1) List Comprehensions

In [1]:
```
if __name__ == '__main__':
    x = int(input())
    y = int(input())
    z = int(input())
    n = int(input())
    print( [[i,j,k] for i in range( x + 1) for j in range( y + 1) for k in r
```

```
1
1
1
2
[[0, 0, 0], [0, 0, 1], [0, 1, 0], [1, 0, 0], [1, 1, 1]]
```

2) Find the Runner-Up Score!

In [2]:
```
if __name__ == '__main__':
    n = int(input())
    arr = map(int, input().split())
    print("Runner-Up Score:",sorted(list(set(arr)))[-2])
```

```
5
2 3 6 6 5
Runner-Up Score: 5
```

3) Nested Lists.

In [3]:
```
if __name__ == '__main__':
    markslist = []
for i in range(int(input())):
    name = str(input())
    score = float(input())
    markslist.append([name, score])
second_highest = sorted(([score for name, score in markslist]))[1]
print('Second Highest Scores:')
print('\n'.join(sorted([name for name, score in markslist if score == second
```

```
3
alpha
50
beta
50
gamma
100
Second Highest Scores:
alpha
beta
```

4) Finding the Percentage

In [6]:

```python
if __name__ == '__main__':
    n = int(input())
    student_marks = {}
    for i in range(n):
        name, *line = input().split()
        scores = list(map(float, line))
        student_marks[name] = scores
    query_name = input()
op = list(student_marks[query_name])
per = sum(op)/len(op)
print("%.2f" % per)
```

```
2
Harsh 25 26.5 28
Anurag 26 28 30
Harsh
26.50
```

5) Lists

In [6]:
```
if __name__ == '__main__':
    N = int(input())
    answer = []
    for i in range(0,N):
        ip = input().split();
        if ip[0] == "print":
            print(answer)
        elif ip[0] == "insert":
            answer.insert(int(ip[1]),int(ip[2]))
        elif ip[0] == "remove":
            answer.remove(int(ip[1]))
        elif ip[0] == "pop":
            answer.pop()
        elif ip[0] == "append":
            answer.append(int(ip[1]))
        elif ip[0] == "sort":
            answer.sort()
        else:
            answer.reverse()
```

```
12
insert 0 5
insert 1 10
insert 0 6
print
[6, 5, 10]
remove 6
append 9
append 1
sort
print
[1, 5, 9, 10]
pop
reverse
print
[9, 5, 1]
```

6) Tuples

In [8]:
```
if __name__ == '__main__':
    n = int(input())
    integer_list = map(int, input().split())
    print(hash(tuple(integer_list)))
```

```
2
1 2
-3550055125485641917
```

7) Introduction to Sets

In [10]:
```python
from __future__ import division

def average(array):
    # your code goes here
    array = set(array)
    return sum(array) / len(array)

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

```
10
161 182 161 154 176 170 167 171 170 174
169.375
```

8) Symmetric Difference

In [1]:
```python
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 i in sorted(set_m ^ set_n):
    print(i)
```

```
4
2 4 5 9
4
2 9 11 14
4
5
11
14
```

9) set.add()

In [3]:
```python
N = int(input())
country_names = set([])
for i in range(N):
    country_names.add(input())
print(len(country_names))
```

```
5
India
China
Russia
Russia
Germany
4
```

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

In [6]:
```python
n = int(input())
s = set(map(int, input().split()))

for i in range(int(input())):
    option=input().split()
    if option[0]=="pop" :
        s.pop()
    elif option[0]=="remove" :
        s.remove(int(option[1]))
    elif option[0]=="discard" :
        s.discard(int(option[1]))
print(sum(s))
```

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

11) set.union()

In [7]:
```python
n = int(input())
l1 = list(input().split())
m = int(input())
l2 = list(input().split())

s1 = set(l1)
s2 = set(l2)

print(len(s1.union(s2)))
```

```
9
1 2 3 4 5 6 7 8 9
9
10 2 3 1 11 21 55 6 8
13
```

12) set.intersection()

In [8]:
```python
n = int(input())
l1 = list(input().split())
m = int(input())
l2 = list(input().split())

s1 = set(l1)
s2 = set(l2)

print(len(s1.intersection(s2)))
```

```
9
1 2 3 4 5 6 7 8 9
9
10 2 3 2 1 11 21 55 6 8
5
```

13) set.difference()

In [9]:
```python
n = int(input())
l1 = list(input().split())
m = int(input())
l2 = list(input().split())

s1 = set(l1)
s2 = set(l2)

print(len(s1.difference(s2)))
```

```
9
1 2 3 4 5 6 7 8 9
9
10 2 3 2 1 4 5 11 21 9
3
```

14) Set .symmetric_difference() Operation

In [12]:
```python
n = int(input())
l1 = list(input().split())
m = int(input())
l2 = list(input().split())

s1 = set(l1)
s2 = set(l2)

print(len(s1.symmetric_difference(s2)))
```

```
9
1 2 3 4 5 6 7 8 9
9
1 2 3 4 14 15 16 17 19
10
```

15) Set Mutation

In [13]:
```
len_set = int(input())

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

op_len = int(input())

for i in range(op_len):
    operation = input().split()
    if operation[0] == 'intersection_update':
        temp_storage = set(map(int, input().split()))
        storage.intersection_update(temp_storage)
    elif operation[0] == 'update':
        temp_storage = set(map(int, input().split()))
        storage.update(temp_storage)
    elif operation[0] == 'symmetric_difference_update':
        temp_storage = set(map(int, input().split()))
        storage.symmetric_difference_update(temp_storage)
    elif operation[0] == 'difference_update':
        temp_storage = set(map(int, input().split()))
        storage.difference_update(temp_storage)
    else :
        assert False

print(sum(storage))
```

```
16
1 2 3 4 5 6 7 8 9 10 11 12 13 14 24 52
4
intersection_update 10
2 3 5 6 8 9 1 4 7 11
update 2
55 66
symmetric_difference_update 5
22 7 35 62 58
difference_update 7
11 22 35 55 58 62 66
38
```

16) The Captain's Room

In [14]:
```
k,arr = int(input()),list(map(int, input().split()))

my_set = set(arr)

print(((sum(my_set)*k)-(sum(arr)))//(k-1))
```

```
5
1 2 3 6 5 4 4 2 5 3 6 1 6 5 3 2 4 1 2 5 1 4 3 6 8 4 3 1 5 6 2
8
```

17) Check Subset

In [15]:
```python
for i in range(int(input())):
    a = int(input())
    set_a = set(map(int, input().split()))
    b = int(input())
    set_b = set(map(int, input().split()))
    if len(set_a - set_b) == 0:
        print("True")
    else:
        print("False")
```

```
3
5
1 2 3 5 6
9
9 8 7 6 5 3 2 1 7
True
1
2
5
3 6 5 4 1
False
7
1 2 3 4 5 6 7 8
3
9 8 2
False
```

18) Check Strict Superset

In [16]:
```python
def isstrictsuperset(a,b):
    # true if a is a strict superset of b
    return b.issubset(a) and not(a.issubset(b))

a = set(int(x) for x in input().split(' '))
n = int(input())
res = True

for i in range(n):
    b = set(int(x) for x in input().split(' '))
    res &= isstrictsuperset(a,b)

print(res)
```

```
2 3 4 5 6 7 8 9 10 11 12 23 45 84 78 1
2
1 2 3 4 5
100 11 12
False
```

19) No Idea!

In [17]:

```python
if __name__ == "__main__":
    happiness = 0
    n, m = map(int, input().strip().split(' '))
    arr = list(map(int, input().strip().split(' ')))

    good = set(map(int, input().strip().split(' ')))
    bad = set(map(int, input().strip().split(' ')))

    for i in arr:
        if i in good:
            happiness += 1
        elif i in bad:
            happiness -= 1
    print(happiness)
```

```
3 2
1 5 3
3 1
5 7
1
```