Codeforces Round #710 (Div. 3)

A. Strange Table

A. Strange Table

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Polycarp found a rectangular table consisting of n rows and m columns. He noticed that each cell of the table has its number, obtained by the following algorithm **"by columns"**:

- cells are numbered starting from one;
- cells are numbered from left to right by columns, and inside each column from top to bottom;
- number of each cell is an integer one greater than in the previous cell.

For example, if $n = 3$ and $m = 5$, the table will be numbered as follows:

$$\begin{matrix} 1 & 4 & 7 & 10 & 13 \\ 2 & 5 & 8 & 11 & 14 \\ 3 & 6 & 9 & 12 & 15 \end{matrix}$$

However, Polycarp considers such numbering inconvenient. He likes the numbering **"by rows"**:

- cells are numbered starting from one;
- cells are numbered from top to bottom by rows, and inside each row from left to right;
- number of each cell is an integer one greater than the number of the previous cell.

For example, if $n = 3$ and $m = 5$, then Polycarp likes the following table numbering:

$$\begin{matrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{matrix}$$

Polycarp doesn't have much time, so he asks you to find out what would be the cell number in the numbering **"by rows"**, if in the numbering **"by columns"** the cell has the number x?

Input

The first line contains a single integer t ($1 \le t \le 10^4$). Then t test cases follow.

Each test case consists of a single line containing three integers n, m, x ($1 \le n, m \le 10^6$, $1 \le x \le n \cdot m$), where n and m are the number of rows and columns in the table, and x is the cell number.

Note that the numbers in some test cases do not fit into the 32-bit integer type, so you must use at least the 64-bit integer type of your programming language.

Output

For each test case, output the cell number in the numbering **"by rows"**.

Example

input Copy

```
5
1 1 1
2 2 3
3 5 11
100 100 7312
1000000 1000000 1000000000000
```

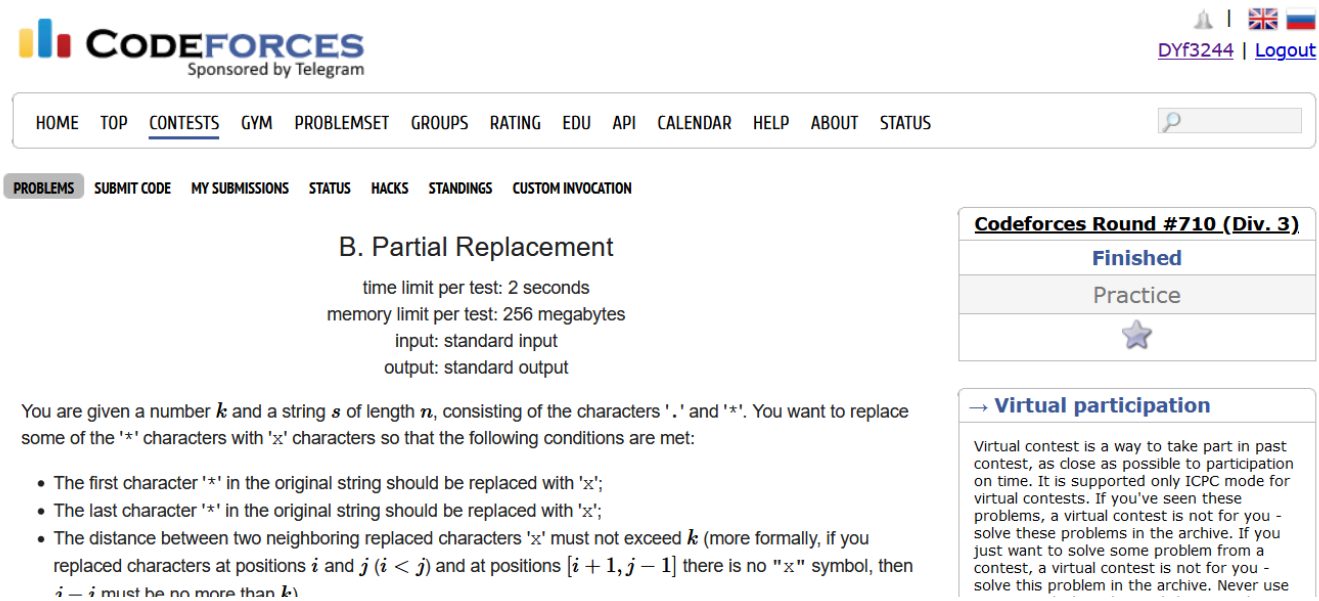
output Copy

```
1
2
9
1174
1000000000000
```

```cpp
#include <bits/stdc++.h>
using namespace std;
#define IOS ios::sync_with_stdio(0)
#define LL long long
#define maxn (int)(2e5 + 10)


LL ans[maxn];
int main ()
{
    IOS;
    int T ; cin >> T;
    for ( int cas = 1 ; cas <= T ; cas++)
    {
        LL n , m , x ;
        cin >> n >> m >> x ;
        LL col = x / n ;
        LL row = x % n ;
        if ( row ) col ++ ;
        if ( !row ) row = n ;
        ans[cas] = ( row - 1 ) * m + col ;
    }
    for ( int i = 1 ; i <= T ; i++ )
        cout << ans[i] << endl;
}
```

B. Partial Replacement



For example, if $n = 7$, $s =$ `.**.***` and $k = 3$, then the following strings will satisfy the conditions above:

- `.xx.*xx`;
- `.x*.x*x`;
- `.xx.xxx`.

But, for example, the following strings will not meet the conditions:

- `.**.*xx` (the first character '*' should be replaced with 'x');
- `.x*.xx*` (the last character '*' should be replaced with 'x');
- `.x*.*xx` (the distance between characters at positions 2 and 6 is greater than $k = 3$).

Given n, k, and s, find the minimum number of '*' characters that must be replaced with 'x' in order to meet the above conditions.

Input

The first line contains one integer t ($1 \le t \le 500$). Then t test cases follow.

The first line of each test case contains two integers n and k ($1 \le k \le n \le 50$).

The second line of each test case contains a string s of length n, consisting of the characters '.' and '*'.

It is guaranteed that there is at least one '*' in the string s.

It is guaranteed that the distance between any two neighboring '*' characters does not exceed k.

Output

For each test case output the minimum number of '*' characters that must be replaced with 'x' characters in order to satisfy the conditions above.

Example

input Copy

```
5
7 3
.**.***
5 1
..*..
5 2
*.*.*
3 2
*.*
1 1
*
```

output Copy

```
3
1
3
2
1
```

someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Practice

You are registered for practice. You can solve problems unofficially. Results can be found in the contest status and in the bottom of standings.

→ Clone Contest to Mashup

You can clone this contest to a mashup.

Clone Contest

→ Submit?

Language: GNU G++11 5.1.0

Choose file: 浏览... 未选择文件。

Submit

→ Last submissions

Submission	Time	Verdict
111123528	Mar/26/2021 13:40	Accepted

→ Problem tags

greedy implementation *1100

No tag edit access

→ Contest materials

- Announcement
- Tutorial


```cpp
#include <bits/stdc++.h>
using namespace std;
#define IOS ios::sync_with_stdio(0)
#define LL long long
#define maxn (int)(2e5 + 10)

int ans[maxn];

int main ()
{
    IOS;
```

```
    int T ; cin >> T;
    for ( int cas = 1 ; cas <= T ; cas++)
    {
        int n , k; cin >> n >> k ;
        string s ; cin >> s;
        vector <int> v;
        for ( int i = 0 ; i < n ; i++ )
            if ( s[i] == '*' )
                v.push_back(i);
        int cnt = 0 ;
        for ( int i = 0 ; i < v.size() ; i++ )
        {
            int x = v[i];
            cnt++;
            while ( i < v.size() && v[i] - x <= k )
                i++;
            if ( i == v.size() ) {
                if ( i != 1 )
                    cnt++ ;
                break;
            }
            i -= 2;
        }
        ans[cas] = cnt;
    }
    for ( int i = 1 ; i <= T ; i++ )
        cout << ans[i] << endl;
}
```

C. Double-ended Strings

CODEFORCES
Sponsored by Telegram

DYf3244 | Logout

HOME TOP CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP ABOUT STATUS

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS HACKS STANDINGS CUSTOM INVOCATION

C. Double-ended Strings

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given the strings a and b, consisting of lowercase Latin letters. You can do any number of the following operations in any order:

- if $|a| > 0$ (the length of the string a is greater than zero), delete the first character of the string a, that is, replace a with $a_2a_3 \dots a_n$;
- if $|a| > 0$, delete the last character of the string a, that is, replace a with $a_1a_2 \dots a_{n-1}$;
- if $|b| > 0$ (the length of the string b is greater than zero), delete the first character of the string b, that is, replace b with $b_2b_3 \dots b_n$;
- if $|b| > 0$, delete the last character of the string b, that is, replace b with $b_1b_2 \dots b_{n-1}$.

Note that after each of the operations, the string a or b may become empty.

For example, if $a =$ "hello" and $b =$ "icpc", then you can apply the following sequence of operations:

- delete the first character of the string $a \Rightarrow a =$ "ello" and $b =$ "icpc";
- delete the first character of the string $b \Rightarrow a =$ "ello" and $b =$ "cpc";
- delete the first character of the string $b \Rightarrow a =$ "ello" and $b =$ "pc";

Codeforces Round #710 (Div. 3)
Finished
Practice

→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Practice

You are registered for practice. You can solve problems unofficially. Results can be found in

- delete the last character of the string $a \Rightarrow a =$"ell" and $b =$"pc";
- delete the last character of the string $b \Rightarrow a =$"ell" and $b =$"p".

For the given strings a and b, find the minimum number of operations for which you can make the strings a and b equal. Note that empty strings are also equal.

Input
The first line contains a single integer t ($1 \le t \le 100$). Then t test cases follow.

The first line of each test case contains the string a ($1 \le |a| \le 20$), consisting of lowercase Latin letters.

The second line of each test case contains the string b ($1 \le |b| \le 20$), consisting of lowercase Latin letters.

Output
For each test case, output the minimum number of operations that can make the strings a and b equal.

Example

input Copy

```
5
a
a
abcd
bc
hello
codeforces
hello
helo
dhjakjsnasjhfksafasd
adjsnasjhfksvdafdser
```

output Copy

```
0
2
13
3
20
```

the contest status and in the bottom of standings.

→ **Clone Contest to Mashup**

You can clone this contest to a mashup.

Clone Contest

→ **Submit?**

Language: GNU G++11 5.1.0

Choose file: 浏览... 未选择文件。

Submit

→ **Last submissions**

Submission	Time	Verdict
111124489	Mar/26/2021 13:52	Accepted

→ **Problem tags**

brute force dp hashing strings *1000

No tag edit access

→ **Contest materials**

- Announcement
- Tutorial


```cpp
#include <bits/stdc++.h>
using namespace std;
#define IOS ios::sync_with_stdio(0)
#define LL long long
#define maxn (int)(1e6 + 10)

int ans[105];

int main  ()
{
    IOS;
    int T ; cin >> T;
    for ( int cas = 1 ; cas <= T ; cas++)
    {
        string a , b ;
        cin >> a >> b;
        int n = a.length() , m = b.length();
        a = ' ' + a ; b = ' ' + b ;
        int maxx = 0 ;
```

```
        for ( int i = 1 ; i <= n ; i++ )
        {
            for ( int j = 1 ; j <= m ; j++ )
            {
                int cnt = 0 ;
                if ( a[i] != b[j] ) continue;
                int w = i ;
                int k = j ;
                while ( w <= n && k <= m && a[w] == b[k] )
                {
                    cnt++;
                    w++ ; k++;
                }
                maxx = max ( maxx , cnt );
            }
        }
        ans[cas] = n + m - 2*maxx ;
    }
    for ( int i = 1 ; i <= T ; i++ )
        cout << ans[i] << endl;
}
```

D. Epic Transformation

CODEFORCES
Sponsored by Telegram

DYf3244 | Logout

HOME TOP CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP ABOUT STATUS

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS HACKS STANDINGS CUSTOM INVOCATION

D. Epic Transformation

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given an array a of length n consisting of integers. You can apply the following operation, consisting of several steps, on the array a **zero** or more times:

- you select two **different** numbers in the array a_i and a_j;
- you remove i-th and j-th elements from the array.

For example, if $n = 6$ and $a = [1, 6, 1, 1, 4, 4]$, then you can perform the following sequence of operations:

- select $i = 1, j = 5$. The array a becomes equal to $[6, 1, 1, 4]$;
- select $i = 1, j = 2$. The array a becomes equal to $[1, 4]$.

What can be the minimum size of the array after applying some sequence of operations to it?

Input

The first line contains a single integer t ($1 \le t \le 10^4$). Then t test cases follow.

The first line of each test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$) is length of the array a.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output the minimum possible size of the array after applying some sequence of operations to it.

Example

input Copy

```
5
6
```

Codeforces Round #710 (Div. 3)
Finished
Practice

→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Practice

You are registered for practice. You can solve problems unofficially. Results can be found in the contest status and in the bottom of standings.

→ Clone Contest to Mashup

You can clone this contest to a mashup.

Clone Contest

→ Submit?

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

output Copy

```
0
0
2
1
0
```

Language: GNU G++11 5.1.0

Choose file: 浏览... 未选择文件。

Submit

→ **Last submissions**

Submission	Time	Verdict
111133261	Mar/26/2021 15:32	Accepted
111127198	Mar/26/2021 14:25	Wrong answer on test 2
111126454	Mar/26/2021 14:15	Wrong answer on test 2

→ **Problem tags**

constructive algorithms | data structures | greedy | *1400

No tag edit access

→ **Contest materials**

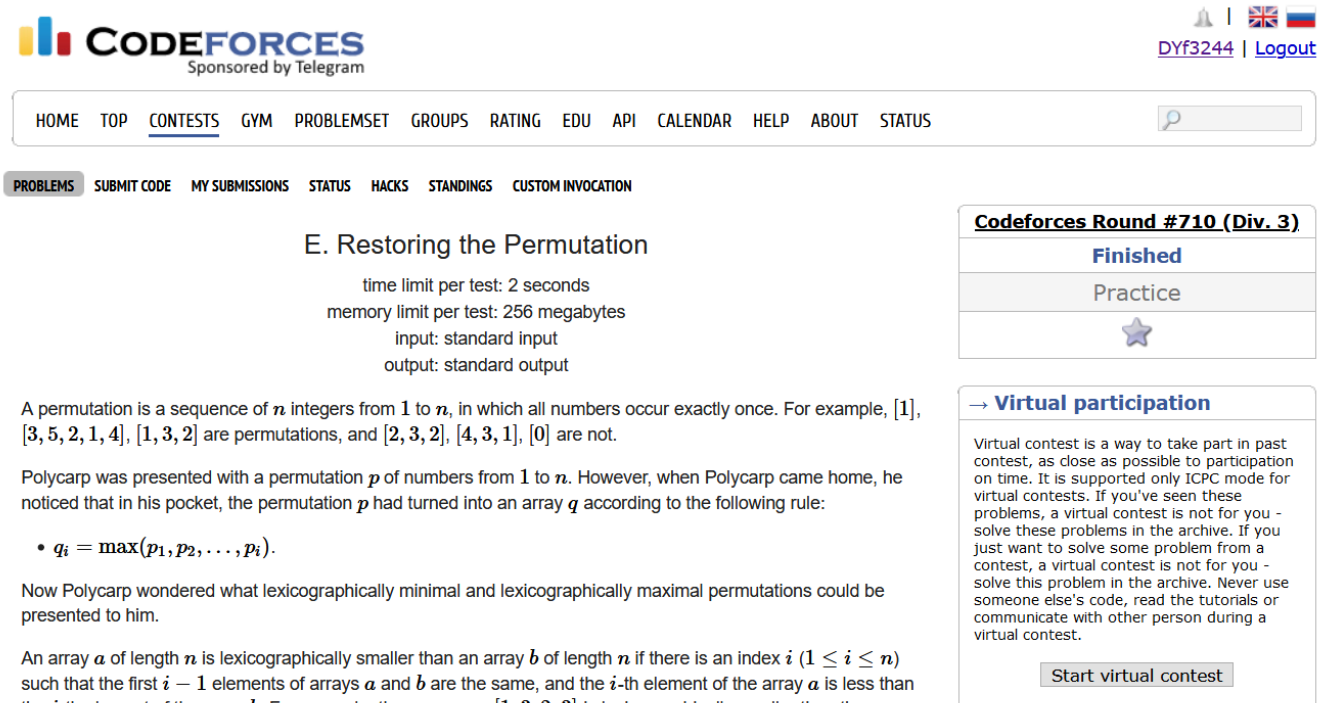
- Announcement
- Tutorial


```cpp
#include <bits/stdc++.h>
using namespace std;
#define IOS ios::sync_with_stdio(0)
#define LL long long
#define maxn (int)(2e5 + 10)


int ans[maxn];
int a[maxn];
int main()
{
    IOS;
    int T ; cin >> T;
    for ( int cas = 1 ; cas <= T ; cas++)
    {
        int n ; cin >> n;
        for ( int i = 1; i <= n ; i++ ) cin >> a[i];
        sort ( a+1 , a+1+n );
        vector<int> v;
        for ( int i = 1; i <= n ; i++ )
        {
            int cnt = 0 ;
            int x = a[i];
```

```
            while ( i <= n && a[i] == x )
            {
                cnt ++;
                i++;
            }
            i--;
            v.push_back(cnt);
        }
        priority_queue<int> q;
        for ( int x : v )
            q.push(x);
        while ( q.size() >= 2 )
        {
            int cnt1 = q.top();
            q.pop();
            int cnt2 = q.top();
            q.pop();
            cnt1-- ; cnt2-- ;
            if ( cnt1 )
                q.push(cnt1);
            if ( cnt2 )
                q.push(cnt2);
        }
        if ( q.empty() )
            ans[cas] = 0 ;
        else
            ans[cas] = q.top();
    }
    for ( int i = 1 ; i <= T ; i++ )
        cout << ans[i] << endl;
}
```

E. Restoring the Permutation



the i-th element of the array b. For example, the array $a = [1, 3, 2, 3]$ is lexicographically smaller than the array $b = [1, 3, 4, 2]$.

For example, if $n = 7$ and $p = [3, 2, 4, 1, 7, 5, 6]$, then $q = [3, 3, 4, 4, 7, 7, 7]$ and the following permutations could have been as p initially:

- $[3, 1, 4, 2, 7, 5, 6]$ (lexicographically minimal permutation);
- $[3, 1, 4, 2, 7, 6, 5]$;
- $[3, 2, 4, 1, 7, 5, 6]$;
- $[3, 2, 4, 1, 7, 6, 5]$ (lexicographically maximum permutation).

For a given array q, find the lexicographically minimal and lexicographically maximal permutations that could have been originally presented to Polycarp.

Input

The first line contains one integer t ($1 \le t \le 10^4$). Then t test cases follow.

The first line of each test case contains one integer n ($1 \le n \le 2 \cdot 10^5$).

The second line of each test case contains n integers $q_1, q_2, \ldots, q_n$ ($1 \le q_i \le n$).

It is guaranteed that the array q was obtained by applying the rule from the statement to some permutation p.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output two lines:

- on the first line output n integers — lexicographically **minimal** permutation that could have been originally presented to Polycarp;
- on the second line print n integers — lexicographically **maximal** permutation that could have been originally presented to Polycarp;

Example

input Copy

```
4
7
3 3 4 4 7 7 7
4
1 2 3 4
7
3 4 5 5 5 7 7
1
1
```

output Copy

```
3 1 4 2 7 5 6
3 2 4 1 7 6 5
1 2 3 4
1 2 3 4
3 4 5 1 2 7 6
3 4 5 2 1 7 6
1
1
```

→ **Practice**

You are registered for practice. You can solve problems unofficially. Results can be found in the contest status and in the bottom of standings.

→ **Clone Contest to Mashup**

You can clone this contest to a mashup.

Clone Contest

→ **Submit?**

Language: GNU G++11 5.1.0

Choose file: 浏览... 未选择文件。

Submit

→ **Last submissions**

Submission	Time	Verdict
111133904	Mar/26/2021 15:39	Accepted
111132247	Mar/26/2021 15:21	Wrong answer on test 2
111130457	Mar/26/2021 15:02	Time limit exceeded on test 10

→ **Problem tags**

constructive algorithms | data structures | dsu | greedy | *1500

No tag edit access

→ **Contest materials**

- Announcement
- Tutorial


```
#include <bits/stdc++.h>
using namespace std;
#define IOS ios::sync_with_stdio(0)
#define LL long long
#define maxn (int)(2e5 + 10)

int T ;
int n ;
int a[maxn];
```

```
vector<int> ans[maxn];
bool Hash[maxn];
void get_min( int cas )
{
    int x = 0 ;
    int cnt = 1 ;
    for ( int i = 1 ; i <= n ; i++ )
    {
        x = a[i];
        if (!Hash[x]) ans[cas].push_back(x) , Hash[x] = 1;
        else
        {
            while (Hash[cnt]) cnt++;
            ans[cas].push_back(cnt++);
        }
    }
}
void get_max ( int cas )
{
    int x = 0 ;
    stack<int> st;
    for ( int i = 1 ; i <= n ; i++ )
    {
        if ( x != a[i] )
        {
            x = a[i];
            if ( x > 1 && !Hash[x-1] )
                st.push(x-1);
            ans[cas].push_back(x);
            Hash[x] = 1;
        }
        else
        {
            int &cnt = st.top();
            Hash[cnt] = 1 ;
            ans[cas].push_back(cnt--);
            if ( Hash[cnt] || cnt == 0 ) st.pop();
        }
    }
}
int main ()
{
    IOS;
    cin >> T;
    for ( int cas = 1 ; cas <= T ; cas++ )
    {
        cin >> n ;
        for ( int i = 1 ; i <= n ; i++ ) cin >> a[i];
        ans[cas].push_back(0);
        memset ( Hash , 0 , sizeof(bool)*(n+5) );
        get_min(cas);
        memset ( Hash , 0 , sizeof(bool)*(n+5) );
        get_max(cas);
    }
```

```
    for ( int i = 1 ; i <= T ; i++ )
    {
        int tmp = (ans[i].size()-1) / 2;
        for ( int j = 1 ; j <= tmp ; j++ )
            cout << ans[i][j] << " ";
        cout << endl;
        for ( int j = tmp+1 ; j <= 2*tmp ; j++ )
            cout << ans[i][j] << " ";
        cout << endl;
    }
}
```