A - Partition

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 300 points

Problem Statement

You are given integers N and K.

The **cumulative sums** of an integer sequence $X = (X_1, X_2, \dots, X_N)$ of length N is defined as a sequence $Y = (Y_0, Y_1, \dots, Y_N)$ of length $N + 1$ as follows:

- $Y_0 = 0$
- $Y_i = \sum_{j=1}^{i} X_j \ (i = 1, 2, \dots, N)$

An integer sequence $X = (X_1, X_2, \dots, X_N)$ of length N is called a **good sequence** if and only if it satisfies the following condition:

- Any value in the cumulative sums of X that is less than K appears before any value that is not less than K.
  - Formally, for the cumulative sums Y of X, for any pair of integers (i, j) such that $0 \le i, j \le N$, if $(Y_i < K$ and $Y_j \ge K)$, then $i < j$.

You are given an integer sequence $A = (A_1, A_2, \dots, A_N)$ of length N. Determine whether the elements of A can be rearranged to a good sequence. If so, print one such rearrangement.

Constraints

- $1 \le N \le 2 \times 10^5$
- $-10^9 \le K \le 10^9$
- $-10^9 \le A_i \le 10^9$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

```
N K
A_1 A_2 ... A_N
```

Output

If the elements of A can be rearranged to a good sequence, print the rearranged sequence $(A'_1, A'_2, \ldots, A'_N)$ in the following format:

```
Yes
A'_1 A'_2 ... A'_N
```

If there are multiple valid rearrangements, any of them is considered correct.

If a good sequence cannot be obtained, print `No`.

Sample Input 1

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

Sample Output 1

```
Yes
-3 -1 2 4
```

If you rearrange A to $(-3, -1, 2, 4)$, the cumulative sums Y in question will be $(0, -3, -4, -2, 2)$. In this Y, any value less than 1 appears before any value not less than 1.

Sample Input 2

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

Sample Output 2

```
No
```

Sample Input 3

```
10 1000000000
-1000000000 -1000000000 -1000000000 -1000000000 -1000000000 1000000000 1000000000 1000000000 1000000000 1000000000
```

Sample Output 3

```
Yes
-1000000000 -1000000000 -1000000000 -1000000000 -1000000000 1000000000 1000000000 1000000000 1000000000 1000000000
```

B - Between B and B

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 500 points

Problem Statement

You are given a sequence $(X_1, X_2, \dots, X_M)$ of length M consisting of integers between 1 and M, inclusive.

Find the number, modulo 998244353, of sequences $A = (A_1, A_2, \dots, A_N)$ of length N consisting of integers between 1 and M, inclusive, that satisfy the following condition:

- For each $B = 1, 2, \dots, M$, the value X_B exists between any two different occurrences of B in A (including both ends).

More formally, for each $B = 1, 2, \dots, M$, the following condition must hold:

- For every pair of integers (l, r) such that $1 \leq l < r \leq N$ and $A_l = A_r = B$, there exists an integer m ($l \leq m \leq r$) such that $A_m = X_B$.

Constraints

- $1 \leq M \leq 10$
- $1 \leq N \leq 10^4$
- $1 \leq X_i \leq M$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
M \quad N \\
X_1 \quad X_2 \quad \dots \quad X_M
\end{array}
$$

Output

Print the answer.

Sample Input 1

```
3 4
2 1 2
```

Sample Output 1

```
14
```

Here are examples of sequences A that satisfy the condition:

- $(1, 3, 2, 3)$
- $(2, 1, 2, 1)$
- $(3, 2, 1, 3)$

Here are non-examples:

- $(1, 3, 1, 3)$
  - There is no $X_3 = 2$ between the 3s.
- $(2, 2, 1, 3)$
  - There is no $X_2 = 1$ between the 2s.

Sample Input 2

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

Sample Output 2

```
65536
```

All sequences of length 8 consisting of integers between 1 and 4 satisfy the condition.

Note that when $X_B = B$, there is always a B between two different occurrences of B.

Sample Input 3

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

Sample Output 3

```
628
```

C - Beware of Overflow

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 500 points

Problem Statement

This is an **interactive problem** (where your program interacts with the judge via input and output).

You are given a positive integer N.

The judge has a hidden positive integer R and N integers $A_1, A_2, \ldots, A_N$. It is guaranteed that $|A_i| \leq R$ and $\left|\sum_{i=1}^{N} A_i\right| \leq R$.

There is a blackboard on which you can write integers with absolute values not exceeding R. Initially, the blackboard is empty.

The judge has written the values $A_1, A_2, \ldots, A_N$ on the blackboard **in this order**. Your task is to make the blackboard contain only one value $\sum_{i=1}^{N} A_i$.

You cannot learn the values of R and A_i directly, but you can interact with the judge up to 25000 times.

For a positive integer i, let X_i be the i-th integer written on the blackboard. Specifically, $X_i = A_i$ for $i = 1, 2, \ldots, N$.

In one interaction, you can specify two distinct positive integers i and j and choose one of the following actions:

- Perform addition. The judge will erase X_i and X_j from the blackboard and write $X_i + X_j$ on the blackboard.
  - $|X_i + X_j| \leq R$ must hold.
- Perform comparison. The judge will tell you whether $X_i < X_j$ is true or false.

Here, at the beginning of each interaction, the i-th and j-th integers written on the blackboard must not have been erased.

Perform the interactions appropriately so that after all interactions, the blackboard contains only one value $\sum_{i=1}^{N} A_i$.

The values of R and A_i are determined before the start of the conversation between your program and the judge.

Constraints

- $2 \leq N \leq 1000$
- $1 \leq R \leq 10^9$
- $|A_i| \leq R$
- $\left|\sum_{i=1}^{N} A_i\right| \leq R$
- N, R, and A_i are integers.

Input and Output

This is an **interactive problem** (where your program interacts with the judge via input and output).

First, read N from Standard Input.

```
N
```

Next, repeat the interactions until the blackboard contains only one value $\sum_{i=1}^{N} A_i$.

When performing addition, make an output in the following format to Standard Output. Append a newline at the end. Here, i and j are distinct positive integers.

```
+ i j
```

The response from the judge will be given from Standard Input in the following format:

```
P
```

Here, P is an integer:

- If $P \geq N+1$, it means that the value $X_i + X_j$ has been written on the blackboard, and it is the P-th integer written.
- If $P = -1$, it means that i and j do not satisfy the constraints, or the number of interactions has exceeded 25000.

When performing comparison, make an output in the following format to Standard Output. Append a newline at the end. Here, i and j are distinct positive integers.

```
? i j
```

The response from the judge will be given from Standard Input in the following format:

```
Q
```

Here, Q is an integer:

- If $Q = 1$, it means that $X_i < X_j$ is true.
- If $Q = 0$, it means that $X_i < X_j$ is false.
- If $Q = -1$, it means that i and j do not satisfy the constraints, or the number of interactions has exceeded 25000.

For both types of interactions, if the judge's response is -1, your program is already considered incorrect. In this case, terminate your program immediately.

When the blackboard contains only one value $\sum_{i=1}^{N} A_i$, report this to the judge in the following format. This does not count towards the number of interactions. Then, terminate your program immediately.

```
!
```

If you make an output in a format that does not match any of the above, `-1` will be given from Standard Input.

```
-1
```

In this case, your program is already considered incorrect. Terminate your program immediately.

Notes

- **For each output, append a newline at the end and flush Standard Output. Otherwise, the verdict may be TLE.**
- Terminate your program immediately after outputting the result (or receiving `-1`). Otherwise, the verdict will be indeterminate.
- Extra newlines will be considered as malformed output.

Sample Input and Output

Here is a possible conversation with $N = 3, R = 10, A_1 = -1, A_2 = 10, A_3 = 1$.

Input	Output	Explanation
`3`		First, the integer N is given.
	`? 1 2`	Perform a comparison.
`1`		The judge returns 1 because $X_1 < X_2$ $(-1 < 10)$.
	`+ 1 3`	Perform an addition.
`4`		The judge erases $X_1 = -1$ and $X_3 = 1$ from the blackboard and writes $X_1 + X_3 = 0$. This is the fourth integer written.
	`+ 2 4`	Perform an addition.
`5`		The judge erases $X_2 = 10$ and $X_4 = 0$ from the blackboard and writes $X_2 + X_4 = 10$. This is the fifth integer written.
	`!`	The blackboard contains only one value $\sum_{i=1}^{N} A_i$, so report this to the judge.

D - Portable Gate

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 700 points

Problem Statement

You are given a tree with N vertices numbered $1, 2, \ldots, N$. The i-th edge connects vertices u_i and v_i bidirectionally.

Initially, all vertices are painted white.

To efficiently visit all vertices of this tree, Alice has invented a magical gate. She uses one piece and one gate to travel according to the following procedure.

First, she chooses a vertex and places both the piece and the gate on that vertex. Then, she repeatedly performs the following operations until all vertices are painted black.

- Choose one of the following actions:
    1. Paint the vertex where the piece is placed black.
    2. Choose a vertex adjacent to the vertex where the piece is placed and move the piece to that vertex. The cost of this action is 1.
    3. Move the piece to the vertex where the gate is placed.
    4. Move the gate to the vertex where the piece is placed.

Note that only the second action incurs a cost.

It can be proved that it is possible to paint all vertices black in a finite number of operations. Find the minimum total cost required.

Constraints

- $2 \leq N \leq 2 \times 10^5$
- $1 \leq u_i, v_i \leq N$
- The given graph is a tree.
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
u_1 \quad v_1 \\
\vdots \\
u_{N-1} \quad v_{N-1}
\end{array}
$$

Output

Print the answer.

Sample Input 1

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

Sample Output 1

```
3
```

Here is an example of Alice's procedure. Let (u, v) denote the state where the piece is at vertex u and the gate is at vertex v.

- Place the piece and the gate at vertex 4.
  - The state is now $(4, 4)$.
- Perform action 1.
  - Vertex 4 is painted black.
  - The state is now $(4, 4)$.
- Perform action 2 and move the piece to vertex 1.
  - This costs 1.
  - The state is now $(1, 4)$.
- Perform action 1.
  - Vertex 1 is painted black.
- Perform action 4.
  - The state is now $(1, 1)$.
- Perform action 2 and move the piece to vertex 2.
  - This costs 1.
  - The state is now $(2, 1)$.
- Perform action 1.
  - Vertex 2 is painted black.
- Perform action 3.
  - The state is now $(1, 1)$.
- Perform action 2 and move the piece to vertex 3.
  - This costs 1.
  - The state is now $(3, 1)$.
- Perform action 1.
  - Vertex 3 is painted black.
  - All vertices are now painted black, so the procedure ends.

The total cost of performing action 2 is 3, and there is no procedure with a smaller cost.

Sample Input 2

```
10
1 7
7 10
10 8
8 3
8 4
10 9
9 6
9 5
7 2
```

Sample Output 2

```
10
```

E - Rectangle Concatenation

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 800 points

Problem Statement

For positive integers h and w, let (h, w) denote a rectangle with height h and width w. In this problem, we do not consider rotating the rectangles, and the rectangles (h, w) and (w, h) are distinguished when $h \neq w$.

A sequence of rectangles $((h_1, w_1), (h_2, w_2), \dots, (h_n, w_n))$ is called a **rectangle generation sequence** if there exists a method that successfully follows the steps below:

- Let the rectangle X be (h_1, w_1). Hereafter, let H and W respectively denote the height and width of the rectangle X at each step.
- For $i = 2, 3, \dots, n$, perform one of the following operations. If neither can be performed, the procedure unsuccessfully terminates.
  - If the height of X is equal to h_i, attach the rectangle (h_i, w_i) horizontally to X. Formally, if $H = h_i$ at that time, replace X with the rectangle $(H, W + w_i)$.
  - If the width of X is equal to w_i, attach the rectangle (h_i, w_i) vertically to X. Formally, if $W = w_i$ at that time, replace X with the rectangle $(H + h_i, W)$.
- If the above series of operations does not fail, the procedure successfully terminates.

You are given N rectangles. The i-th rectangle has a height of H_i and a width of W_i.

Find the number of pairs of positive integers (l, r) that satisfy $1 \leq l \leq r \leq N$ and the following condition:

- The sequence of rectangles $((H_l, W_l), (H_{l+1}, W_{l+1}), \dots, (H_r, W_r))$ is a rectangle generation sequence.

Constraints

- $1 \leq N \leq 3 \times 10^5$
- $1 \leq H_i, W_i \leq 10^6$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{aligned}
&N \\
&H_1 \quad W_1 \\
&H_2 \quad W_2 \\
&\vdots \\
&H_N \quad W_N
\end{aligned}
$$

Output

Print the answer.

Sample Input 1

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

Sample Output 1

```
7
```

The pairs (l, r) that satisfy the condition are $(1, 1), (1, 2), (2, 2), (2, 3), (2, 4), (3, 3), (4, 4)$; there are seven.

For example, for $(l, r) = (2, 4)$, the procedure succeeds if the first attachment is done vertically and the second is done horizontally.

Sample Input 2

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

Sample Output 2

```
10
```

Sample Input 3

```
1
1000000 1000000
```

Sample Output 3

```
1
```

Sample Input 4

```
10
1 1
1 1
1 1
1 1
1 1
1 1
1 1
1 1
1 1
1 1
```

Sample Output 4

```
55
```

F - All the Same

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 1000 points

Problem Statement

You are given a string S of length N consisting of the characters A and B.

For a string X of length N consisting of the characters 1, 2, and 3, the **score** is determined by the following procedure:

> First, initialize the variables h_1, h_2, h_3, P to 0.
>
> Then, for $i = 1, 2, \ldots, N$ in this order, perform the following operations:
>
> - If the i-th character of S is A, perform operation A; if it is B, perform operation B. Let d be the number represented by the i-th character of X.
>   - **Operation A**: Add 2 to h_d.
>   - **Operation B**: If $d = 2$ or $h_d \neq h_2$, set P to -10^{100}. Otherwise, add 1 to both h_d and h_2.
> - If $h_1 = h_2 = h_3$, add 1 to P.
>
> The final value of P is the score.

Print one string X that maximizes the score.

You have T test cases to solve.

Constraints

- $1 \leq T \leq 10^5$
- $1 \leq N \leq 10^6$
- S is a string of length N consisting of A and B.
- T and N are integers.
- The sum of N across all test cases is at most 10^6.

Input

The input is given from Standard Input in the following format. Here, test_i denotes the i-th test case.

$$
\begin{array}{l}
T \\
\mathrm{test}_1 \\
\mathrm{test}_2 \\
\vdots \\
\mathrm{test}_T
\end{array}
$$

Each test case is given in the following format:

$$
\begin{array}{l}
N \\
S
\end{array}
$$

Output

Print T lines.

The i-th line $(1 \leq i \leq T)$ should contain a string X that maximizes the score for the i-th test case.

If multiple strings X maximize the score, any of them is considered correct.

Sample Input 1

```
5
4
ABBA
5
AAAAA
6
BBBBBB
7
ABABABA
20
AAABBBBBBBBAAABBBABA
```

Sample Output 1

```
1333
12321
333333
1313212
333111111133121111311
```

Let us describe the changes in (h_1, h_2, h_3, P) as we proceed with $i = 1, 2, \ldots, N$.

- For the first test case, $(0,0,0,0) \to (2,0,0,0) \to (2,1,1,0) \to (2,2,2,1) \to (2,2,4,1)$. The maximum score is 1.
- For the second test case, $(0,0,0,0) \to (2,0,0,0) \to (2,2,0,0) \to (2,2,2,1) \to (2,4,2,1) \to (4,4,2,1)$. The maximum score is 1.

For the third, fourth, and fifth test cases, the maximum scores are 0, 0, and 2, respectively. There may be multiple strings X that maximize the score, but you only need to print one of them.