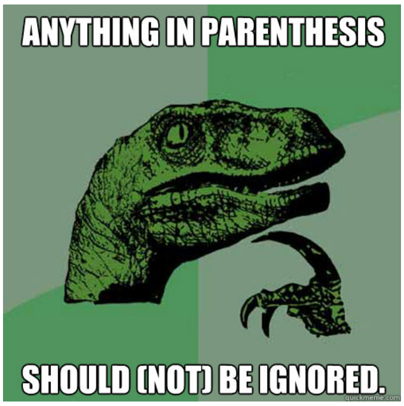
Quantum Brackets (brackets)

Dario is experimenting with a new kind of brackets: the *quantum* brackets. A *quantum* bracket is in a *superposition* of (meaning, being at the same time) open and closed, and will collapse to either option when observed. Dario is working with K kinds of *quantum* brackets and he managed to get a sequence of N of those. He is wondering whether there exists a way in which they can *collapse* (meaning, become definitely open or closed) to form a well-parenthesized expression.



An expression of K kinds of brackets is well-parenthesized if it is possible to one-to-one match each bracket with one of the same kind such that the first is open, the second is closed and taken any two intervals between matched brackets they are either disjoint or contained in one another.

☞ Among the attachments of this task you may find a template file `brackets.*` with a sample incomplete implementation.

Input

The first line contains an integer T, the number of test cases in the input file. Then $2T$ lines follow, describing each test case, one after the other. The j-th test case is described by two lines: the first one contains two integers N_j and K_j, the number of brackets and the number of kinds of brackets, respectively; the second line contains N_j integers P_{ji}, the kinds of the N_j brackets.

Output

You need to write T lines, one for each test case. If there is a well-parenthesized collapse for the *quantum* brackets print `1`, otherwise `0`.

Constraints

- $1 \leq T \leq 10$.
- $\sum_{j=1}^{T} N_j \leq 400\,000$.
- $1 \leq K_j \leq N_j$ for each $j = 1 \ldots T$.
- $0 \leq P_{ji} < K_j$ for each $i = 1 \ldots N_j$, $j = 1 \ldots T$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 1** (0 points) Examples.
- **Subtask 2** (20 points) $\sum_{j=1}^{T} N_j \leq 20$.
- **Subtask 3** (15 points) $K_j = 1$ for $j = 1 \ldots T$.
- **Subtask 4** (30 points) $\sum_{j=1}^{T} N_j \leq 3000$.
- **Subtask 5** (35 points) No additional limitations.

Examples

input	output
3 8 2 0 1 0 0 1 1 1 0 6 3 2 1 0 2 0 1 16 4 0 1 0 2 3 3 2 1 1 0 0 1 1 0 1 0	1 0 1

Explanation

In the **first example** we have 3 cases.

In the *first case*, if we associate 0 with round brackets and 1 with square brackets, the quantum bracket can collapse to the following expression: ([( )] [ ]) . This way the first bracket is matched with the eighth one, the second with the fifth one, the third with the fourth one, and lastly the sixth with the seventh. All intervals bounded by pairs of corresponding brackets are disjoint or contained in one another. Therefore the solution is 1.

In the *second case* the quantum bracket can never collapse to a well-parenthesized expression. Indeed, since there are only two brackets of each kind, they necessarily need to be matched with one another. Now if we associate 0 with round brackets, 1 with square brackets, and 2 with curly brackets, and impose that for each pair of brackets, the first one should be open while the second should be closed, our quantum bracket collapses to: ([{) }] . From this expression we can easily notice that the second condition is not satisfied: the interval bounded by the round brackets overlaps the interval bounded by the square brackets, and this happens taking into account curly brackets and any other pair of brackets as well. Thus the solution is 0.

In the *third case*, if we associate 0 with round brackets, 1 with square brackets, 2 with curly brackets, and 3 with angle brackets, the quantum bracket can collapse to: ([({ < > } [[( )]])]) . Every open bracket has a corresponding closed bracket of the same type and all intervals bounded by pairs of corresponding brackets are disjoint or contained in one another. Therefore the solution is 1.