Gracias Sponsors!

Organizador

UNR

Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

FCEIA

Diamond

GTS

Gold

Avature

accenture

Contest 7 - Avanzado

A

Bandits

time limit per test: 5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

There is a kingdom with N villages and $N-1$ bidirectional roads that allow the citizens to travel between any pair of villages following a path consisting of one or more roads. The i-th road connects villages A_i and B_i and has length C_i.

The king has noticed an increasing number of complaints about bandits attacking the merchants travelling along the roads in the kingdom. He has tasked his advisor with solving this problem by hiring loyal groups of thugs that will act as security agencies. Each such security contract guarantees security of all roads in a radius of R_j from the village X_j with the group's headquarters. A road is protected by the contract if it is part of a path of length at most R_j from X_j to some other village. Some roads may be protected by several contracts and are therefore more secure.

Write a program that will process queries about new contracts and answer queries about the security of individual roads, that is the number of contracts currently securing that road.

Input

The first line contains the number of villages N. The roads connecting these villages are described in the following $N-1$ lines. The description of each road consists of space-separated integers A_i, B_i and C_i, which represent a road of length C_i between villages A_i and B_i. The villages are numbered from 1 to N.

Next line contains the number of queries Q. The following Q lines describe the queries. The query that represents a new security contract starts with character "`.`" and is followed by the headquarters village X_j and security radius R_j. The query about the security of some road starts with character "?" and is followed by the number Y_j of that road. The roads are numbered from 1 to $N-1$ in order in which they are given in the input.

Input limits

- $1 \leq N, Q \leq 10^5$
- $0 \leq C_i, R_j \leq 10^9$

Output

Process the queries in the given order and for every query of type "?" output one line with the current number of contracts securing the road Y_j.

Example

input Copy

```
7
1 2 4
4 2 7
5 1 3
3 6 4
1 6 9
2 7 1
7
+ 2 6
? 3
? 1
+ 6 14
? 1
? 2
? 3
```

output Copy

```
0
1
2
0
1
```

C

Impartial Strings

time limit per test: 4 seconds
memory limit per test: 1024 megabytes
input: standard input
output: standard output

Alice builds machines that generates strings. Alice's machines each consist of N states, numbered from 1 to N, and a set of directed edges between these states, each labelled with a character from a fixed set. A subset of the states are "final" states. The machine generates strings by starting at state 1, following a path that terminates at a final state, and concatenating the characters of the edge labels together in the order that the edges are traversed. The path is allowed to visit the same state more than once and can traverse the same edge more than once. The path can pass through final states before eventually terminating at a final state. Self loops are allowed and having two or more edges to and/or from the same state labelled with the same letter is also allowed.

Bob has a favorite string S. Carol has a favorite string T. Alice wonders if she can build a machine that can generate exactly the strings that have an equal number of occurrences of S and T as substrings. That is, the machine should generate every string that has an equal number of occurrences of S and T as substrings and it should not generate any strings that do not satisfy this property. Occurrences may overlap. For example, the string `banana` has two occurrences of the substring `ana`. Help Alice determine if it is possible to complete the task for Bob and Carol's favorite strings.

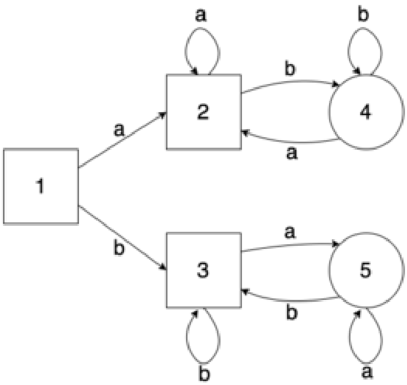


Figure 1: Example machine for the first case in the sample input.

Figure 1 gives an example machine for the first case in the sample input. The square states represent the final states.

Input

The first line of input contains a single positive integer K $(1 \leq K \leq 50)$, the number of test cases. This is followed by K lines each containing three strings A, S, T. The first string, A, is the fixed set of characters used in the machine. The characters in A are distinct lowercase english letters. The second string, S, is Bob's favorite string. The third string, T, is Carol's favorite string. The lengths of S and T satisfy $1 \leq |S|, |T| \leq 500$. It is guaranteed that the distinct characters in S and T are a subset of those in A.

Output

Output one line for each test case. Output 1 if Alice can build a machine as described. Otherwise, output 0.

Example

input Copy

```
3
ab ab ba
abc ab ba
cz cczz zzcc
```

output Copy

```
1
0
0
```

C

Impartial Strings

time limit per test: 4 seconds
memory limit per test: 1024 megabytes
input: standard input
output: standard output

Alice builds machines that generates strings. Alice's machines each consist of N states, numbered from 1 to N, and a set of directed edges between these states, each labelled with a character from a fixed set. A subset of the states are "final" states. The machine generates strings by starting at state 1, following a path that terminates at a final state, and concatenating the characters of the edge labels together in the order that the edges are traversed. The path is allowed to visit the same state more than once and can traverse the same edge more than once. The path can pass through final states before eventually terminating at a final state. Self loops are allowed and having two or more edges to and/or from the same state labelled with the same letter is also allowed.

Bob has a favorite string S. Carol has a favorite string T. Alice wonders if she can build a machine that can generate exactly the strings that have an equal number of occurrences of S and T as substrings. That is, the machine should generate every string that has an equal number of occurrences of S and T as substrings and it should not generate any strings that do not satisfy this property. Occurrences may overlap. For example, the string `banana` has two occurrences of the substring `ana`. Help Alice determine if it is possible to complete the task for Bob and Carol's favorite strings.

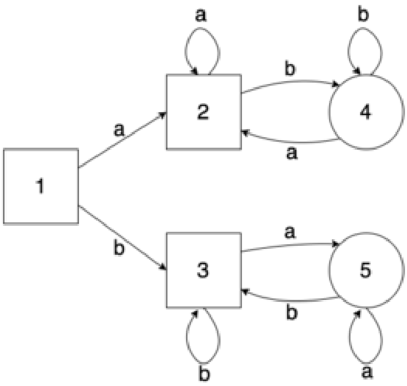


Figure 1: Example machine for the first case in the sample input.

Figure 1 gives an example machine for the first case in the sample input. The square states represent the final states.

Input

The first line of input contains a single positive integer K $(1 \leq K \leq 50)$, the number of test cases. This is followed by K lines each containing three strings A, S, T. The first string, A, is the fixed set of characters used in the machine. The characters in A are distinct lowercase english letters. The second string, S, is Bob's favorite string. The third string, T, is Carol's favorite string. The lengths of S and T satisfy $1 \leq |S|, |T| \leq 500$. It is guaranteed that the distinct characters in S and T are a subset of those in A.

Output

Output one line for each test case. Output 1 if Alice can build a machine as described. Otherwise, output 0.

Example

input Copy

```
3
ab ab ba
abc ab ba
cz cczz zzcc
```

output Copy

```
1
0
0
```

D

Irrigation

time limit per test: 2.5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Misha was interested in water delivery from childhood. That's why his mother sent him to the annual Innovative Olympiad in Irrigation (IOI). Pupils from all Berland compete there demonstrating their skills in watering. It is extremely expensive to host such an olympiad, so after the first n olympiads the organizers introduced the following rule of the host city selection.

The host cities of the olympiads are selected in the following way. There are m cities in Berland wishing to host the olympiad, they are numbered from 1 to m. The host city of each next olympiad is determined as the city that hosted the olympiad the **smallest** number of times before. If there are several such cities, the city with the **smallest** index is selected among them.

Misha's mother is interested where the olympiad will be held in some specific years. The only information she knows is the above selection rule and the host cities of the first n olympiads. Help her and if you succeed, she will ask Misha to avoid flooding your house.

Input

The first line contains three integers n, m and q ($1 \le n, m, q \le 500\,000$) — the number of olympiads before the rule was introduced, the number of cities in Berland wishing to host the olympiad, and the number of years Misha's mother is interested in, respectively.

The next line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le m$), where a_i denotes the city which hosted the olympiad in the i-th year. Note that before the rule was introduced the host city was chosen arbitrarily.

Each of the next q lines contains an integer k_i ($n + 1 \le k_i \le 10^{18}$) — the year number Misha's mother is interested in host city in.

Output

Print q integers. The i-th of them should be the city the olympiad will be hosted in the year k_i.

Examples

input Copy

```
6 4 10
3 1 1 1 2 2
7
8
9
10
11
12
13
14
15
16
```

output Copy

```
4
3
4
2
3
4
1
2
```

```
3
4
```

input Copy

```
4 5 4
4 4 5 1
15
9
13
6
```

output Copy

```
5
3
3
3
```

Note

In the first example Misha's mother is interested in the first 10 years after the rule was introduced. The host cities these years are 4, 3, 4, 2, 3, 4, 1, 2, 3, 4.

In the second example the host cities after the new city is introduced are 2, **3**, 1, 2, **3**, 5, 1, 2, **3**, 4, **5**, 1.

E

Mirror in the String

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have a string $s_1 s_2 \dots s_n$ and you stand on the left of the string looking right. You want to choose an index k ($1 \le k \le n$) and place a mirror after the k-th letter, so that what you see is $s_1 s_2 \dots s_k s_k s_{k-1} \dots s_1$. What is the lexicographically smallest string you can see?

A string a is lexicographically smaller than a string b if and only if one of the following holds:

- a is a prefix of b, but $a \ne b$;
- in the first position where a and b differ, the string a has a letter that appears earlier in the alphabet than the corresponding letter in b.

Input

The first line of input contains one integer t ($1 \le t \le 10\,000$): the number of test cases.

The next t lines contain the description of the test cases, two lines per a test case.

In the first line you are given one integer n ($1 \le n \le 10^5$): the length of the string.

The second line contains the string s consisting of n lowercase English characters.

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

Output

For each test case print the lexicographically smallest string you can see.

Example

input Copy

```
4
10
codeforces
9
cbacbacba
3
aaa
4
bbaa
```

output Copy

```
cc
cbaabc
aa
bb
```

Note

In the first test case choose $k = 1$ to obtain "cc".

In the second test case choose $k = 3$ to obtain "cbaabc".

In the third test case choose $k = 1$ to obtain "aa".

In the fourth test case choose $k = 1$ to obtain "bb".

F

Differences

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

We have a list of N strings S_i. All strings have length M and consist only of characters A, B, C and D. Let us define the distance between two strings X and Y as the number of indices j, where the strings have different characters ($X_j \neq Y_j$). We know that the list of strings S_i contains precisely one special string that has distance K to all other strings. Note that there might be other pairs of strings with a distance of K. We are experiencing problems finding this special string, so please write a program to help us out.

Input

The first line contains space-separated integers N, M and K. Strings S_i are given in the following N lines.

Input limits

- $2 \leq N, M \leq 10^5$
- $1 \leq K \leq M$
- $NM \leq 2 \cdot 10^7$

Output

Output the index i of the special string. Strings are numbered from 1 to N as given in the input.

Examples

input Copy

```
5 10 2
DCDDDCCADA
ACADDCCADA
DBADDCCBDC
DBADDCCADA
ABADDCCADC
```

output Copy

```
4
```

input Copy

```
4 6 5
AABAAA
BAABBB
ABAAAA
ABBAAB
```

output Copy

```
2
```

G

New Year and Curling

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

Carol is currently curling.

She has n disks each with radius r on the 2D plane.

Initially she has all these disks above the line $y = 10^{100}$.

She then will slide the disks towards the line $y = 0$ one by one in order from 1 to n.

When she slides the i-th disk, she will place its center at the point $(x_i, 10^{100})$. She will then push it so the disk's y coordinate continuously decreases, and x coordinate stays constant. The disk stops once it touches the line $y = 0$ or it touches any previous disk. Note that once a disk stops moving, it will not move again, even if hit by another disk.

Compute the y-coordinates of centers of all the disks after all disks have been pushed.

Input

The first line will contain two integers n and r ($1 \le n, r \le 1\,000$), the number of disks, and the radius of the disks, respectively.

The next line will contain n integers $x_1, x_2, ..., x_n$ ($1 \le x_i \le 1\,000$) — the x-coordinates of the disks.

Output

Print a single line with n numbers. The i-th number denotes the y-coordinate of the center of the i-th disk. The output will be accepted if it has absolute or relative error at most 10^{-6}.

Namely, let's assume that your answer for a particular value of a coordinate is a and the answer of the jury is b. The checker program will consider your answer correct if $\frac{|a-b|}{\max(1, b)} \le 10^{-6}$ for all coordinates.

Example

input Copy

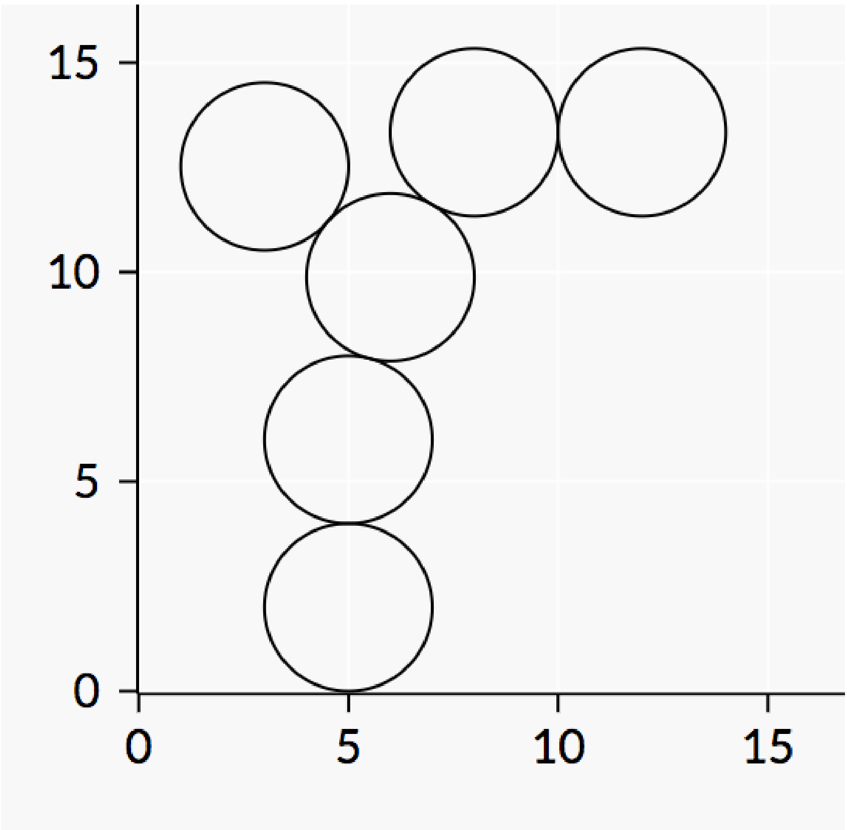
```
6 2
5 5 6 8 3 12
```

output Copy

```
2 6.0 9.87298334621 13.3370849613 12.5187346573 13.3370849613
```

Note

The final positions of the disks will look as follows:



In particular, note the position of the last disk.

H

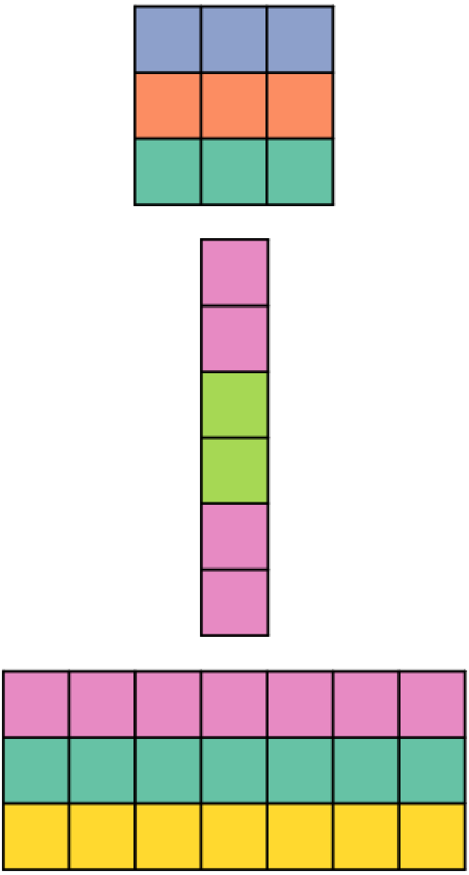
Flag

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

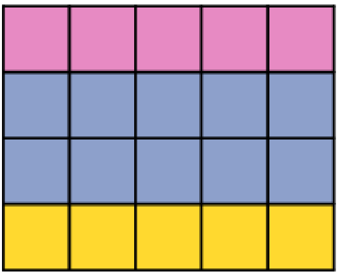
Innokenty works at a flea market and sells some ~~random stuff~~ rare items. Recently he found an old rectangular blanket. It turned out that the blanket is split in $n \cdot m$ colored pieces that form a rectangle with n rows and m columns.

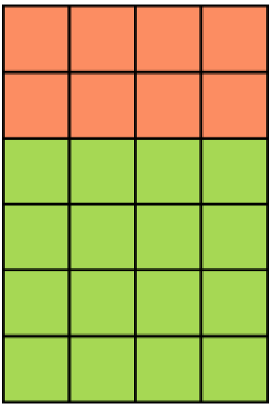
The colored pieces attracted Innokenty's attention so he immediately came up with the following business plan. If he cuts out a subrectangle consisting of three colored stripes, he can sell it as a flag of some country. Innokenty decided that a subrectangle is similar enough to a flag of some country if it consists of three stripes of **equal** heights placed one above another, where each stripe consists of cells of equal color. Of course, the color of the top stripe must be different from the color of the middle stripe; and the color of the middle stripe must be different from the color of the bottom stripe.

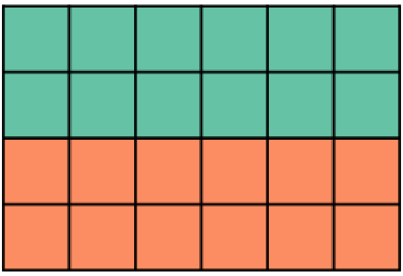
Innokenty has not yet decided what part he will cut out, but he is sure that the flag's boundaries should go along grid lines. Also, Innokenty won't rotate the blanket. Please help Innokenty and count the number of different subrectangles Innokenty can cut out and sell as a flag. Two subrectangles located in different places but forming the same flag are still considered different.

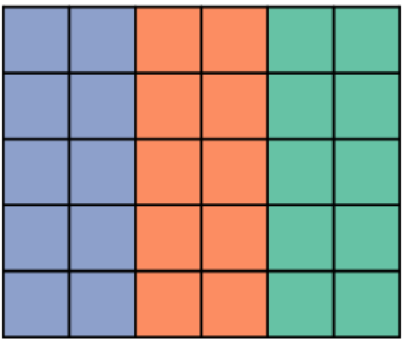


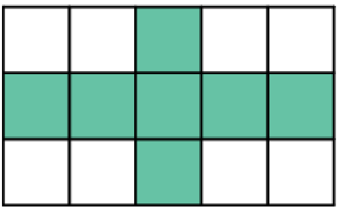
These subrectangles are flags.

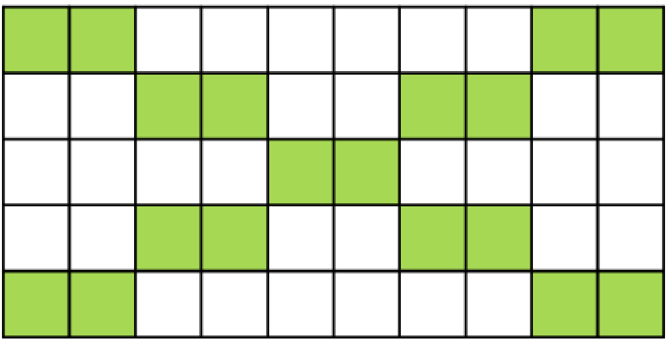












These subrectangles are not flags.

Input

The first line contains two integers n and m ($1 \le n, m \le 1\,000$) — the number of rows and the number of columns on the blanket.

Each of the next n lines contains m lowercase English letters from 'a' to 'z' and describes a row of the blanket. Equal letters correspond to equal colors, different letters correspond to different colors.

Output

In the only line print the number of subrectangles which form valid flags.

Examples

input Copy

```
4 3
aaa
bbb
ccb
ddd
```

output Copy

```
6
```

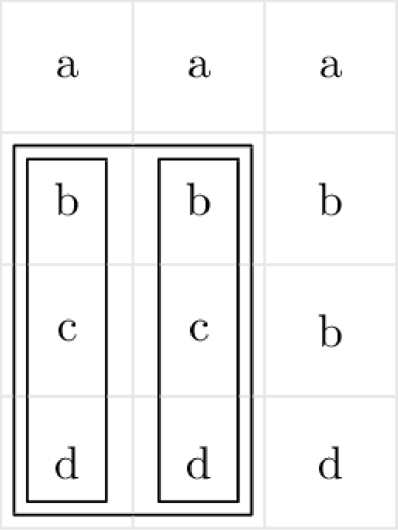
input Copy

```
6 1
a
a
b
b
c
c
```

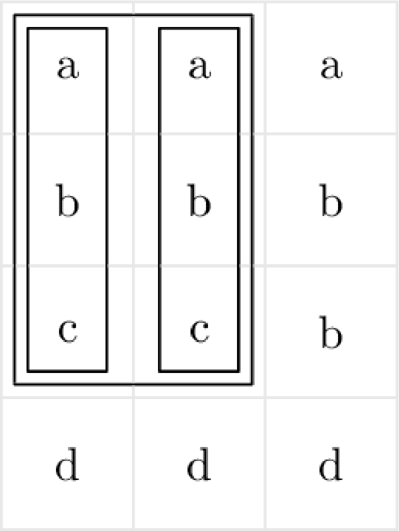
output Copy

```
1
```

Note



a	a	a
b	b	b
c	c	b
d	d	d



a	a	a
b	b	b
c	c	b
d	d	d

The selected subrectangles are flags in the first example.

I

Vera and ABCDE

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

When Vera was learning English, she had 5 types of toy blocks, one for each of the first 5 uppercase letters. A block can be represented as a 5×3 grid with characters `.` and `*`. The 5 types of blocks look like the following:

```
*** *** *** *** ***
*.* *.* *.. *.* *..
*** *** *.. *.* ***
*.* *.* *.. *.* *..
*.* *** *** *** ***
```

Vera has a word S with N letters and wonders what the corresponding sequence of blocks look like when arranged in a row.

Input

Line 1 contains integer N ($1 \le N \le 2017$).

Line 2 contains string S of length N, and consists of only letters `A`, `B`, `C`, `D`, `E`.

Output

Print 5 lines, each with $3N$ characters, the corresponding sequence of blocks.

Examples

input Copy

```
5
ABCDE
```

output Copy

```
***************
*.**.**..*.**..
*******..*.****
*.**.**..*.**..
*.*************
```

input Copy

```
10
ECADBECADB
```

output Copy

```
******************************
*..*..*.**.**.**..*..*.**.**.*
****..****.********..****.****
*..*..*.**.**.**..*..*.**.**.*
*******.**************.*******
```

J

Drazil Likes Heap

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

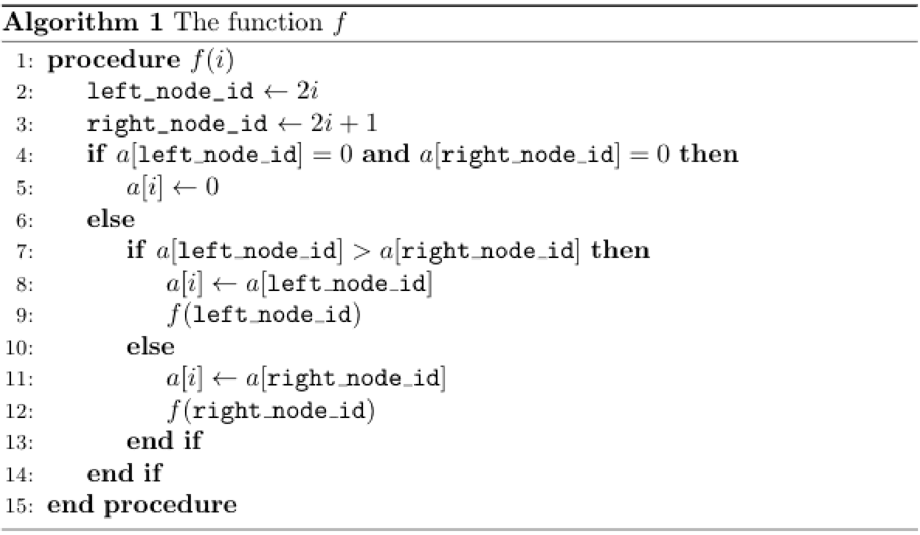
Drazil likes heap very much. So he created a problem with heap:

There is a max heap with a height h implemented on the array. The details of this heap are the following:

This heap contains exactly $2^h - 1$ **distinct** positive non-zero integers. All integers are distinct. These numbers are stored in the array a indexed from 1 to $2^h - 1$. For any $1 < i < 2^h$, $a[i] < a[\left\lfloor \frac{i}{2} \right\rfloor]$.

Now we want to reduce the height of this heap such that the height becomes g with exactly $2^g - 1$ numbers in heap. To reduce the height, we should perform the following action $2^h - 2^g$ times:

Choose an index i, which contains an element and call the following function f in index i:



Algorithm 1 The function f

```
procedure $f(i)$
    left_node_id $\leftarrow 2i$
    right_node_id $\leftarrow 2i + 1$
    if $a[$left_node_id$] = 0$ and $a[$right_node_id$] = 0$ then
        $a[i] \leftarrow 0$
    else
        if $a[$left_node_id$] > a[$right_node_id$]$ then
            $a[i] \leftarrow a[$left_node_id$]$
            $f($left_node_id$)$
        else
            $a[i] \leftarrow a[$right_node_id$]$
            $f($right_node_id$)$
        end if
    end if
end procedure
```

Note that we suppose that if $a[i] = 0$, then index i don't contain an element.

After all operations, the remaining $2^g - 1$ element must be located in indices from 1 to $2^g - 1$. Now Drazil wonders what's the minimum possible sum of the remaining $2^g - 1$ elements. Please find this sum and find a sequence of the function calls to achieve this value.

Input

The first line of the input contains an integer t ($1 \leq t \leq 70\,000$): the number of test cases.

Each test case contain two lines. The first line contains two integers h and g ($1 \leq g < h \leq 20$). The second line contains $n = 2^h - 1$ **distinct** positive integers $a[1], a[2], \ldots, a[n]$ ($1 \leq a[i] < 2^{20}$). For all i from 2 to $2^h - 1$, $a[i] < a[\left\lfloor \frac{i}{2} \right\rfloor]$.

The total sum of n is less than 2^{20}.

Output

For each test case, print two lines.

The first line should contain one integer denoting the minimum sum after reducing the height of heap to g. The second line should contain $2^h - 2^g$ integers $v_1, v_2, \ldots, v_{2^h-2^g}$. In i-th operation $f(v_i)$ should be called.

Example

input Copy

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

output Copy

```
10
3 2 3 1
8
2 1 3 1
```

K

BareLee

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

Lee is used to finish his stories in a stylish way, this time he barely failed it, but Ice Bear came and helped him. Lee is so grateful for it, so he decided to show Ice Bear his new game called "Critic"...

The game is a one versus one game. It has t rounds, each round has two integers s_i and e_i (which are determined and are known before the game begins, s_i and e_i may differ from round to round). The integer s_i is written on the board at the beginning of the corresponding round.

The players will take turns. Each player will erase the number on the board (let's say it was a) and will choose to write either $2 \cdot a$ or $a + 1$ instead. Whoever writes a number strictly greater than e_i loses that round and the other one wins that round.

Now Lee wants to play "Critic" against Ice Bear, for each round he has chosen the round's s_i and e_i in advance. Lee will start the first round, the loser of each round will start the next round.

The winner of the last round is the winner of the game, and the loser of the last round is the loser of the game.

Determine if Lee can be the winner independent of Ice Bear's moves or not. Also, determine if Lee can be the loser independent of Ice Bear's moves or not.

Input

The first line contains the integer t ($1 \le t \le 10^5$) — the number of rounds the game has.

Then t lines follow, each contains two integers s_i and e_i ($1 \le s_i \le e_i \le 10^{18}$) — the i-th round's information.

The rounds are played in the same order as given in input, s_i and e_i for all rounds are known to everyone before the game starts.

Output

Print two integers.

The first one should be `1` if Lee can be the winner independent of Ice Bear's moves, and `0` otherwise.

The second one should be `1` if Lee can be the loser independent of Ice Bear's moves, and `0` otherwise.

Examples

input Copy

```
3
5 8
1 4
3 10
```

output Copy

```
1 1
```

input Copy

```
4
1 2
```

```
2 3
3 4
4 5
```

output Copy

```
0 0
```

input Copy

```
1
1 1
```

output Copy

```
0 1
```

input Copy

```
2
1 9
4 5
```

output Copy

```
0 0
```

input Copy

```
2
1 2
2 8
```

output Copy

```
1 0
```

input Copy

```
6
216986951114298167 235031205335543871
148302405431848579 455670351549314242
506251128322958430 575521452907339082
1 768614336404564650
189336074809158272 622104412002885672
588320087414024192 662540324268197150
```

output Copy

```
1 0
```

Note

Remember, whoever writes an integer greater than e_i loses.

L

Vera and Dogs

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Vera owns N doghouses numbered from 1 to N and $M = X \cdot N$ dogs numbered from 1 to M. Each doghouse should be the *primary home* of X dogs $P_{i,1}, \ldots, P_{i,X}$ and the *secondary home* of X dogs $S_{i,1}, \ldots, S_{i,X}$. Each dog should have one primary home and one secondary home different from its primary home. Every night, at most one doghouse might be unavailable due to cleaning. Each dog will sleep in its primary home if it is available, otherwise it will sleep in its secondary home. Each doghouse should contain at most $X + 1$ sleeping dogs on any night.

Help Vera find a valid assignment of doghouses to dogs, or determine that it is impossible.

Input

Line 1 contains integers N and X ($1 \le N, X \le 2017$, $X \cdot N \le 50000$).

Output

If it is impossible to find a valid assignment, print one line with -1.

Otherwise print N lines. The i-th line should contain $2X$ integers $P_{i,1}, \ldots, P_{i,X}, S_{i,1}, \ldots, S_{i,X}$. If there are multiple possible assignments, print any of them.

Examples

input Copy

```
3 2
```

output Copy

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

input Copy

```
2 2
```

output Copy

```
-1
```

Note

For the first example:

Doghouse 1 is the primary home of dogs 5 and 1 and secondary home of dogs 6 and 4. If doghouse 1 is unavailable, then dog 1 will sleep in doghouse 3 and dog 5 will sleep in doghouse 2.

Doghouse 2 is the primary home of dogs 3 and 6 and secondary home of dogs 5 and 2. If doghouse 2 is unavailable, then dog 3 will sleep in doghouse 3 and dog 6 will sleep in doghouse 1.

Doghouse 3 is the primary home of dogs 4 and 2 and secondary home of dogs 1 and 3. If doghouse 3 is unavailable, then dog 4 will sleep in doghouse 1 and dog 2 will sleep in doghouse 2.

So it can be seen that no doghouse will ever contain more than 3 dogs.