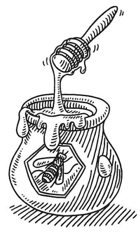
Croatian Open Competition in Informatics

Round 6, April 9th 2022

Tasks

Task	Time limit	Memory limit	Points
Med	1 second	512 MiB	50
Zemljište	2 seconds	512 MiB	70
Naboj	1 second	512 MiB	110
Palindromi	1 sekunda	512 MiB	110
Superpozicija	1 sekunda	512 MiB	110
Total			450

Task Med



The moment has finally arrived. Not only is it the last round of COCI this season, it is also the last round of COCB - the Croatian Open Competition in Beekeeping. Not many people know that the two competitions share the same scoring system. More precisely, both competitions consist of six rounds, the points on each round are between 0 and 500, and the scores of the individual rounds are summed up for the final tally.

After the six rounds, the competitors are ranked based on their total score over the six rounds. If two competitors have the same score, the one with the lexicographically smaller name appears sooner on the ranking. No two competitors have the same name.

The beekeepers are very impatient and they would like to know what their final ranking will be in advance. Each beekeeper wants to know their best and worst possible positions on the final ranking. Unlike skilled COCI programmers, the beekeepers don't know how to code. Therefore, they are asking you to determine the range of positions they could occupy after the sixth round.

Input

The first line contains a positive integer n ($1 \leq n \leq 500$), the number of beekeepers.

Each of the following n lines contains the name of a beekeeper s_i ($1 \leq |s_i| \leq 10$) and five numbers $b_{i1}, b_{i2}, b_{i3}, b_{i4}, b_{i5}$ from the range $[0, 500]$, the scores of the i-th beekeeper on the first five rounds of COCB. The names of the beekeepers are distinct and are made up of at most ten lowercase letters of the latin alphabet.

Output

Print n lines. In the i-th line print the best and worst possible position on the ranking for the i-th beekeeper.

Scoring

Subtask	Points	Constraints
1	10	$n = 2$
2	40	No additional constraints.

Examples

input

```
3
pavel 120 200 300 400 500
keko 150 400 300 200 100
bartol 470 120 90 93 189
```

output

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

input

```
2
ante 275 275 275 275 275
mate 25 100 175 250 325
```

output

```
1 1
2 2
```

Clarification of the second example:
So far, Ante has a sum of 1375, and Mate has 875. If Mate were to win 500 points on the last round,

and Ante 0, the result would be tied and they would both have 1375 points. However, since Ante is lexicographically smaller than Mate, Ante will still be ahead on the ranking.

Task Zemljište

Matej is, as is well known, the biggest Croatian innovator and businessman. His company is expanding, which is why he decided to buy a plot of land near Velika Gorica. The available land is a rectangle-shaped region consisting of $r \times s$ square unit cells. Each cell has its own price and it's not possible to purchase only a part of a cell. Matej is an experienced businessman and he knows that the key to success is not simply to buy the biggest plot of land or the least expensive one. Instead, he should buy a plot of land whose price is as close as possible to the magic numbers given to him by Milan the psychic.

At the beginning of his career, Milan revealed to Matej two magic numbers a and b crucial for commercial success. Therefore, Matej wishes to buy a (nonempty) rectangular plot of land so that the distance between its price and the magic numbers is as small as possible. The distance between the price and a single magic number is just the absolute value of their difference, and the distance between the price and both of the magic numbers is the sum of these absolute differences. Help Matej and determine the minimum possible distance between the price of the plot of land and the two magic numbers.

Input

The first line contains positive integers r, s, a and b ($1 \le r, s \le 500$, $1 \le a, b \le 10^9$) from the problem.

The i-th of the following r lines contains a sequence of s positive integers c_{ij} ($1 \le c_{ij} \le 10^9$), the prices of the individual cells, in order.

Output

In the only line print the minimum possible distance between the price of the plot of land and the two magic numbers.

Scoring

Subtask	Points	Constraints
1	10	$1 \le r, s \le 20$
2	20	$1 \le r, s \le 100$
3	40	No additional constraints.

Examples

input

```
2 2 10 10
1 3
4 1
```

output

```
2
```

input

```
3 2 3 4
1 9
1 1
8 1
```

output

```
3
```

input

```
3 4 5 3
1 1 1 1
9 6 7 6
8 1 9 7
```

output

```
2
```

Clarification of the second example:
Matej can buy a plot of land consisting of two adjacent cells of cost 1. The total price is then $1 + 1 = 2$, and the distance between it and the magic numbers is $|3 - 2| + |4 - 2| = 3$.

Task Naboj

Mr. Šikić, a chemistry teacher, is playing around with n metal balls and m copper wires. He joined together some pairs of balls with a wire so that all the balls are (directly or indirectly) linked to each other. He wants to teach his students about electric charge so he'll demonstrate it by charging the metal balls in a sequence.

Mr Šikić can either charge each of the balls positively or negatively. When a ball is charged negatively, the electrons in all the wires connected to the ball are repulsed to the other ball connected to that wire. Conversely, if a ball is positively charged, the electrons from all the wires connected to that ball are pulled towards it. Charging the balls has the same effect on the wires irrespective of the wire's previous state.

At the beginning of the class all the balls hold no charge and the electrons in all the wires are still. For every wire Mr. Šikić has a specific direction of the electron flow in mind. Help him find a sequence of ball chargings that results in the desired electron flows.

Input

The first line contains two integers n and m ($1 \le n \le 200\,000$, $1 \le m \le 500\,000$) from the task statement.

The following m lines contain integers a_i and b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$) denoting that the balls a_i and b_i are connected by a wire and the electrons in the wire should be closer to a_i, and not b_i. There is at most one wire between a pair of balls. All the balls are directly or indirectly connected by wires.

Output

If it is impossible to direct the flow of electrons according to Mr. Šikić's wishes print `-1`. Otherwise print k, the required number of ball chargings. k must be less than or equal 200 000.

In the following k lines print integers c_i and d_i ($1 \le c_i \le n$, $0 \le d_i \le 1$), the number of the ball Mr. Šikić should charge in i-th step and whether it should be charged positively (denoted by $d_i = 1$) or negatively ($d_i = 0$). If there are multiple solutions, print any one of them.

Scoring

Subtask	Points	Constraints
1	15	$1 \le n \le 10$
2	25	$m = n - 1$
3	70	No additional constraints.

Examples

input

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

output

```
3
2 1
3 0
1 1
```

input

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

output

```
4
2 1
4 0
3 1
1 1
```

input

```
5 10
2 4
3 4
1 4
4 5
3 2
2 1
5 2
1 3
5 3
1 5
```

output

```
-1
```

Clarification of the first example:
First, we give the ball 2 a positive charge. The electrons on wires between balls 1 and 2, and balls 2 and 3 are now closer to the ball 2. The wire connecting balls 1 and 3 remains neutral.

Now we give ball 3 a negative change. The arrangement of electrons between wires 2 and 3 remains unchanged, while the electrons on the wire between 1 and 3 are closer to the ball 1.

Finally we give ball 1 a positive charge. The wire between 1 and 3 remains unchanged, but on the wire between balls 1 i 2 electrons are now closer to the ball 1 and the desired arrangement is achieved.

Task Palindromi

You are given a sequence of n characters `0` or `1`, indexed by numbers $1, 2, \ldots, n$. Initially every character represents a string of length one. During a *concatenation* two words a and b are chosen, deleted, and replaced by the string ab such that the characters of b are written after the characters of a.

The n initial strings are concatenated to one final string using a sequence of $n-1$ concatenations. The i-th of those concatenation is described by a pair of indexes (a_i, b_i), which denotes that the string containing a_i-th character and the string containing b_i-th character are to be concatenated. It is guaranteed that characters with indexes a_i and b_i are not in the same string.

Palindromic value of some string w is defined as the total number of unique substrings of w which are palindromes. We define palindromes as strings that are the same when read left to right and right to left. A substring of a string is defined as a string obtained by erasing zero or more characters from the beginning and/or ending of the string.

For every concatenation print the palindromic value of the resulting string.

Input

The first line contains an integer n ($1 \leq n \leq 100\ 000$), number of characters.

In the second line there is a string of n characters `0` and `1` which represent the initial strings.

The i-th of following $n-1$ lines contains two integers a_i i b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$) representing the i-th concatenation.

Output

Print $n-1$ lines, the palindromic values of words obtained after each concatenation.

Scoring

Subtask	Points	Constraints
1	10	$1 \leq n \leq 100$.
2	20	$1 \leq n \leq 1000$.
3	30	$a_i = 1$, $b_i = i + 1$ for all $i = 1, 2, \ldots, n-1$.
4	50	No additional constraints.

Examples

input

```
3
010
1 2
2 3
```

output

```
2
3
```

input

```
5
00111
4 1
1 5
2 1
3 1
```

output

```
2
3
4
5
```

input

```
8
10010000
7 5
4 2
3 6
1 3
6 8
5 3
1 2
```

output

```
2
2
2
3
4
6
8
```

Clarification of the third example:
Newly created strings after every concatenation are: 00, 10, 00, 100, 1000, 001000 and 00100010. Their respective palindromic values are given in the example output. E. g. the palindromic value of 00100010 is 8 because the string contains 8 palindromic substring: 0, 00, 000, 10001, 0100010, 1, 010 i 00100.

Task Superpozicija

World-renowned physicist Juraj has recently discovered a new kind of elementary particle – a parenthesision. A parenthesision can have either an open '(' or ')' closed configuration. Using his homemade particle accelerator, Juraj has created t sequences of superpositions of n parenthesisions. In each of the t sequences there are n parenthesisions in a superposition between two different positions positions and (not necessarily different) configurations. If the sequence is observed, the wave function of parenthesisions collapses and each of them ends up in one of it's possible positions and configurations. Juraj wants to know if it is possible that the parenthesisions collapse into a valid sequence of parenthesis?

Juraj M. PhD knows that the quantum physics of these revolutionary and completely scientifically founded particles are way over the head of an average COCI contestant, so he provided a formal task statement:

You are given t sequences of $2n$ open and closed parenthesis. Each of the parenthesis is a member of exactly one pair of parenthesis. Two parenthesis within a pair can be different, both open, or both closed. Juraj wants to know if it is possible to chose a single parenthesis from each of the pairs such that the chosen parentheses form a valid sequence of parentheses. Furthermore, if this is possible he asks you to print which parentheses he should chose to get a valid sequence. A sequence of parentheses is valid if it is empty or it can be written as `(A)` or `AB` where `A` and `B` are arbitrary valid sequences of parentheses.

Input

The first line contains an integer t ($1 \leq t \leq 100\ 000$), number of parentheses sequences. t sequence descriptions follow.

The first line of sequence description contains an integer n ($1 \leq n \leq 100\ 000$), number of pairs of parentheses in the sequence.

The second line contains z, a string of length $2n$. z contains exclusively characters `(` and `)`.

The following n lines of sequence description contain two integers a_i and b_i ($1 \leq a_i < b_i \leq 2n$). Each of the numbers $1, 2, \ldots 2n$ appears exactly once.

Sum of all n will be less than or equal to 100 000.

Output

In the i-th of t lines print a sequence of zeros and ones representing a possible choice of parentheses. If parenthesis at index a_j of the j-th pair of i-th sequence is chosen, print `0`, otherwise if parenthesis at index b_j is chosen, print `1`. If there is no valid sequence of parentheses print `-1`.

Scoring

Subtask	Points	Constraints
1	10	$1 \leq n \leq 10$
2	10	$z[a_i] = z[b_i]$ for all $i = 1, 2, \ldots n$.
3	20	$b_i = a_i + 1$ for all $i = 1, 2, \ldots n$.
4	70	No additional constraints.

Examples

input

```
1
4
()))(((()
1 2
3 5
4 6
7 8
```

output

```
0 1 0 1
```

input

```
1
4
)()()()(
1 2
3 4
5 6
7 8
```

output

```
1 1 0 0
```

input

```
1
3
(()())
1 6
2 4
3 5
```

output

```
-1
```

Clarification of the first example:
From the original sequence `())((()`, only the bolded parentheses will remain `())((()`. That is `()()`, which is a valid sequence of parentheses.

Clarification of the second example:
From the original sequence `)()()()(`, only the bolded parentheses will remain `)()()()(`. That is `(())`, which is a valid sequence of parentheses.