The 2026 ICPC China Shaanxi National Invitational Programming Contest

Contest Session

May 2, 2026

西北工業大學

NORTHWESTERN POLYTECHNICAL UNIVERSITY

Problem List

A	North and South
B	Operating Robot
C	Palindromic and Balanced
D	Qenerals
E	Registration
F	Split Sticks
G	Transform
H	Unreachable Land
I	VIP Coupon
J	Would You Make a Convex?
K	XOR and LCA
L	Yesterday Once More (Easy Version)
M	Yesterday Once More (Hard Version)
N	Zebra Crossing

This problem set should contain 14 (fourteen) problems on 27 (twenty-seven) numbered pages.

Please inform a runner immediately if something is missing from your problem set.

Hosted By

西北工業大學

NORTHWESTERN POLYTECHNICAL UNIVERSITY

Problem Set Prepared By

Cfz 出题组

Cfz Problem Setting Group

Problem A. North and South

Input file:	`standard input`
Output file:	`standard output`

Yuki has a sequence a of length n.

Yuki defines an operation as follows:

- Choose an interval $[l, r]$ of **even length**. For every integer i such that $l \le i \le r$:
  - If $i - l$ is odd, the value of a_i decreases by 1, i.e., $a_i \leftarrow a_i - 1$.
  - If $i - l$ is even, the value of a_i increases by 1, i.e., $a_i \leftarrow a_i + 1$.

Now, Yuki wants to perform some number of operations such that all numbers in the sequence a are equal. You need to help Yuki find the minimum number of operations required to make all numbers in the sequence a equal, or report if it is impossible.

Input

This problem contains multiple test cases.

The first line contains a positive integer t ($1 \le t \le 10^5$), representing the number of test cases.

For each test case:

- The first line contains a positive integer n ($1 \le n \le 10^6$).
- The second line contains n integers $a_1, \ldots, a_n$ ($0 \le a_i \le 10^{12}$).

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

For each test case, output one line:

- If it is impossible, output -1.
- If it is possible, output an integer representing the minimum number of operations to make all numbers in the sequence a equal.

Example

standard input	standard output
3 2 1 3 4 1 5 1 5 5 1 3 1 3 1	1 2 -1

Note

For the first test case:

- Perform the operation on the interval $[1, 2]$. The sequence becomes $2, 2$, where all numbers are equal.

- It can be proven that no solution with fewer operations exists, so the answer is 1.

For the second test case:

- Perform the operation on the interval $[1, 4]$. The sequence becomes $2, 4, 2, 4$.
- Perform the operation on the interval $[1, 4]$. The sequence becomes $3, 3, 3, 3$, where all numbers are equal.
- It can be proven that no solution with fewer operations exists, so the answer is 2.

For the third test case:

- It is easy to prove that it is impossible to make all numbers equal regardless of the number of operations, so the answer is -1.

Problem B. Operating Robot

Input file: `standard input`
Output file: `standard output`

A robot is located on a 2D Cartesian coordinate system. Initially, the robot is at $(0, 0)$, and Yuki wants to guide the robot to (x, y) using a sequence of instructions.

Specifically, an instruction string consists only of `0` and `1`:

- `0` represents moving one step to the right, changing the robot's position from (a, b) to $(a + 1, b)$.
- `1` represents moving one step upward, changing the robot's position from (a, b) to $(a, b + 1)$.

Yuki has an instruction string $s = s_1 \dots s_n$ of length n containing only `0`, `1`, and `2`. Yuki must first replace all `2`s in the string with either `0` or `1`. Then, the robot operates according to the following rule:

- For every non-negative integer i, if the robot is not at (x, y) at time i, the robot executes the $((i \bmod n) + 1)$-th instruction of the string.

Yuki wants to find a replacement such that the robot reaches (x, y) and the resulting instruction string is lexicographically as small as possible. You need to help Yuki find the lexicographically smallest instruction string that satisfies the condition, or report that no such string exists.

Input

This problem contains multiple test cases.

The first line contains a positive integer t ($1 \le t \le 10^5$), representing the number of test cases.

For each test case:

- The first line contains three integers n, x, y ($1 \le n \le 10^6,\ 0 \le x, y \le 10^{18}$).
- The second line contains a string s of length n ($s_i \in \{\texttt{0}, \texttt{1}, \texttt{2}\}$).

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

For each test case, output one line:

- If no such instruction string exists, output -1.
- If such an instruction string exists, output a string of length n representing the lexicographically smallest valid instruction string.

Example

standard input	standard output
6 5 2 4 01111 5 3 3 02221 5 3 3 00022 6 1 0 011201 4 8 7 2020 5 0 0 22102	01111 00111 -1 011001 -1 00100

Note

For the first test case:

- Initially, the robot is at $(0, 0)$, and the instruction string is `01111`.
- Following the rules, the robot moves sequentially to $(1, 0), (1, 1), (1, 2), (1, 3), (1, 4), (2, 4)$.
- Since the robot reaches $(2, 4)$, `01111` is a valid instruction string. It can be proven that `01111` is the lexicographically smallest valid string, so the answer is `01111`.

For the second test case:

- Initially, the robot is at $(0, 0)$, and we replace the instruction string `02221` with `00111`.
- Following the rules, the robot moves sequentially to $(1, 0), (2, 0), (2, 1), (2, 2), (2, 3), (3, 3)$.
- Since the robot reaches $(3, 3)$, `00111` is a valid instruction string. It can be proven that `00111` is the lexicographically smallest valid string, so the answer is `00111`.

For the third test case:

- It can be proven that there is no way to replace the `2`s in the instruction string such that the robot reaches $(3, 3)$.

Problem C. Palindromic and Balanced

Input file: `standard input`
Output file: `standard output`

Yuki discovered that a palindromic bracket sequence cannot be a balanced bracket sequence, so she designed another way to define a palindromic balanced bracket sequence.

Yuki defines a **palindromic bracket sequence** according to the following rules:

- The empty string is a palindromic bracket sequence.
- `(` and `)` are palindromic bracket sequences.
- If a bracket sequence s is a palindromic bracket sequence, then `(`s`(` and `)`s`)` are palindromic bracket sequences.

Yuki defines a **balanced bracket sequence** according to the following rules:

- The empty string is a balanced bracket sequence.
- If a bracket sequence s is a balanced bracket sequence, then `(`s`)` is a balanced bracket sequence.
- If bracket sequences s and t are both balanced bracket sequences, then st (the concatenation of the two bracket sequences) is a balanced bracket sequence.

For a bracket sequence $s = s_1 \dots s_n$, Yuki defines s as a **palindromic balanced bracket sequence** if and only if:

- $s_2 \dots s_{n-1}$ is a palindromic bracket sequence.
- $s_1 \dots s_n$ is a balanced bracket sequence.

Specifically, the empty string and `()` are also palindromic balanced bracket sequences.

For example, `(())()` and `()()(()())` are palindromic balanced bracket sequences, while `((()))` and `()()(())` are not.

Now, Yuki has a bracket sequence s of length n, and she wants to find the longest subsequence* of s that is a palindromic balanced bracket sequence. However, Yuki does not know how to do this, so you need to help her find the length of the longest such subsequence.

*: A sequence a is a subsequence of sequence b if and only if a can be obtained by deleting zero or more elements from b; specifically, the empty sequence is a subsequence of any sequence.

Input

This problem contains multiple test cases.

The first line contains a positive integer t ($1 \le t \le 5000$), representing the number of test cases.

For each test case:

- The first line contains a positive integer n ($1 \le n \le 5000$).
- The second line contains a bracket sequence s of length n ($s_i \in \{(,)\}$).

It is guaranteed that the sum of n over all test cases does not exceed 10^4.

Output

For each test case, output a single line containing an integer representing the length of the longest subsequence that satisfies the condition.

Example

standard input	standard output
3 5 (()((7)))((((8 ()))(()()	2 0 6

Note

For the first test case:

- The longest subsequences satisfying the condition are $s_1s_3 =$ () and $s_2s_3 =$ (), so the answer is 2.

For the second test case:

- The longest subsequence satisfying the condition is the empty string, so the answer is 0.

For the third test case:

- The longest subsequence satisfying the condition is $s_1s_2s_4s_5s_6s_8 =$ ()(()), so the answer is 6.

Problem D. Qenerals

Input file:	`standard input`
Output file:	`standard output`

Yuki is playing a game called Qenerals.

At time $t = 0$, Yuki has $x = 0$ soldiers and occupies $y = 1$ fortress. There are n unoccupied fortresses on the map, where the i-th fortress has a parameter a_i.

The game lasts for m seconds. For each positive integer i such that $1 \le i \le m$:

- At the beginning of the i-th second, each fortress occupied by Yuki produces 1 soldier, i.e., $x \leftarrow x+y$.
- At the end of the i-th second, Yuki can perform any number of operations (including zero). In each operation, Yuki chooses an unoccupied fortress j such that $a_j \le x$, consumes a_j soldiers, and occupies fortress j, i.e., $x \leftarrow x - a_j$ and $y \leftarrow y + 1$.

You need to help Yuki determine the maximum number of soldiers she can have after the game ends.

Input

This problem contains multiple test cases.

The first line contains a positive integer t $(1 \le t \le 10^5)$, representing the number of test cases.

For each test case:

- The first line contains two positive integers n, m $(1 \le n \le 5 \cdot 10^5,\ 1 \le m \le 10^9)$.
- The second line contains n positive integers $a_1, \ldots, a_n$ $(1 \le a_i \le 10^9)$.

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

Output

For each test case, output a single line containing an integer representing the maximum number of soldiers Yuki can have after the game ends.

Example

standard input	standard output
3 2 3 2 1 3 6 1 1 3 3 5 1 1 1	4 13 12

Note

For the first test case:

- At the beginning of the 1-st second, the number of fortresses occupied by Yuki is $y = 1$, so the number of soldiers x increases from 0 to 1.
- At the end of the 1-st second, Yuki can choose to occupy the 2-nd fortress, so y increases from 1 to 2, and x decreases from 1 to 0.

- At the beginning of the 2-nd second, $y = 2$, so x increases from 0 to 2.
- At the end of the 2-nd second, Yuki can choose not to perform any operations.
- At the beginning of the 3-rd second, $y = 2$, so x increases from 2 to 4.
- At the end of the 3-rd second, Yuki can choose not to perform any operations.
- After the game ends, the number of soldiers x is 4. It can be proven that 4 is the maximum number of soldiers Yuki can have.

For the second test case:

- Yuki can occupy the 1-st, 2-nd, and 3-rd fortresses at the end of the 1-st, 2-nd, and 3-rd seconds respectively, allowing her to have 13 soldiers after the game ends.

For the third test case:

- Yuki can occupy the 1-st fortress at the end of the 1-st second, and the 2-nd and 3-rd fortresses at the end of the 2-nd second, allowing her to have 12 soldiers after the game ends.

Problem E. Registration

Input file:	`standard input`
Output file:	`standard output`

The ICPC 2026 Chang'an Invitational Tournament is about to begin. There are n teams planning to register for the competition, where the i-th team will visit the registration website at the v_i-th second.

For every positive integer s:

- If the number of teams visiting the registration website at the s-th second is no more than x, then all teams visiting at the s-th second register successfully.
- Otherwise, the registration website crashes at that second due to too many teams visiting, and all teams visiting at the s-th second fail to register.

As the chief judge of the ICPC 2026 Chang'an Invitational Tournament, Yuki wants to know the registration status of this competition. Therefore, you need to help her calculate the number of teams that successfully register.

Input

This problem contains multiple test cases.

The first line contains a positive integer t ($1 \le t \le 10^5$), representing the number of test cases.

For each test case:

- The first line contains two positive integers n, x ($1 \le n \le 2 \cdot 10^5,\ 1 \le x \le n$).
- The second line contains n positive integers $v_1, \ldots, v_n$ ($1 \le v_i \le 10^9$).

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 a single line containing an integer representing the number of teams that successfully register.

Example

standard input	standard output
3 5 2 1 2 3 1 1 5 2 3 1 4 1 5 6 1 2 3 1 3 1 2	2 5 0

Note

For the first test case:

- At the 1-st second, teams $1, 4, 5$ visited the registration website. Since $3 > 2$, the website crashed at this second, and these 3 teams failed to register.
- At the 2-nd second, team 2 visited the registration website. Since $1 \le 2$, the website did not crash, and this team registered successfully.

- At the 3-rd second, team 3 visited the registration website. Since $1 \leq 2$, the website did not crash, and this team also registered successfully.
- A total of 2 teams registered successfully, so the answer is 2.

For the second test case:

- The website did not crash at any second, all 5 teams registered successfully, so the answer is 5.

For the third test case:

- The website crashed in each of the first 3 seconds, all 6 teams failed to register, so the answer is 0.

Problem F. Split Sticks

Input file: `standard input`
Output file: `standard output`

Yuki has n sticks arranged in a row, where the length of the i-th stick is a_i.

Yuki defines an operation as follows:

- Choose a stick and cut it into two parts, both of which must have integer lengths (one part can have a length of 0).
- Merge the left part of the cut stick with the stick immediately to its left; if there is no stick to its left, the left part becomes a new, independent stick.
- Merge the right part of the cut stick with the stick immediately to its right; if there is no stick to its right, the right part becomes a new, independent stick.
- Remove all sticks with a length of 0.

Now, Yuki wants to perform a number of operations such that all sticks have the same length. You need to help Yuki find the minimum number of operations required to make all sticks have the same length.

It can be proven that there always exists at least one sequence of operations that makes all sticks have the same length.

Input

This problem contains multiple test cases.

The first line contains a positive integer t ($1 \le t \le 10^5$), representing the number of test cases.

For each test case:

- The first line contains a positive integer n ($1 \le n \le 10^6$).
- The second line contains n positive integers $a_1, \ldots, a_n$ ($1 \le a_i \le 10^6$).

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

For each test case, output a single line containing an integer representing the minimum number of operations required to make all sticks have the same length.

Example

standard input	standard output
3 3 1 5 4 4 1 4 2 5 5 3 3 3 3 3	1 2 0

Note

For the first test case:

- In the first operation, choose the second stick and cut it into two parts of lengths 4 and 1. The lengths of the sticks from left to right become $5, 5$, and all sticks have the same length.
- It can be proven that the minimum number of operations required is 1.

For the second test case:

- In the first operation, choose the first stick and cut it into two parts of lengths 0 and 1. The lengths of the sticks from left to right become $5, 2, 5$.
- In the second operation, choose the second stick and cut it into two parts of lengths 1 and 1. The lengths of the sticks from left to right become $6, 6$, and all sticks have the same length.
- It can be proven that the minimum number of operations required is 2.

For the third test case:

- Initially, all sticks have the same length, so the minimum number of operations required is 0.

Problem G. Transform

Input file: standard input
Output file: standard output

Yuki has a multiset $S = \{s_1, \ldots, s_n\}$ of size n and an integer k.

Yuki defines a transformation as follows:

- Choose a subset S' of S (where S' can be an empty set), remove S' from S, and add the mex* of S' to S.

Now, Yuki wants to perform several transformations such that S becomes $\{k\}$. You need to help Yuki find the minimum number of transformations required to make S equal to $\{k\}$. Since the answer can be very large, you only need to output the answer modulo 998244353.

It can be proven that there always exists at least one sequence of operations that can transform S into $\{k\}$.

*: The mex of a multiset is the smallest non-negative integer that does not appear in the multiset. For example, $\text{mex}\{0, 1, 2\} = 3$, $\text{mex}\{1, 0, 3, 1\} = 2$, and $\text{mex}\,\varnothing = 0$.

Input

This problem contains multiple test cases.

The first line contains a positive integer t ($1 \le t \le 10^5$), representing the number of test cases.

For each test case:

- The first line contains two integers n, k ($1 \le n \le 5 \cdot 10^5,\ 0 \le k \le 10^9$).
- The second line contains n integers $s_1, \ldots, s_n$ ($0 \le s_i \le 10^9$).

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

Output

For each test case, output a single line containing an integer representing the minimum number of transformations required to make S equal to $\{k\}$, modulo 998244353.

Example

standard input	standard output
6 1 2 1 1 4 4 3 3 0 2 2 4 2 1 0 3 2 4 3 2 1 0 2 3 52 20 2 6	2 0 3 2 1 262875292

Note

For the 1st test case:

- Yuki can choose $S' = \varnothing$ in the 1st transformation, making $S = \{0, 1\}$, and then choose $S' = \{0, 1\}$ in the 2nd transformation, making $S = \{2\}$.
- It can be proven that no sequence of operations with fewer transformations exists, so the answer is 2.

For the 2nd test case:

- Yuki does not need to perform any transformations to make $S = \{4\}$, so the answer is 0.

For the 3rd test case:

- Yuki can choose $S' = \varnothing$ in the 1st transformation, making $S = \{0, 0, 2, 2\}$, choose $S' = \{0, 2\}$ in the 2nd transformation, making $S = \{0, 1, 2\}$, and then choose $S' = \{0, 1, 2\}$ in the 3rd transformation, making $S = \{3\}$.
- It can be proven that no sequence of operations with fewer transformations exists, so the answer is 3.

For the 4th test case:

- Yuki can choose $S' = \{2, 3\}$ in the 1st transformation, making $S = \{0, 0, 1\}$, and then choose $S' = \{0, 0, 1\}$ in the 2nd transformation, making $S = \{2\}$.
- It can be proven that no sequence of operations with fewer transformations exists, so the answer is 2.

For the 5th test case:

- Yuki can directly choose $S' = \{0, 1, 2, 2\}$ in the 1st transformation, making $S = \{3\}$.
- It can be proven that no sequence of operations with fewer transformations exists, so the answer is 1.

Problem H. Unreachable Land

Input file: `standard input`
Output file: `standard output`

Yuki dreams of reaching the unreachable land. After years of hard work, only this problem remains before her.

Given three integers a, b, m. You need to perform m rounds of operations. In the i-th round, you can either set $a \leftarrow a \bmod (m - i + 1)$ or choose not to modify a. Find the number of ways to make $a = b$ after m rounds, modulo 998244353.

Two schemes are considered different if and only if there exists some $1 \le i \le m$ such that in one scheme you performed a modification in the i-th round, while in the other you did not. Note that choosing to perform $a \leftarrow a \bmod (m - i + 1)$ is considered a modification, regardless of whether the value of a changes after the operation.

You once dreamed of reaching the unreachable land that only exists in fairy tales. Now that Yuki has a chance to realize this dream, you must help her.

Input

This problem contains multiple test cases.

The first line contains a positive integer t ($1 \le t \le 10^5$), representing the number of test cases.

For each test case:

- A single line contains three integers a, b, m ($0 \le b < m \le a \le 2 \cdot 10^5$).

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

Output

For each test case, output a single line containing an integer representing the answer modulo 998244353.

Example

standard input	standard output
5 5 0 5 5 2 3 10 1 7 10 6 10 100000 114 514	25 1 14 0 837481226

Note

For the first test case:

- One valid operation scheme is to perform modifications in the 3-rd and 4-th rounds.
- Another valid operation scheme is to perform modifications in all rounds from 1 to 5.

For the second test case:

- The only valid operation scheme is to perform a modification in the 3-rd round.

For the fourth test case:

- It can be proven that no valid operation scheme exists.

Problem I. VIP Coupon

Input file: `standard input`
Output file: `standard output`

Yuki has many fond memories of Chang'an, so she visited a shop intending to buy some souvenirs to take home.

The shop sells n souvenirs and m VIP coupons. The price of the i-th souvenir is a_i, and the j-th VIP coupon has a price b_j and a parameter c_j. The effect of a VIP coupon with parameter v is as follows:

- If the price of the item (including souvenirs and other VIP coupons) purchased immediately after buying this coupon is x, the price of that item becomes $\max(x - v, 0)$.

The effect of a VIP coupon is mandatory for the very next purchase and cannot be deferred. Clearly, based on this rule, the effects of VIP coupons cannot be stacked. Each item (including souvenirs and VIP coupons) can be purchased **at most once**; they cannot be bought repeatedly.

Yuki intends to purchase all souvenirs and any number of VIP coupons (possibly zero) in any order. You need to help Yuki find the minimum cost to purchase all the souvenirs.

Input

The input contains multiple test cases.

The first line contains a positive integer t ($1 \le t \le 10^5$), representing the number of test cases.

For each test case:

- The first line contains two positive integers n, m ($1 \le n, m \le 5 \cdot 10^5$).
- The second line contains n integers $a_1, \ldots, a_n$ ($0 \le a_i \le 10^9$).
- The third line contains m integers $b_1, \ldots, b_m$ ($0 \le b_i \le 10^9$).
- The fourth line contains m integers $c_1, \ldots, c_m$ ($0 \le c_i \le 10^9$).

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

Output

For each test case, output a single integer representing the minimum cost to purchase all souvenirs.

Example

standard input	standard output
2 2 4 4 7 1 3 2 4 5 2 6 5 3 3 2 3 8 0 5 2 4 7 5	4 5

Note

For the first test case:

- Yuki can purchase the coupons and souvenirs in the following order: 1st coupon, 1st souvenir, 3rd coupon, 2nd souvenir.
- After the discounts, the price of the 1st souvenir becomes 0, and the price of the 2nd souvenir becomes 1. The total cost is $1 + 0 + 2 + 1 = 4$.

For the second test case:

- Yuki can purchase the items in the following order: 1st souvenir, 1st coupon, 2nd souvenir, 3rd coupon, 2nd coupon, 3rd souvenir.
- After the discounts, the price of the 2nd souvenir becomes 0, the price of the 2nd coupon becomes 0, and the price of the 3rd souvenir becomes 1. The total cost is $2 + 0 + 0 + 2 + 0 + 1 = 5$.

Problem J. Would You Make a Convex?

Input file: `standard input`
Output file: `standard output`

Yuki is the chief judge of the International Convex Polygon Championship (ICPC). He proposed a geometry problem for the competition. However, due to his lack of experience in geometry, he failed to generate correct convex polygon data.

To prove his geometric ability, Yuki started playing with sticks again. He has n sticks, where the length of the i-th stick is a_i. He intends to choose at least 3 sticks from them such that these sticks can form a non-degenerate convex polygon*.

However, since Yuki does not know much about geometry, he does not know how to choose the sticks. As a good friend of Yuki, you need to help him find a subset of sticks such that:

- The subset contains at least 3 sticks, and the number of sticks is as large as possible;
- For **any** subset of at least 3 sticks chosen from this subset, these sticks can form a non-degenerate convex polygon.

Or report that no such subset exists.

*: A non-degenerate convex polygon is a polygon where all side lengths are greater than zero, no three vertices are collinear, and all interior angles are strictly less than 180°.

Input

This problem contains multiple test cases.

The first line contains a positive integer t ($1 \le t \le 10^5$), representing the number of test cases.

For each test case:

- The first line contains a positive integer n ($3 \le n \le 5 \cdot 10^5$).
- The second line contains n positive integers $a_1, \ldots, a_n$ ($1 \le a_i \le 10^9$).

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

Output

For each test case, output one line:

- If no such subset exists, output a single integer 0.
- If such a subset exists, first output an integer k, representing the size of the subset you found, followed by k integers $b_1, \ldots, b_k$, representing the lengths of the k sticks in your subset.

Example

standard input	standard output
3 4 6 9 2 6 3 4 4 9 6 3 1 4 6 5 9	3 6 2 6 0 4 3 4 5 6

Note

For the first test case:

- $\{6, 2, 6\}$ is a valid subset; since $2 + 6 > 6$, these 3 sticks can form a non-degenerate triangle; it is easy to prove that no larger valid subset exists.
- $\{6, 6, 9\}$ is also a valid subset.

For the second test case:

- It can be proven that no valid subset exists.

For the third test case:

- $\{3, 4, 5, 6\}$ is a valid subset; in this case, Yuki has 5 ways to choose sticks: $\{3, 4, 5\}, \{3, 4, 6\}, \{3, 5, 6\}, \{4, 5, 6\}, \{3, 4, 5, 6\}$, and each way can form a non-degenerate convex polygon; it is easy to prove that no larger valid subset exists.
- $\{3, 4, 5, 6, 9\}$ is not a valid subset because the sticks with lengths $3, 4, 9$ cannot form a non-degenerate triangle.
- $\{3, 5, 6\}$ is not a valid subset because there exists a larger valid subset.

Problem K. XOR and LCA

Input file: standard input
Output file: standard output

Yuki has a tree with 2^n nodes, labeled from 0 to $2^n - 1$. The i-th edge connects node u_i and node v_i.

Let $\text{lca}_r(u, v)$ denote the lowest common ancestor of nodes u and v when the tree is rooted at node r. You need to help Yuki calculate:

$$\bigoplus_{0 \le u < v < 2^n} \text{lca}_{u \oplus v}(u, v)$$

where $\oplus$ denotes the bitwise XOR operation.

Input

This problem contains multiple test cases.

The first line contains a positive integer t ($1 \le t \le 10^4$), representing the number of test cases.

For each test case:

- The first line contains a positive integer n ($1 \le n \le 21$).
- The next $2^n - 1$ lines each contain two integers u_i, v_i ($0 \le u_i, v_i < 2^n,\ u_i \ne v_i$).

It is guaranteed that the input forms a tree, and the sum of 2^n over all test cases does not exceed 2^{21}.

Output

For each test case, output a single line containing an integer representing the answer.

Example

standard input	standard output
4 1 0 1 2 0 1 1 2 2 3 3 0 1 0 2 0 3 0 4 0 5 0 6 0 7 3 4 5 2 6 3 7 0 2 1 5 2 7 6 4	1 2 0 4

Note

For the first test case:

- When the tree is rooted at node 1, the lowest common ancestor of nodes 0 and 1 is node 1, so the answer is $\text{lca}_1(0,1) = 1$.

For the second test case:

- We calculate $\text{lca}_{u \oplus v}(u,v)$ for all pairs (u,v):
  - $(0,1)$: $0 \oplus 1 = 1$, $\text{lca}_1(0,1) = 1$.
  - $(0,2)$: $0 \oplus 2 = 2$, $\text{lca}_2(0,2) = 2$.
  - $(0,3)$: $0 \oplus 3 = 3$, $\text{lca}_3(0,3) = 3$.
  - $(1,2)$: $1 \oplus 2 = 3$, $\text{lca}_3(1,2) = 2$.
  - $(1,3)$: $1 \oplus 3 = 2$, $\text{lca}_2(1,3) = 2$.
  - $(2,3)$: $2 \oplus 3 = 1$, $\text{lca}_1(2,3) = 2$.
- The XOR sum is $1 \oplus 2 \oplus 3 \oplus 2 \oplus 2 \oplus 2 = 2$.

Problem L. Yesterday Once More (Easy Version)

Input file:	`standard input`
Output file:	`standard output`

This is the easy version of this problem. The only difference between the easy and hard versions is the limit on the number of moves in your solution.

Yuki lives on a grid with $n+1$ rows and n columns. The rows are numbered 1 to $n+1$ from top to bottom, and the columns are numbered 1 to n from left to right. Let (i, j) denote the cell at row i and column j.

There are $n-1$ obstacles on the grid, and their distribution satisfies the following conditions:

- There are no obstacles in row 1 and row $n+1$.
- For all $2 \le i \le n$, there is **exactly** one obstacle in row i.
- For all $1 \le j \le n$, there is **at most** one obstacle in column j.

Initially, Yuki is at $(1, 1)$. She has heard that a group of kangaroos lives in row $n+1$, so she wants to go to row $n+1$ to see the scenery.

To achieve her goal, Yuki can perform several moves. In each move, she chooses one of the four directions (up, down, left, right) and moves one cell in that direction. Specifically, if the target cell is outside the grid or contains an obstacle, the move is not executed.

Unfortunately, Yuki only knows the rules for the obstacle distribution, not the specific locations of the obstacles. Therefore, she wants you to help her specify the sequence of moves such that for any valid obstacle distribution, Yuki **reaches** row $n+1$ at some point (she only needs to have reached row $n+1$ at least once; she does not need to remain there after all moves are completed).

Since Yuki is not in a hurry, the number of moves in your solution must not exceed $\mathbf{30 \cdot n}$.

Input

A single line contains a positive integer n ($2 \le n \le 10^3$).

Output

The first line output an integer k ($1 \le k \le 30 \cdot n$), representing the number of moves in your solution.

The second line output a string s of length k, where s_i represents the direction of Yuki's i-th move:

- If $s_i =$ `U`, Yuki moves up.
- If $s_i =$ `D`, Yuki moves down.
- If $s_i =$ `L`, Yuki moves left.
- If $s_i =$ `R`, Yuki moves right.

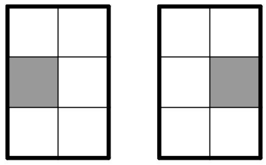
Examples

standard input	standard output
2	4 DRDD
3	17 DDDUUURDDDUUURDDD

Note

For the first example:

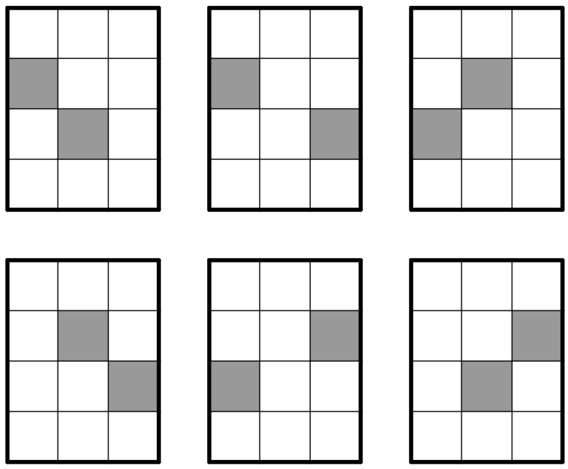
- Let gray cells represent obstacles and white cells represent empty cells. The following image shows all possible obstacle distributions satisfying the requirements:



- For the first obstacle distribution, Yuki's path is $(1,1) \to (1,1) \to (1,2) \to (2,2) \to (3,2)$.
- For the second obstacle distribution, Yuki's path is $(1,1) \to (2,1) \to (2,1) \to (3,1) \to (3,1)$.
- For every valid obstacle distribution, Yuki reaches row $n+1$, so the sample output is correct.

For the second example:

- Let gray cells represent obstacles and white cells represent empty cells. The following image shows all possible obstacle distributions satisfying the requirements:



- It is easy to prove that for any of these obstacle distributions, following the moves given in the sample output, Yuki will reach row $n+1$.

Problem M. Yesterday Once More (Hard Version)

Input file: `standard input`
Output file: `standard output`

This is the hard version of this problem. The only difference between the easy and hard versions is the limit on the number of moves in your solution.

Yuki lives on a grid with $n+1$ rows and n columns. The rows are numbered 1 to $n+1$ from top to bottom, and the columns are numbered 1 to n from left to right. Let (i, j) denote the cell at row i and column j.

There are $n-1$ obstacles on the grid, and their distribution satisfies the following conditions:

- There are no obstacles in row 1 and row $n+1$.
- For all $2 \le i \le n$, there is **exactly** one obstacle in row i.
- For all $1 \le j \le n$, there is **at most** one obstacle in column j.

Initially, Yuki is at $(1, 1)$. She has heard that a group of kangaroos lives in row $n+1$, so she wants to go to row $n+1$ to see the scenery.

To achieve her goal, Yuki can perform several moves. In each move, she chooses one of the four directions (up, down, left, right) and moves one cell in that direction. Specifically, if the target cell is outside the grid or contains an obstacle, the move is not executed.

Unfortunately, Yuki only knows the rules for the obstacle distribution, not the specific locations of the obstacles. Therefore, she wants you to help her specify the sequence of moves such that for any valid obstacle distribution, Yuki **reaches** row $n+1$ at some point (she only needs to have reached row $n+1$ at least once; she does not need to remain there after all moves are completed).

Since Yuki is in a hurry, the number of moves in your solution must not exceed $\mathbf{10 \cdot n}$.

Input

A single line contains a positive integer n ($2 \le n \le 10^3$).

Output

The first line output an integer k ($1 \le k \le 10 \cdot n$), representing the number of moves in your solution.

The second line output a string s of length k, where s_i represents the direction of Yuki's i-th move:

- If $s_i =$ `U`, Yuki moves up.
- If $s_i =$ `D`, Yuki moves down.
- If $s_i =$ `L`, Yuki moves left.
- If $s_i =$ `R`, Yuki moves right.

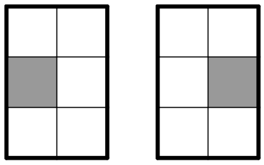
Examples

standard input	standard output
2	4 DRDD
3	17 DDDUUURDDDUUURDDD

Note

For the first example:

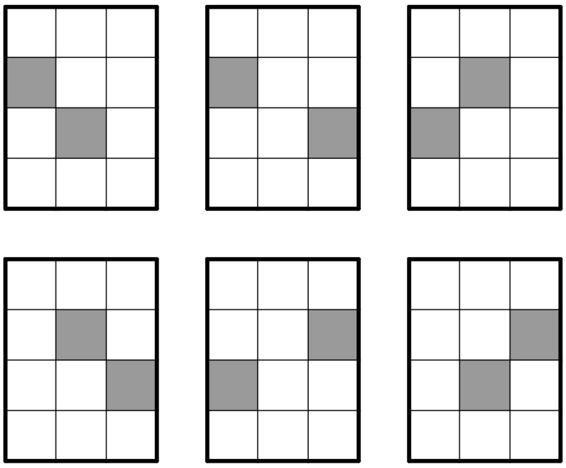
- Let gray cells represent obstacles and white cells represent empty cells. The following image shows all possible obstacle distributions satisfying the requirements:



- For the first obstacle distribution, Yuki's path is $(1,1) \to (1,1) \to (1,2) \to (2,2) \to (3,2)$.
- For the second obstacle distribution, Yuki's path is $(1,1) \to (2,1) \to (2,1) \to (3,1) \to (3,1)$.
- For every valid obstacle distribution, Yuki reaches row $n+1$, so the sample output is correct.

For the second example:

- Let gray cells represent obstacles and white cells represent empty cells. The following image shows all possible obstacle distributions satisfying the requirements:



- It is easy to prove that for any of these obstacle distributions, following the moves given in the sample output, Yuki will reach row $n+1$.

Problem N. Zebra Crossing

Input file: `standard input`
Output file: `standard output`

In front of Yuki is a strange zebra crossing.

This zebra crossing can be viewed as a tree with n nodes. The i-th edge connects node u_i and node v_i. Each node is colored either black or white, described by a binary string s of length n:

- If $s_i = \texttt{0}$, the color of node i is black.
- If $s_i = \texttt{1}$, the color of node i is white.

Yuki has a jumping ability k, which means that when she is at node x, she can jump to any node y such that $\text{dist}(x, y) \le k$. Here, $\text{dist}(x, y)$ denotes the number of edges on the simple path between node x and node y.

Next, Yuki will perform $n - 1$ rounds of jumping on the zebra crossing. In the i-th round, Yuki starts at node 1 and wants to reach node $i + 1$ through a sequence of jumps. At the same time, Yuki wants to minimize the number of times she lands on a black node after her jumps.

You need to help Yuki find the minimum number of times she lands on a black node for each round of jumping.

Input

This problem contains multiple test cases.

The first line contains a positive integer t ($1 \le t \le 10^5$), representing the number of test cases.

For each test case:

- The first line contains two positive integers n, k ($1 \le n \le 5 \cdot 10^5,\ 1 \le k \le n$).
- The second line contains a binary string s of length n ($s_i \in \{\texttt{0}, \texttt{1}\}$).
- The next $n - 1$ lines each contain two positive integers u_i, v_i ($1 \le u_i, v_i \le n,\ u_i \ne v_i$).

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

Output

For each test case, output a single line containing $n - 1$ integers, where the i-th integer represents the minimum number of times Yuki lands on a black node during the i-th round of jumping.

Example

standard input	standard output
2 5 1 01010 3 5 2 1 1 3 3 4 9 3 100010000 1 2 2 3 2 4 3 5 3 6 4 7 6 8 7 9	0 1 1 2 1 1 1 0 1 1 1 2

Note

For the first test case:

- For the 1-st round of jumping, one valid sequence of visited nodes is $1, 2$.
- For the 4-th round of jumping, one valid sequence of visited nodes is $1, 3, 5$.

For the second test case:

- For the 4-th round of jumping, one valid sequence of visited nodes is $1, 5$.
- For the 7-th round of jumping, one valid sequence of visited nodes is $1, 5, 8$.
- For the 8-th round of jumping, one valid sequence of visited nodes is $1, 4, 9$.