ operation returns `None`.

Example

```
>>> s = set([1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> s.discard(5)
>>> print s
set([1, 2, 3, 4, 6, 7, 8, 9])
>>> print s.discard(4)
None
>>> print s
set([1, 2, 3, 6, 7, 8, 9])
>>> s.discard(0)
>>> print s
set([1, 2, 3, 6, 7, 8, 9])
```

.pop()

This operation removes and return an arbitrary element from set.
If there are no elements to remove, it raises a `KeyError`.

Example

```
>>> s = set([1])
>>> print s.pop()
1
>>> print s
set([])
```

```
>>> print s.pop()
KeyError: pop from an empty set
```

Task

You have a non-empty set s and you have to execute N commands given in N lines.

Commands will be pop, $remove$ and $discard$.

Input Format

First line contains integer n, number of elements in set.
Second line contains space separated elements of set s. All elements are non-negative integers, less than or equal to 9.
Third line contains integer N, number of commands.
Next N lines contains pop, remove and discard commands.

Constraints

$0 < n < 20$
$0 < N < 20$

Output Format

Print sum of elements of set 's' in output.

Sample Input

```
9
1 2 3 4 5 6 7 8 9
10
pop
remove 9
discard 9
discard 8
remove 7
pop
discard 6
remove 5
pop
discard 5
```

Sample Output

```
4
```

Explanation

On application of these 10 operations on set, we get set([4]). Hence, sum is 4.

Note : Convert elements of set s to $integers$ while assigning. To ensure proper input of set we have added, first two lines of code to the editor.