Gracias Sponsors!

Organizador

Diamond

Gold

Contest 7 - Inicial

A

Division by Two and Permutation

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given an array a consisting of n positive integers. You can perform operations on it.

In one operation you can replace any element of the array a_i with $\lfloor \frac{a_i}{2} \rfloor$, that is, by an integer part of dividing a_i by 2 (rounding down).

See if you can apply the operation some number of times (possible 0) to make the array a become a permutation of numbers from 1 to n —that is, so that it contains all numbers from 1 to n, each exactly once.

For example, if $a = [1, 8, 25, 2]$, $n = 4$, then the answer is yes. You could do the following:

1. Replace 8 with $\lfloor \frac{8}{2} \rfloor = 4$, then $a = [1, 4, 25, 2]$.
2. Replace 25 with $\lfloor \frac{25}{2} \rfloor = 12$, then $a = [1, 4, 12, 2]$.
3. Replace 12 with $\lfloor \frac{12}{2} \rfloor = 6$, then $a = [1, 4, 6, 2]$.
4. Replace 6 with $\lfloor \frac{6}{2} \rfloor = 3$, then $a = [1, 4, 3, 2]$.

Input

The first line of input data contains an integer t ($1 \le t \le 10^4$) —the number of test cases.

Each test case contains exactly two lines. The first one contains an integer n ($1 \le n \le 50$), the second one contains integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 10^9$).

Output

For each test case, output on a separate line:

- YES if you can make the array a become a permutation of numbers from 1 to n,
- NO otherwise.

You can output YES and NO in any case (for example, strings yEs, yes, Yes and YES will be recognized as a positive response).

Example

input

```
6
4
1 8 25 2
2
1 1
9
9 8 3 4 2 7 1 5 6
3
8 2 1
4
24 7 16 7
5
22 6 22 4 22
```

output

```
YES
NO
YES
```

```
NO
NO
YES
```

Note

The first test case is explained in the text of the problem statement.

In the second test case, it is not possible to get a permutation.

B Sanatorium

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

Vasiliy spent his vacation in a sanatorium, came back and found that he completely forgot details of his vacation!

Every day there was a breakfast, a dinner and a supper in a dining room of the sanatorium (of course, in this order). The only thing that Vasiliy has now is a card from the dining room contaning notes how many times he had a breakfast, a dinner and a supper (thus, the card contains three integers). Vasiliy could sometimes have missed some meal, for example, he could have had a breakfast and a supper, but a dinner, or, probably, at some days he haven't been at the dining room at all.

Vasiliy doesn't remember what was the time of the day when he arrived to sanatorium (before breakfast, before dinner, before supper or after supper), and the time when he left it (before breakfast, before dinner, before supper or after supper). So he considers any of these options. After Vasiliy arrived to the sanatorium, he was there all the time until he left. Please note, that it's possible that Vasiliy left the sanatorium on the same day he arrived.

According to the notes in the card, help Vasiliy determine the minimum number of meals in the dining room that he could have missed. We shouldn't count as missed meals on the arrival day before Vasiliy's arrival and meals on the departure day after he left.

Input

The only line contains three integers b, d and s ($0 \le b, d, s \le 10^{18}$, $b + d + s \ge 1$) — the number of breakfasts, dinners and suppers which Vasiliy had during his vacation in the sanatorium.

Output

Print single integer — the minimum possible number of meals which Vasiliy could have missed during his vacation.

Examples

input

```
3 2 1
```

output

```
1
```

input

```
1 0 0
```

output

```
0
```

input

```
1 1 1
```

output

```
0
```

input

```
1000000000000000000 0 1000000000000000000
```

output

```
999999999999999999
```

Note

In the first sample, Vasiliy could have missed one supper, for example, in case he have arrived before breakfast, have been in the sanatorium for two days (including the day of arrival) and then have left after breakfast on the third day.

In the second sample, Vasiliy could have arrived before breakfast, have had it, and immediately have left the sanatorium, not missing any meal.

In the third sample, Vasiliy could have been in the sanatorium for one day, not missing any meal.

C And It's Non-Zero

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

You are given an array consisting of all integers from $[l, r]$ inclusive. For example, if $l = 2$ and $r = 5$, the array would be $[2, 3, 4, 5]$. What's the minimum number of elements you can delete to make the bitwise AND of the array non-zero?

A *bitwise AND* is a binary operation that takes two equal-length binary representations and performs the *AND* operation on each pair of the corresponding bits.

Input

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

The first line of each test case contains two integers l and r ($1 \le l \le r \le 2 \cdot 10^5$) — the description of the array.

Output

For each test case, output a single integer — the answer to the problem.

Example

input

```
5
1 2
2 8
4 5
1 5
100000 200000
```

output

```
1
3
0
2
31072
```

Note

In the first test case, the array is $[1, 2]$. Currently, the *bitwise AND* is 0, as $1\ \&\ 2 = 0$. However, after deleting 1 (or 2), the array becomes $[2]$ (or $[1]$), and the *bitwise AND* becomes 2 (or 1). This can be proven to be the optimal, so the answer is 1.

In the second test case, the array is $[2, 3, 4, 5, 6, 7, 8]$. Currently, the *bitwise AND* is 0. However, after deleting 4, 5, and 8, the array becomes $[2, 3, 6, 7]$, and the *bitwise AND* becomes 2. This can be proven to be the optimal, so the answer is 3. Note that there may be other ways to delete 3 elements.

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

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

output

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

```
3
4
```

input

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

output

```
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

C+=

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

Leo has developed a new programming language C+=. In C+=, integer variables can only be changed with a "+=" operation that adds the right-hand side value to the left-hand side variable. For example, performing "a += b" when a $= 2$, b $= 3$ changes the value of a to 5 (the value of b does not change).

In a prototype program Leo has two integer variables a and b, initialized with some positive values. He can perform any number of operations "a += b" or "b += a". Leo wants to test handling large integers, so he wants to make the value of either a or b **strictly greater** than a given value n. What is the smallest number of operations he has to perform?

Input

The first line contains a single integer T ($1 \leq T \leq 100$) — the number of test cases.

Each of the following T lines describes a single test case, and contains three integers a, b, n ($1 \leq a, b \leq n \leq 10^9$) — initial values of a and b, and the value one of the variables has to exceed, respectively.

Output

For each test case print a single integer — the smallest number of operations needed. Separate answers with line breaks.

Example

input

```
2
1 2 3
5 4 100
```

output

```
2
7
```

Note

In the first case we cannot make a variable exceed 3 in one operation. One way of achieving this in two operations is to perform "b += a" twice.

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

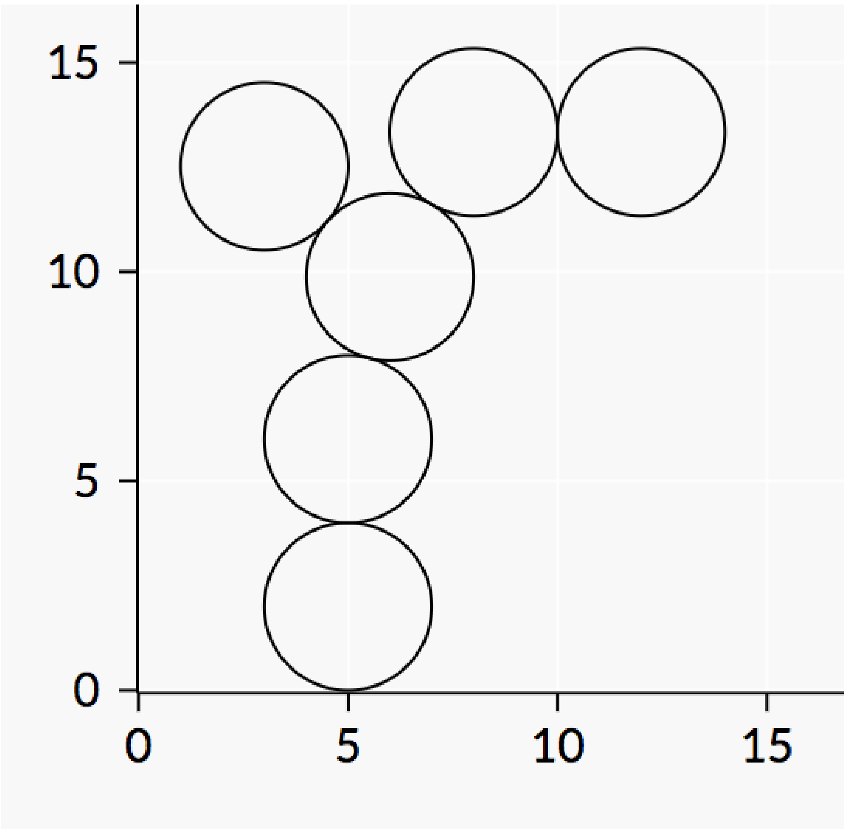
```
6 2
5 5 6 8 3 12
```

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

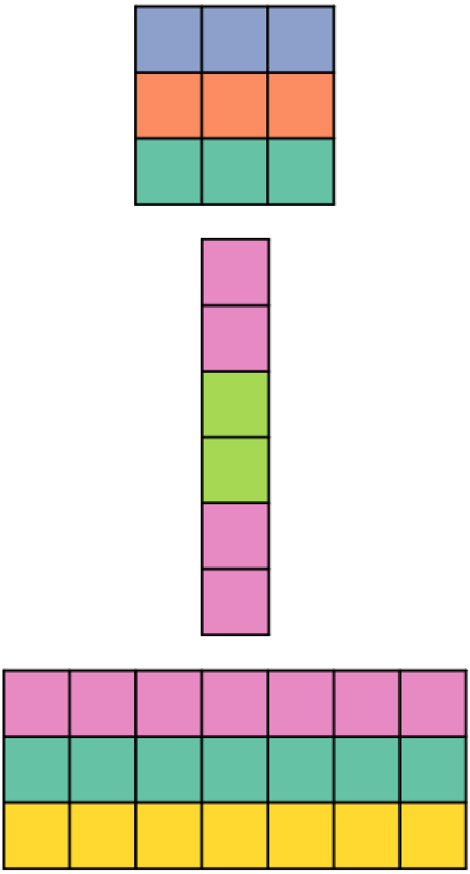
Flag

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

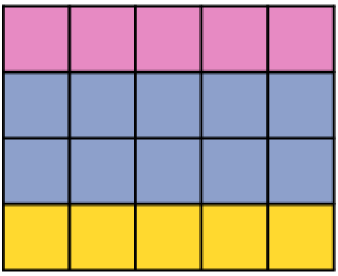
Innokenty works at a flea market and sells some ~~random stuff~~ rare items. Recently he found an old rectangular blanket. It turned out that the blanket is split in $n \cdot m$ colored pieces that form a rectangle with n rows and m columns.

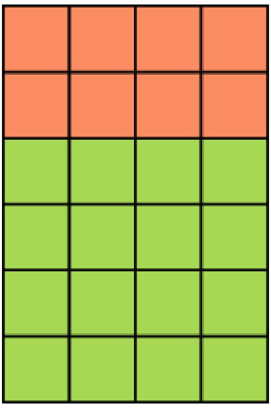
The colored pieces attracted Innokenty's attention so he immediately came up with the following business plan. If he cuts out a subrectangle consisting of three colored stripes, he can sell it as a flag of some country. Innokenty decided that a subrectangle is similar enough to a flag of some country 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.

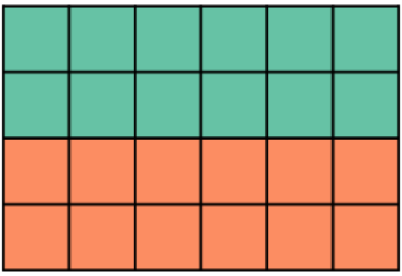
Innokenty has not yet decided what part he will cut out, but he is sure that the flag's boundaries should go along grid lines. Also, Innokenty won't rotate the blanket. Please help Innokenty and count the number of different subrectangles Innokenty can cut out and sell as a flag. Two subrectangles located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

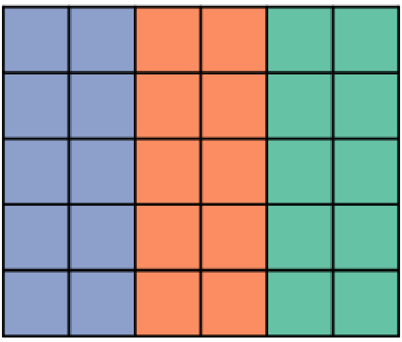


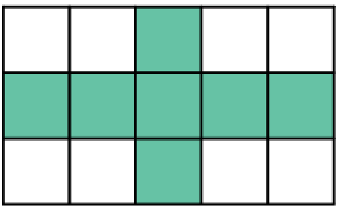
These subrectangles are flags.

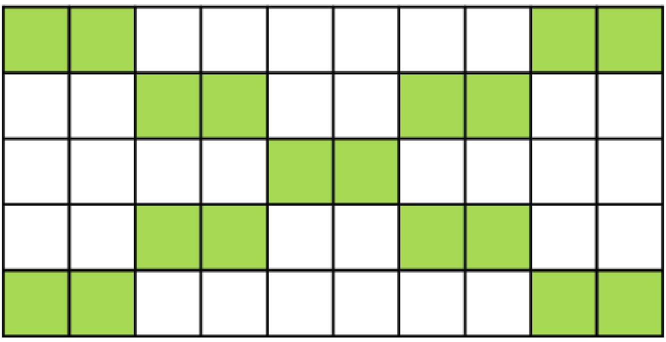












These subrectangles are not flags.

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
```

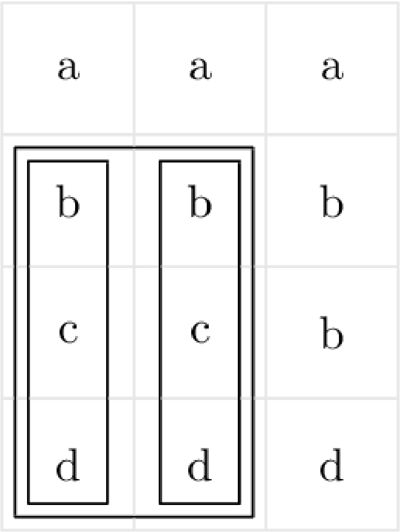
input Copy

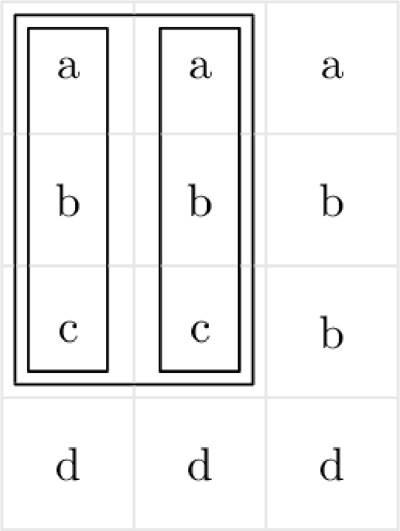
```
6 1
a
a
b
b
c
c
```

output Copy

```
1
```

Note





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

```
5
ABCDE
```

output

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

input

```
10
ECADBECADB
```

output

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

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 ← 2i
    right_node_id ← 2i + 1
    if a[left_node_id] = 0 and a[right_node_id] = 0 then
        a[i] ← 0
    else
        if a[left_node_id] > a[right_node_id] then
            a[i] ← a[left_node_id]
            f(left_node_id)
        else
            a[i] ← 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 ($1 \le t \le 70\,000$): the number of test cases.

Each test case contain two lines. The first line contains two integers h and g ($1 \le g < h \le 20$). The second line contains $n = 2^h - 1$ **distinct** positive integers $a[1], a[2], \ldots, a[n]$ ($1 \le a[i] < 2^{20}$). 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

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

output

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

K Equality

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

You are given a string s of length n, which consists only of the first k letters of the Latin alphabet. All letters in string s are uppercase.

A *subsequence* of string s is a string that can be derived from s by deleting some of its symbols without changing the order of the remaining symbols. For example, "ADE" and "BD" are subsequences of "ABCDE", but "DEA" is not.

A subsequence of s called *good* if the number of occurences of each of the first k letters of the alphabet is the same.

Find the length of the longest good subsequence of s.

Input

The first line of the input contains integers n ($1 \le n \le 10^5$) and k ($1 \le k \le 26$).

The second line of the input contains the string s of length n. String s only contains uppercase letters from 'A' to the k-th letter of Latin alphabet.

Output

Print the only integer — the length of the longest good subsequence of string s.

Examples

input

```
9 3
ACAABCCAB
```

output

```
6
```

input

```
9 4
ABCABCABC
```

output

```
0
```

Note

In the first example, "ACBCAB" ("**AC**AA**BC**C**AB**") is one of the subsequences that has the same frequency of 'A', 'B' and 'C'. Subsequence "CAB" also has the same frequency of these letters, but doesn't have the maximum possible length.

In the second example, none of the subsequences can have 'D', hence the answer is 0.

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

```
3 2
```

output

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

input

```
2 2
```

output

```
-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.