

Busy Beaver's Dam Logs

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

Busy Beaver has arranged N logs in a line to form the base of his dam. Each log has an integer quality value between -2 and 2 . From time to time, some logs are replaced. He occasionally wants to know if there is a contiguous segment of logs whose total quality equals a given **nonzero** number X , and may also want to know the segment. Help Busy Beaver process all updates and queries efficiently.

Input

The first line of input contains an integer T ($1 \leq T \leq 10^4$) — the number of testcases.

The first line of each test case contains two integers N and Q ($1 \leq N, Q \leq 2 \cdot 10^5$) — the size of the array and the number of queries, respectively.

The second line of each test case contains N integers a_1, a_2, \dots, a_N ($-2 \leq a_i \leq 2$) — the qualities of the logs.

The next Q lines of each test case are of two forms:

- 1 i x ($1 \leq i \leq N$, $-2 \leq x \leq 2$) — set the quality of the i -th log to x .
- 2 X ($-2N \leq X \leq 2N$, $X \neq 0$) — find a contiguous segment of logs with total quality X .

There is at least one query of the second type.

The sum of N over all test cases does not exceed $2 \cdot 10^5$.

The sum of Q over all test cases does not exceed $2 \cdot 10^5$.

Output

For each query of the second type, if a valid contiguous segment exists, print **YES** on the first line, followed by the left and right indices of that segment l and r on the next line ($1 \leq l \leq r \leq N$). If no such segment exists, print **NO** on a single line.

You can output the answer in any case (upper or lower). For example, the strings “yEs”, “yes”, “Yes”, and “YES” will be recognized as positive responses.

Scoring

There are four subtasks in this problem. You can receive 50% of the points for each subtask if the YES/NO answers are correct, but the indices are incorrect. **To obtain these partial points, you must still output integers l and r between 1 and N after all YES answers.**

- (10 points): The sum of N over all test cases is at most 10^3 , and the sum of Q over all test cases is at most 10^3 .
- (20 points): $-1 \leq a_i \leq 1$ for all i , and for any query of the first type, $-1 \leq x \leq 1$.
- (30 points): $0 \leq a_i \leq 2$ for all i , and for any query of the first type, $0 \leq x \leq 2$.
- (40 points): No additional constraints.

Example

standard input	standard output
4	YES
5 5	2 2
-1 2 0 -2 1	YES
1 2 1	3 5
2 1	NO
1 4 2	YES
2 3	2 2
2 -2	NO
4 3	YES
-1 -1 -1 -1	1 1
1 4 1	YES
2 -1	2 4
2 -4	NO
6 6	YES
1 -1 1 -1 1 -1	1 1
1 3 0	NO
2 1	YES
2 -2	1 1
2 3	
1 3 2	
2 1	
6 3	
0 0 0 0 0 0	
2 2	
1 1 1	
2 1	

Note

For the first test case:

- The initial array is $[-1, 2, 0, -2, 1]$.
- After the first query 1 2 1, the array becomes $[-1, 1, 0, -2, 1]$.
- The second query 2 1 asks for a contiguous segment summing to 1, which can be achieved by [1] (element 2).
- After the third query 1 4 2, the array becomes $[-1, 1, 0, 2, 1]$.
- The fourth query 2 3 asks for a contiguous segment summing to 3, which can be achieved by [1, 0, 2] (elements 2-4).
- The fifth query 2 -2 asks for a contiguous segment summing to -2. It can be checked that no such subarray exists.

For the second test case:

- The initial array is $[-1, -1, -1, -1]$.
- After the first query 1 4 1, the array becomes $[-1, -1, -1, 1]$.
- The second query 2 -1 asks for a contiguous segment summing to -1, which can be achieved by [-1] (any of the first three elements).

- The third query $2 -4$ asks for a contiguous segment summing to -4 . It can be checked that no such subarray exists.