Task2 : Datatypes & Sets

Q1) https://www.hackerrank.com/challenges/list-comprehensions?isFullScreen=true

```
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 range(z + 1) if ((i + j + k) != n)])
```

Q2) https://www.hackerrank.com/challenges/nested-list?isFullScreen=true

```
if __name__ == '__main__':
    score_list = {}
    N = int(input("N :"))
    for _ in range(N):
        name = input("Name :")
        score = float(input("Score : "))
        if score in score_list:
            score_list[score].append(name)
        else:
            score_list[score] = [name]

    new_list = []
    for i in score_list:
        new_list.append([i, score_list[i]])

    new_list.sort()
    result = new_list[1][1]
    result.sort()
    print(*result, sep="\n")
```

```
N :2
Name :Ram
Score : 65
Name :Sita
Score : 87
Sita
```

Q3) https://www.hackerrank.com/challenges/find-second-maximum-number-in-a-list?isFullScreen=true

```
if __name__ == '__main__':
    n = int(input("N :"))
    ar = set(map(int, input().split()))
    ar = sorted(ar, key=int, reverse=True)
    ar.pop(0)
    print(ar.pop(0))
```

```
N :5
2 3 4 5 6
5
```

Q4) https://www.hackerrank.com/challenges/python-lists?isFullScreen=true

```
if __name__ == '__main__':
    N = int(input())
    A = []
    for i in range(0, N):
        input_str = input()
        l = input_str.split()
        if l[0] == 'insert':
            A.insert(int(l[1]), int(l[2]))
        elif l[0] == 'print':
            print(A)
        elif l[0] == 'remove':
            A.remove(int(l[1]))
        elif l[0] == 'append':
            A.append(int(l[1]))
        elif l[0] == 'sort':
            A.sort()
        elif l[0] == 'pop':
```

```
            A.pop()
        elif l[0] == 'reverse':
            A.reverse()
```

Q5) https://www.hackerrank.com/challenges/python-tuples?isFullScreen=true

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

Q6) https://www.hackerrank.com/challenges/finding-the-percentage?isFullScreen=true

```
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()
    avg = sum(student_marks[query_name]) / 3
    print("{:.2f}".format(avg))
```

Q7) https://www.hackerrank.com/challenges/py-introduction-to-sets?isFullScreen=true

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

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

Q8) https://www.hackerrank.com/challenges/symmetric-difference?isFullScreen=true

```
M = int( input() )
m_set = set( input().split() )
N = int( input() )
n_set = set( input().split() )

data_list = list(m_set.difference(n_set)) + list(n_set.difference(m_set));
data_list = list( map(int, data_list) )
data_list.sort(reverse=False)

for item in range(0, len(data_list)):
    print( data_list[item] )
```

Q9) https://www.hackerrank.com/challenges/py-set-add?isFullScreen=true

```
def distinctstamps(N):
    stamp = set()
    while N > 0:
        stamp.add( input() )
        N -= 1
    return stamp


if __name__ == '__main__':
    N = int( input() )
    print( len(distinctstamps(N)) )
```

Q10) https://www.hackerrank.com/challenges/py-set-discard-remove-pop?isFullScreen=true

```
n = int(input())
s = set( map(int, input().split()) )
N = int( input() )
```

```
for i in range(0,N):
    inp_data = input().split()
    if inp_data[0] == 'pop':
        s.pop()
    if inp_data[0] == 'remove':
        s.remove( int(inp_data[1]) )
    if inp_data[0] == 'discard':
        s.discard( int(inp_data[1]) )

print(sum(s))
```

Q11) https://www.hackerrank.com/challenges/py-set-union?isFullScreen=true

```
n = int(input())
english = set( map( int, input().split() ) )

p = int(input())
french = set( map( int, input().split() ) )
new_set = english.union(french)
print( len(new_set) )
```

Q12) https://www.hackerrank.com/challenges/py-set-intersection-operation?isFullScreen=true

```
n = int(input())
english = set( map( int, input().split() ) )

p = int(input())
french = set( map( int, input().split() ) )
new_set = english.intersection(french)
print( len(new_set) )
```

Q13) https://www.hackerrank.com/challenges/py-set-difference-operation?isFullScreen=true

```
n = int(input())
english = set( map( int, input().split() ) )

p = int(input())
french = set( map( int, input().split() ) )
new_set = english.difference(french)
print( len(new_set) )
```

Q14) https://www.hackerrank.com/challenges/py-set-symmetric-difference-operation?isFullScreen=true

```
n = int(input())
english = set( map( int, input().split() ) )

p = int(input())
french = set( map( int, input().split() ) )
new_set = english.symmetric_difference(franch)
print( len(new_set) )
```

Q15) https://www.hackerrank.com/challenges/py-set-mutations?isFullScreen=true

```
a = int(input())
A = set( map( int, input().split() ) )
N = int(input())
for i in range(0, N):
    opns = input().split()
    set2 = set( map( int, input().split() ) )
    if opns[0] == 'intersection_update':
        A.intersection_update(set2)
    if opns[0] == 'update':
        A.update(set2)
    if opns[0] == 'symmetric_difference_update':
        A.symmetric_difference_update(set2)
    if opns[0] == 'difference_update':
        A.difference_update(set2)

print(sum(A))
```

Q16) https://www.hackerrank.com/challenges/py-the-captains-room?isFullScreen=true

```
def cap_room(rooms, k):
    set1 = set(rooms)
    ans = (sum(set1)*k) - sum(rooms)
    return (ans // (k-1))


if __name__ == '__main__':
    k = int( input() )
    rooms = list(map(int, input().split()))
    print(cap_room(rooms, k))
```

Q17) https://www.hackerrank.com/challenges/py-check-subset?isFullScreen=true

```
T = int( input() )
for x in range(0,T):
    a = int(input())
    A = set(input().split())
    b = int(input())
    B = set(input().split())

    if len(A.difference(B)):
        print('False')
    else:
        print('True')
```

Q18) https://www.hackerrank.com/challenges/py-check-strict-superset?isFullScreen=true

```
def strictsuperset(A):
    n = int( input() )
    for x in range(0, n):
        sub_set = set(input().split())
        if len(sub_set.difference(A)):
            return False

    return True


if __name__ == '__main__':
    A = set(input().split())
    if strictsuperset(A):
        print('True')
    else:
        print('False')
```

Q19) https://www.hackerrank.com/challenges/no-idea?isFullScreen=true

```
def·happ(arr,·A,·B):
····happiness_score·=·0
····for·i·in·arr:
········if·i·in·A:
············happiness_score·+=·1
········if·i·in·B:
············happiness_score·-=·1
····return·happiness_score

n,·m·=·input().split()
arr·=·list(·map(·int,·input().split()·)·)
A·=·set(·map(·int,·input().split()·)·)
B·=·set(·map(·int,·input().split()·)·)
print(·happ(arr,·A,·B·)·)
```