Problem A
ABC String
Time Limit: 1 second

You're given a string consisting of the characters A, B, and C. The string contains the same count of A, B, and C characters.

A string is *beautiful* if

- Its length is divisible by 3.
- The string can be split evenly into contiguous substrings of size 3, where each substring has one A, one B, and one C, in any order.

For example: ABCCBA is a beautiful string, but ABCAB and CCBAAB are not beautiful.

Given a string, you want to partition it into subsequences (not necessarily contiguous) such that each subsequence is a beautiful string.

For example, for the string ABACBCAACCBB, we can do the following:

```
AB    CA  C  B
  ACB   A  C  B
```

This partitions the string into two subsequences ABCACB and ACBACB, both of which are beautiful strings.

For the given string, find the minimum number of subsequences you can partition it into such that each subsequence is beautiful. It can be proven that there is always at least one such partition for all possible inputs that satisfy the input constraints.

Input

The first line of input contains a string s ($3 \leq |s| \leq 3 \cdot 10^5$). $|s|$ is divisible by 3. s contains an equal number of characters A, B, and C.

Output

Output a single integer, which is the minimum subsequences that s can be partitioned into so each subsequence is a beautiful string.

Sample Input 1	Sample Output 1
ABACBCAACCBB	2

Problem B
Balanced Tree Path
Time Limit: 2 seconds

You are given a tree where each node is annotated with a character from `()[]{}`. A path is a sequence of one or more nodes where no node is repeated and every pair of adjacent nodes is connected with an edge. A path is balanced if the characters at each node, when concatenated, form a balanced string. A string is balanced if it satisfies the following definition:

- An empty string is balanced.
- If s is a balanced string, then `(`s`)`, `[`s`]`, and `{`s`}` are balanced strings.
- if a and b are balanced strings, then ab (a concatenated with b) is a balanced string.

Compute the number of balanced paths over the entire tree.

Input

The first line of input contains a single integer n ($2 \le n \le 5 \cdot 10^3$).

The next line contains a string of n characters, where each character is one of `()[]{}`.

Each of the next $n-1$ lines contains two integers, u and v ($1 \le u < v \le n$), indicating that nodes u and v are connected with an edge. It is guaranteed the graph is a tree.

Output

Output a single integer, which is the number of balanced paths over the entire tree.

Sample Input 1	Sample Output 1
4 ()() 1 2 2 3 3 4	4

Sample Input 2	Sample Output 2
4 [[]] 1 2 2 3 3 4	2

Sample Input 3	Sample Output 3
6 ([]{}) 1 2 2 3 3 4 4 5 5 6	4

Problem C
Cramming for Finals
Time Limit: 4 seconds

It's final exam season and Ashley is heading to her favorite library to cram for finals.

The library has a dedicated floor for studying where there are r rows of c tables evenly spaced. Each table only has room for one student, and some students have already arrived and claimed their favorite tables.

Because the floor is usually very quiet, it is possible to hear sounds from other students who are nearby – for example, frustrated typing on a laptop keyboard or nervous leg shaking. Specifically, if one student is studying at the table in row i_1 and column j_1, and another student is studying at the table in row i_2 and column j_2, it is possible for the two students to hear sounds from each other if and only if $\sqrt{(i_1 - i_2)^2 + (j_1 - j_2)^2} \leq d$.

With this, Ashley wants to find an empty table where she can hear as few other students as possible. Compute the minimum number of students that Ashley can hear if she selects her table optimally.

Input

The first line of input has four integers r, c $(2 \leq r, c \leq 10^9)$, d $(1 \leq d \leq 2\,500)$, and n $(1 \leq n \leq 10^3$ and $n \leq r \cdot c - 1)$.

Each of the next n lines contains two integers i $(1 \leq i \leq r)$ and j $(1 \leq j \leq c)$, indicating that a student is studying at the table at row i and column j. It is guaranteed that no two students are sitting at the same table.

Output

Output a single integer, which is the minimum number of students that Ashley can hear if she selects her table optimally.

Sample Input 1	Sample Output 1
3 2 1 3 1 1 2 2 3 1	2

This page is intentionally left blank.

Problem D
Ordered Problem Set
Time Limit: 1 second

You are running a programming contest that features n problems of distinct difficulties. You wish to announce ahead of time that the problems are ordered in such a way that, if the problems are divided into k sections numbered 1 through k, each with exactly $\frac{n}{k}$ problems, and problem p is assigned to section $\lceil \frac{kp}{n} \rceil$, then for every pair of sections i and j with $i < j$, every problem in section i is easier than every problem in section j. Note that k must be greater than 1 and be a factor of n.

However, you have just sent your problems to the printer so the order cannot be changed. For what values of k would this claim be true?

Input

The first line of input contains a single integer n ($2 \le n \le 50$), which is the number of problems.

Each of the next n lines contains a single integer d ($1 \le d \le n$). These are the difficulties for the problems in the order that they appear in the problem set. The difficulties are distinct. The problem with difficulty 1 is the easiest problem and the problem with difficulty n is the hardest problem.

Output

Output a list of integers, one per line. The integers are all valid values of k in increasing order. If no such values exist, output -1.

Sample Input 1	Sample Output 1
6 1 3 2 4 5 6	2

Sample Input 2	Sample Output 2
6 1 2 3 4 5 6	2 3 6

Sample Input 3	Sample Output 3
6 6 5 4 3 2 1	-1

Problem E
Eccentric Excursion
Time Limit: 6 seconds

Eddy is planning a cross-country trip across n different cities. There are $n - 1$ roads connecting the cities. Each road connects two cities and is bidirectional. The roads are laid out such that it is possible to travel between any two cities using only roads.

Eddy wants to plan a trip so that he visits each city exactly once. He may start or end at any city. It might not be possible to visit each city exactly once using only roads. Luckily, Eddy can take a flight between any two cities that aren't directly connected by a road. Eddy would like to take exactly k flights during his trip.

Help Eddy plan his trip.

Input

The first line contains two integers n, k ($0 \leq k < n \leq 500$) where n is the number of cities Eddy is visiting and k is the number of flights Eddy would like to take.

The next $n - 1$ lines each contain two integers a, b ($1 \leq a < b \leq n$) indicating that there is a road between cities a and b. It is guaranteed it is possible to travel from any city to any other city only using roads.

Output

Output n integers that specify the sequence of cities that Eddy shall visit in order. The sequence must visit each city exactly once and use exactly k flights. If there are multiple possible itineraries, output the lexicographically smallest sequence. If there is no possible itinerary, output -1.

Sample Input 1	Sample Output 1
4 1 1 2 1 3 1 4	2 1 3 4

Sample Input 2	Sample Output 2
4 0 1 2 1 3 1 4	-1

Problem F
Matrix Fraud
Time Limit: 1 second

For the purposes of this problem: A matrix is a *binary matrix* if all its entries are 0 or 1. A matrix is a *banded binary matrix* if its rows and columns satisfy the following properties:

1. Every row has at least one 1.
2. Every column has at least one 1.
3. All 1s in each row are contiguous.
4. For row i, if s_i is the leftmost column that has a 1 entry and t_i is the rightmost column that has a 1 entry, then it must satisfy $s_i \geq s_{i-1}$ and $t_i \geq t_{i-1}$ for $i > 1$.

Detecting banded binary matrices is an important method used in various fields like biology, paleontology, and linguistics to unearth clusters in data sets. Unfortunately, a group called the Immoral Cartel of Pure Cozeners (ICPC) has decided to do the unthinkable: manipulate data! The ICPC wishes to present their groundbreaking scientific results, but the scientific community will not take their results seriously because their matrices may not be banded. To have publishable results, they want to toggle some cells such that their data is a banded binary matrix.

The ICPC gives you its raw data, represented as a binary matrix. They want to toggle some cells (meaning, change a 0 to a 1 or a 1 to a 0) so that the resulting matrix is a banded binary matrix, as defined above. What is the fewest number of toggles needed to turn the given matrix into a banded binary matrix?

Input

The first line of input contains two integers r and c ($1 \leq r \times c \leq 2 \cdot 10^5$), which are the dimensions of the matrix. The matrix has r rows and c columns.

Each of the next r lines contains a string of binary digits of length c. This is the matrix.

Output

Output a single integer, which is the minimum number of entries in the matrix to toggle to make the input matrix a binary banded matrix.

Sample Input 1	Sample Output 1
3 4 1100 0101 0011	1

Problem G
On-Call Team
Time Limit: 1 second

An IT company has formed an on-call team of software engineers who will manage their backend services and make sure that these services run without interruption. When services go down, for each service that is down the on-call team must dispatch one member who is familiar with that service to take care of its issue. One team member can handle at most one service at a time. The company wants to evaluate the *robustness level* of the on-call team, which is defined as the maximum value k such that any k services that go down simultaneously can be handled by the on-call team.

Input

The first line of input contains two integers n ($1 \le n \le 3 \cdot 10^4$) and m ($1 \le m \le 20$), where n is the number of engineers and m is the number of backend services.

Each of the next n lines contains a string of binary digits of length m, describing the n software engineers' familiarity with the m services. The j^{th} digit on the i^{th} line is 1 if software engineer i is familiar with service j, and 0 otherwise.

It is guaranteed that for each of the m services there exists at least one software engineer who is familiar with it.

Output

Output a single integer, which is the robustness level of the on-call team.

Sample Input 1	Sample Output 1
4 6 001101 111001 001110 100100	3

Sample Input 2	Sample Output 2
3 3 001 001 110	1

Problem H
Range Editing
Time Limit: 4 seconds

You are editing a list of spreadsheet cells. Initially all cells are empty. You can perform two types of operations:

1. Select a consecutive range of cells and change their values to a positive integer of your choice. All these cells get the same value after this operation.
2. Select a consecutive range of cells and delete their values. All these cells become empty after this operation.

Given the final cell values that you would like to have in the spreadsheet, calculate the minimum number of editing operations required to obtain those values.

Input

The first line of input contains a single integer n ($1 \le n \le 800$), which is the number of cells you are editing. The cells are number from 1 to n.

Each of the next n lines contains a single integer between 0 and 10^9 inclusive. The integer on the i^{th} line is 0 if cell i should be empty after all operations. Otherwise, it is a positive integer that is the final value of cell i.

Output

Output a single integer, which the minimum number of editing operations required.

Sample Input 1	Sample Output 1
10 0 1 2 3 4 2 1 0 0 1	5

Problem I
Item Selection
Time Limit: 1 second

You are browsing a website that lists items for sale. The website has a paging UI that displays a fixed number of items per page, one page at a time.

For example, if there are 55 items and the page displays exactly 20 at a time, then there are 3 pages in total. Items 1 through 20 are on page 1, items 21 through 40 are on page 2, and items 41 through 55 are on page 3.

You may navigate and select items using these UI elements:

- A checkbox for every item on the current page. After you click a checkbox, a selected item becomes unselected, and an unselected item becomes selected. You cannot click a checkbox for an item that is not on the current page.
- A "Select All" button. All unselected items on the current page become selected after you click this button.
- A "Deselect All" button. All selected items on the current page become unselected after you click this button.
- A "Next Page" button. Clicking it navigates to the next page and increments the current page number by one. This button is disabled on the last page.
- A "Previous Page" button. Clicking it navigates to the previous page and decrements the current page number by one. This button is disabled on the first page.

The website has pre-selected some items for you based on its machine learning recommendation algorithm. The recommendation may or may not work for you. You know exactly the items that you want to purchase, which may differ from the pre-selected items. What is the minimum number of checkbox and button clicks required to select exactly the items you actually want?

Input

The first line of input has five integers n, m ($1 \le m \le n \le 10^3$), s ($1 \le s \le \lceil \frac{n}{m} \rceil$), p, q ($0 \le p, q \le n$), where:

- n is the number of items. The items have item numbers from 1 to n.
- m is the fixed number of items displayed per page.
- s is the number of the page currently displayed.
- p is the number of preselected items.
- q is the number of items you want.

Each of the next p lines contains an integer i ($1 \le i \le n$). These are the item numbers of the preselected items. These p items are distinct and are listed in increasing order. It is possible that the website has pre-selected none of the items ($p = 0$), in which case the input has no lines for pre-selected items.

Each of the next q lines contains an integer j ($1 \le j \le n$). These are the item numbers of the items you want to buy. These q items are distinct and are listed in increasing order. It is possible that you want to buy none of the items ($q = 0$), in which case the input has no lines for items you want.

Output

Output a single integer, which is the minimum number of checkbox and button clicks required to select exactly the items you want.

Sample Input 1	Sample Output 1
11 4 1 5 5 1 4 9 10 11 1 3 6 7 8	7

Problem J
Sequence Guessing
Time Limit: 4 seconds

You are tasked with creating a secret sequence of integers that is difficult to guess.

The sequence is subject to the following constraints:

- The first number in the sequence must be 0.
- The last number in the sequence must be $100\,000$.
- Each number in the sequence must be either 1 or 2 greater than the one preceding it.

At first, all you need to reveal is the length of the sequence.

Then, an adversary will guess the numbers in the sequence one at a time.

- If the guessed number is in your sequence, you must reveal exactly where in the sequence it appears.
- If the guessed number is not in your sequence, you must simply reveal that it is not in the sequence. This is considered a "miss".

Note that because you are not forced to write down the sequence in advance, you can "cheat" by changing the sequence you have in mind, so long as it does not contradict the information you have revealed so far. It turns out that under these conditions, you can always force the adversary to get $33\,333$ misses before they can guess every number in your sequence. Your job is to write a program that does so.

Interaction

This is an interactive problem.

Your program should begin by printing k $(2 \leq k \leq 100\,001)$, the length of your sequence, as a single integer on a single line. After this, you will receive one integer x on a single line. This integer is guaranteed to be between -1 and $100\,000$ inclusive.

- If $x = -1$, the adversary has given up; your program should print all k integers in your sequence in order, one line per integer, and then exit. The adversary is guaranteed to give this input after it has gotten $33\,333$ misses, though it may do so earlier. After printing all

k integers, your program should exit. If you print a valid sequence consistent with your previous responses, your submission will be considered correct for this test case.

- If x is not in your sequence, print -1 on a single line.
- If x is in your sequence, print a single integer i on a single line, such that x is the ith (1-indexed) number in the sequence. If you have printed every integer from 1 to k, your program should now exit, and your submission will receive a Wrong Answer verdict.

Do not forget to flush the output after every integer you print.

After this, if your program has not yet exited, the process will repeat, with you receiving another single integer. The adversary is guaranteed to never repeat integers.

The adversary may employ different guessing strategies on different runs.

Read — **Sample Interaction 1** — **Write**

Read	Write
	50001
0	
	1
1	
	-1
-1	
	0 2 <omitted 49997 lines for brevity> 99998 100000

Problem K
Streets Behind
Time Limit: 1 second

Your running club has some serious runners and some casual runners. You schedule several training runs with a mixture of serious runners and casual runners. Serious runners run at a faster pace than casual runners, and will leave them behind.

You want all the runners to become serious runners, so when you schedule training runs, you carefully choose the number of serious and casual runners who will participate. You know that when there are x serious runners and y casual runners in a training run, if $\frac{x}{x+y}$ is greater than or equal to a threshold $\frac{a}{b}$, then after the run, all y casual runners, feeling the pressure to keep up with the serious runners become serious runners moving forward.

Compute the minimum number of training runs you need to convert all members of the club into serious runners.

Input

The first line of input contains a single integer t ($1 \le t \le 100$). This is the number of test cases.

Each of the next t lines contains a test case. Each test case consists of four integers n, k, a, b ($1 \le n, k, a, b \le 10^9$ and $a \le b$), where n is the number of serious runners, k is the number of casual runners, and $\frac{a}{b}$ is the threshold which converts casual runners to serious runners.

Output

Output t lines, each containing a single integer. For each test case, output the minimum number of training runs needed to convert all casual runners to serious runners. If it is impossible to convert all casual runners to serious runners, output -1 for that test case.

Sample Input 1	Sample Output 1
3 9 5 5 6 2 7 1 8 3 4 1 5	3 1 1

This page is intentionally left blank.

Problem L
Segment Drawing
Time Limit: 5 seconds

You're given some points, paired with an equal number of line segments. All points are strictly above the x-axis. Each line segment lies completely on the x-axis. None of the line segments share any common points.

You would like to draw some new line segments – one for each given point/line segment pair. The i^{th} drawn segment must connect point i to its corresponding line segment i. The correspondence between point i and line segment i is fixed, you cannot rearrange which point connects to which line segment.

No two drawn segments may strictly intersect, but it is allowed to have one drawn segment touching another drawn segment at an endpoint.

You would like to find the minimum total length of all drawn segments, or determine that it is impossible to draw such segments.

Input

The first line contains a single integer n ($1 \le n \le 10^5$), which is the number of given point/line segment pairs.

Each of the next n lines contains four integers x, y, l, r ($-10^6 \le x \le 10^6, 0 < y \le 10^6, -10^6 \le l \le r \le 10^6$). This denotes a point at (x, y) and its corresponding line segment with endpoints $(l, 0)$ and $(r, 0)$. The line segments will be given in order from left to right. No two given line segments will share a common point. It is possible for two points to be at the same location.

Output

Output a single real number, which is the minimum total length of all drawn segments. The answer will be accepted if it is within an absolute or relative error of at most 10^{-6}. If there is no way to draw the new segments without intersecting, output -1.

Sample Input 1	Sample Output 1
3 0 6 -4 -2 -2 1 -1 0 0 4 1 4	11.9995169566

Sample Input 2	Sample Output 2
3 0 6 -4 -2 0 4 -1 0 -2 1 1 4	-1

Sample Input 3	Sample Output 3
2 0 5 -1000000 -1 0 5 1 1000000	10.1980390272

Problem M
Champernowne Verification
Time Limit: 1 second

The k^{th} Champernowne word is obtained by writing down the first k positive integers and concatenating them together. For example, the 10^{th} Champernowne word is 12345678910.

Given a positive integer n, determine if it is a Champernowne word, and if so, which word.

Input

The first line contains a single integer, n ($1 \leq n \leq 10^9$). n will not have leading zeroes.

Output

If n is the k^{th} Champernowne word, output k. Otherwise, output -1.

Sample Input 1	Sample Output 1
`123456789`	`9`

Sample Input 2	Sample Output 2
`1000000000`	`-1`

Sample Input 3	Sample Output 3
`11`	`-1`

Sample Input 4	Sample Output 4
`1324`	`-1`

This page is intentionally left blank.