'''

A courier company has to deliver N items with different heights.

All the items have to deliver to one building only.

The building has K rooms inside indexed 0,1,..-K-1, and the heights of the rooms

are different. All the rooms connected with each other and has only one entry point,

e.g, you can reach room-1 from room-0 only, room-2 from room-1 only, ..so on.

Now the task of the delivery agent is to keep the items inside the building.

To keep the items into the rooms, delivery agent has to follow these rules:

- Agent can't keep, two goods in same room.

- As there is only one entry point, If the height of some room is less than

the height of an item, then the item will be stopped before that room,

so are the items behind it.

- Agent can rearrange the order of items to keep them inside the building.

Return the maximum number of items, the delivery agent can keep inside the building.

Input Format:

-------------

Line-1 -> two integers N and K, number of items and number of rooms.

Line-2 -> N space separated integers, heights of the items.

Line-3 -> K space separated integers, heights of the rooms.

Output Format:

--------------

Print an integer as result.

Sample Input-1:

---------------

4 5

4 3 4 1

5 3 3 4 1

Sample Output-1:

----------------

3

Explanation:

------------

We can first keep the item of height 1 in room 4.

Then we can keep the item of height 3 in either of the 3 rooms 1, 2, or 3.

Lastly, we can keep one box of height 4 in room 0.

There is no way we can keep all 4 items in the building.

Sample Input-1:

---------------

5 4

1 2 2 3 4

3 4 1 2

Sample Output-1:

----------------

3

Explanation:

------------

Notice that it's not possible to keep the item of height 4 into the building,

since it cannot pass the first room of height 3.

Also, for the last two rooms, 2 and 3, only items of height 1 can kept.

In room 2 or 3, keep item of height 1.

In room 1, keep item of height 2.

Finally in room 0, keep item of height 3.

We can keep 3 items maximum.

'''

n1,n2=list(map(int,input().split()))

items=list(map(int,input().split()))

rooms=list(map(int,input().split()))

items.sort()

i=n1-1

j=0

count=0

while(i>=0 and j<n2):

if(items[i]<=rooms[j]):

i-=1

j+=1

count+=1

else:

i-=1

print(count)

Students are given a special string consists of only'1' and '2'.

and has to follow the following rules:

The string s is special because concatenating the number of contiguous

occurrences of characters '1' and '2' generates the string s itself.

The first few elements of s is s = "1221121221221121122……". If we group

the consecutive 1's and 2's in s, it will be "1 22 11 2 1 22 1 22 11 2 11 22 ......"

and the occurrences of 1's or 2's in each group are "1 2 2 1 1 2 1 2 2 1 2 2 ......".

You can see that the occurrence sequence is s itself.

Given an integer n, return the number of 1's in the first n number in the special string s.

Input Format

------------

Line1:An integer

Output Format

-------------

An integer represents number of 1's in the first n number

Sample Input-1:

---------------

6

Sample Output-1:

----------------

3

Explanation:

------------

The first 6 elements of special string s is "122112" and

it contains three 1's, so return 3.

Sample Input-2:

---------------

1

Sample Output-2:

----------------

1

#include<iostream>

using namespace std;

int main(){

string s="122";

string p="12";

int n;

cin>>n;

while(s.length()<n){

char h=s[p.length()];

p+=h;

if(h=='1'){

if(s[s.length()-1]=='1'){

s+='2';

}

else{

s+='1';

}

}

else{

if(s[s.length()-1]=='2'){

s+="11";

}

else{

s+="22";

}

}

}

int c=0;

for(int i=0;i<n;i++){

if(s[i]=='1'){

c++;

}

}

cout<<c;

}

You have given a bulb grid, where the bulb which is turned ON is indicated

with 1, and turned OFF is indicated with 0.

You are allowed to perform an operation:

- Select a row/column in the buld grid, and toggle the bulbs,

the bulbs which are turned ON will be truned OFF and the bulbs which are

turned OFF will be turned ON.

Your task is to find the highest possible sum of all the binary equivalents

of each row in the bulb grid, after performing the above operation any

number of times on the bulb grid.

Input Format:

-------------

Line-1: Two space separated integers, M and N

Next M lines: N space separated integers, grid[][]

Output Format:

--------------

Print an integer result.

Sample Input-1:

---------------

3 5

0 1 1 1 1

1 0 1 1 1

0 0 0 0 1

Sample Output-1:

----------------

78

Explanation:

------------

Given grid is

0 1 1 1 1

1 0 1 1 1

0 0 0 0 1

Perform operation on row-0 and row-2

1 0 0 0 0

1 0 1 1 1

1 1 1 1 0

Perform operation on col-1 and col-4

1 1 0 0 1 = 25

1 1 1 1 0 = 30

1 0 1 1 1 = 23

So, now the total value of Binary Equivalent is 78

Sample Input-2:

---------------

2 3

0 1 0

0 0 1

Sample Output-2:

----------------

11

#include<bits/stdc++.h>

using namespace std;

// int f(vector<vector<int>> )

int main(){

int c,r,rr=0;

cin>>r>>c;

vector<vector<int>> grid(r,vector<int>(c));

for(int i=0;i<r;i++){

for(int j=0;j<c;j++){

cin>>grid[i][j];

}

}

for(int i=0;i<r;i++){

if(grid[i][0]==0){

for(int j=0;j<c;j++){

if(grid[i][j]==0){

grid[i][j]=1;

}

else{

grid[i][j]=0;

}

}

}

}

for(int j=0;j<c;j++){

int co=0;

int c1=0;

for(int i=0;i<r;i++){

if(grid[i][j]==0){

co++;

}

else{

c1++;

}

}

if(c1<co){

for(int i=0;i<r;i++){

if(grid[i][j]==0){

grid[i][j]=1;

}

else{

grid[i][j]=0;

}

}

}

}

for(int i=0;i<r;i++){

int sgg=0;

for(int j=0;j<c;j++){

if(grid[i][j]==1){

sgg+=pow(2,c-1-j);

}

// cout<<grid[i][j]<<" ";

}

rr+=sgg;

}

cout<<rr;

}

An 8th standard student has been assigned a task as part of punishment for his mistake.

The task is, there is an input string STR(without any space) and an array of

strings words[]. Print all the pairs of indices [s, e] where s, e are starting

index and ending index of every string in words[] in the input string STR.

Note: Print the pairs[s, e] in sorted order.

(i.e., sort them by their first coordinate, and in case of ties sort them by

their second coordinate).

Input Format

------------

Line-1: string STR(without any space)

Line-2: space separated strings, words[]

Output Format

-------------

Print the pairs[s, e] in sorted order.

Sample Input-1:

---------------

thekmecandngitcolleges

colleges kmec ngit

Sample Output-1:

----------------

3 6

10 13

14 21

Sample Input-2:

---------------

xyxyx

xyx xy

Sample Output-2:

----------------

0 1

0 2

2 3

2 4

Explanation:

------------

Notice that matches can overlap, see "xyx" is found at [0,2] and [2,4].

Sample Input-3:

---------------

kmecngitkmitarecsecolleges

kmit ngit kmec cse

Sample Output-3:

----------------

0 3

4 7

8 11

15 17