

DAY 7

1. Rahul likes to play with disks, but due to the lockdown, he decided to give some of his disks to needy children. He is going to distribute N disks of blue color and M disks of yellow color. He has X blue colored disks of cost $a_1, a_2, a_3, \dots, a_x$ respectively, Y yellow-colored disks of cost b_1, b_2, \dots, b_y respectively, and Z disks, having no color, of cost c_1, c_2, \dots, c_z . Before distributing the disk with no color, Rahul can color it blue or yellow, and it will count as a blue or yellow disk, respectively. From these disks Rahul will choose the disks he wants to distribute, and the joy he will gain is the sum of the costs of the disks distributed. Find the maximum possible joy that he can get when he optimally colors zero or more disks.

Constraints:

$$1 \leq N \leq X \leq 10^5$$

$$1 \leq M \leq Y \leq 10^5$$

$$1 \leq Z \leq 10^5$$

$$1 \leq a_i \leq 10^6$$

$$1 \leq b_i \leq 10^6$$

$$1 \leq c_i \leq 10^6$$

All values in input are integers.

Input Format

The first line contains 2 space-separated integers N and M , respectively.

The second line contains 3 space-separated integers X , Y and Z , respectively.

The third line contains x space-separated integers $a_1, a_2, a_3, \dots, a_x$, respectively.

The fourth line contains y space-separated integers $b_1, b_2, b_3, \dots, b_y$, respectively.

The fifth line contains z space-separated integers $c_1, c_2, c_3, \dots, c_z$, respectively.

Output Format

Print the maximum possible joy Rahul can get.

Sample Testcase #0

Testcase Input

1 2

2 2 1

2 4

5 1

3

Testcase Output

12

Explanation

The maximum possible joy is as follows:

distribute the 2-nd blue disk.

Distribute the 1-st yellow disk.

Color the 1-st uncolored disk yellow and distribute it.

Sample Testcase #1

Testcase Input

2 2

2 2 2

8 6

9 1

2 1

Testcase Output

25

Explanation

The maximum possible joy is as follows:

Distribute the 1-nd blue disk.

Distribute the 2-nd blue disk.

Distribute the 1-st yellow disk.

Color the 1-st uncolored disk yellow and distribute it.

CODE :

PYTHON

```
n,m=input().split()
x,y,z=input().split()
n=int(n)
m=int(m)
x=int(x)
y=int(y)
z=int(z)
#Getting the list of input
cx=list(map(int,input().split()))
cy=list(map(int,input().split()))
cz=list(map(int,input().split()))
#sorting each list
cx.sort()
cy.sort()
cz.sort()
#getting top n elements from cx and top m elements from cy
cx_n=cx[x-n:x]
cy_n=cy[y-m:y]
#replacing the minimum element in cx and cy with maximum
element in cz in order to get maximum joy
i=0
j=0
k=z-1
while i<n and j<m and k>=0:
    if cx_n[i]<=cy_n[j]:
        if cx_n[i]<=cz[k]:
            cx_n[i]=cz[k]
            i+=1
            k-=1
        else:
            break
```

```

    else:
        if cy_n[j]<=cz[k]:
            cy_n[j]=cz[k]
            j+=1
            k-=1
        else:
            break
while k>=0 and j<m:
    if cy_n[j]<=cz[k]:
        cy_n[j]=cz[k]
        j+=1
        k-=1
    else:
        break
while k>=0 and i<n:
    if cx_n[i]<=cz[k]:
        cx_n[i]=cz[k]
        i+=1
        k-=1
    else:
        break
#getting maximum joy by adding the elements in cx_n and cy_n
result=0
for i in cx_n:
    result+=int(i)
for j in cy_n:
    result+=int(j)
print(result)

```

2. Amazing Subarrays

You are given a string *S*, and you have to find all the amazing substrings of *S*.

Amazing Substring is one that starts with a vowel (a, e, i, o, u, A, E, I, O, U).

Input

Only argument given is string S.

Output

Return a single integer X mod 10003, here X is number of Amazing Substrings in given string.

Constraints

$1 \leq \text{length}(S) \leq 1e6$

S can have special characters

Example**Input**

ABEC

Output

6

Explanation

Amazing substrings of given string are :

1. A
2. AB
3. ABE
4. ABEC
5. E
6. EC

here number of substrings are 6 and $6 \% 10003 = 6$.

CODE :**PYTHON**

class Solution:

@param A : string

@return an integer

def solve(self, A):

vow=['a','e','i','o','u','A','E','I','O','U']

n=len(A)

result=0

for i in range(n):

```
    if A[i] in vow:  
        result+=(n-1)-i+1  
return result%10003
```