# Problem H. Harder Satisfiability

Time limit: 3 seconds

A *fully quantified* boolean 2-CNF formula is a formula in the following form: $Q_1 x_1 \ldots Q_n x_n \, F(x_1, \ldots, x_n)$. Each $Q_i$ is one of two quantifiers: a *universal* quantifier $\forall$ ("for all"), or an *existential* quantifier $\exists$ ("exists"); and $F$ is a conjunction (boolean `AND`) of $m$ clauses $s \vee t$ (boolean `OR`), where $s$ and $t$ are some variables (not necessarily different) with or without negation. This formula has no free variables, so it evaluates to either `true` or `false`. We can evaluate a given fully quantified formula with a simple recursive algorithm:

1. If there are no quantifiers, return the remaining expression's value of `true` or `false`.
2. Otherwise, recursively evaluate formulas: $F_z = Q_2 x_2 \ldots Q_n x_n \, F(z, x_2, \ldots, x_n)$ for $z = 0, 1$.
3. If $Q_1 = \exists$ return $F_0 \vee F_1$; otherwise if $Q_1 = \forall$ return $F_0 \wedge F_1$.

You are given some fully quantified boolean 2-CNF formulas. Find out if they are true or not.

## Input

The first line of the input contains a single integer $t$ ($1 \le t \le 10^5$) — the number of test cases.

The first line of a test case contains two integers $n$ and $m$ ($1 \le n, m \le 10^5$) — the number of variables and the number of clauses in $F$. The next line contains a string $s$ with $n$ characters describing the quantifiers. If $s_i =$ '`A`' then $Q_i$ is a universal quantifier $\forall$, otherwise if $s_i =$ '`E`' then $s_i$ is an existential quantifier $\exists$.

Next $m$ lines describe clauses in $F$. Each line contains two integers $u_i$ and $v_i$ ($-n \le u_i, v_i \le n$; $u_i, v_i \ne 0$). If $u_i \ge 1$ then the first variable in the $i$-th clause is $x_{u_i}$. Otherwise, if $u_i \le -1$ then the first variable is $\overline{x_{-u_i}}$ (negation of $x_{-u_i}$). The second variable in the $i$-th clause is similarly described by $v_i$.

The sum of values of $n$ for all test cases does not exceed $10^5$; the sum of values of $m$ does not exceed $10^5$.

## Output

For each test case output "`TRUE`" if the given formula is true or "`FALSE`" otherwise.

## Example

| standard input | standard output |
|---|---|
| 3<br>2 2<br>AE<br>1 -2<br>-1 2<br>2 2<br>EA<br>1 -2<br>-1 2<br>3 2<br>AEA<br>1 -2<br>-1 -3 | TRUE<br>FALSE<br>FALSE |

## Note

The first sample corresponds to a formula $\forall x_1 \, \exists x_2 \, (x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2) = \forall x_1 \, \exists x_2 \; x_1 \oplus x_2$. For any $x_1$ we can choose $x_2 = \overline{x_1}$ making it true, hence the formula is true.

The second sample changes the order of quantifiers. Now the answer is "`FALSE`", because for any value of $x_1$ we can choose $x_2 = x_1$ and the formula becomes false.

The third formula is $\forall x_1 \, \exists x_2 \, \forall x_3 \, (x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3})$. If we substitute $x_1 = 1$, $x_3 = 1$ then no assignment of $x_2$ can make the second clause true, so the formula is false.