We have v boolean variables $x_0, x_1, ..., x_{v-1}$ that x_i can be either 1 ($true$) or 0 ($false$).
Let's define an environment env, an assignment of either 0 or 1 to all v variables (like $v = 3,\ env = \{x_0 = 0, x_1 = 1, x_2 = 0\}$).
Also let's define an expression exp a string which can be generated by following grammar:$\langle expression\rangle ::= x_0\ j\ x_1\ j\ ...\ jx_{v-1}\ j\ \langle operation\rangle$
$\langle operation\rangle ::= (not\ \langle expression\rangle)$
$j\ (\langle expression\rangle\ and\ \langle expression\rangle)$
$j\ (\langle expression\rangle\ or\ \langle expression\rangle)$
$j\ (\langle expression\rangle\ xor\ \langle expression\rangle)$
For example $((x_0\ xor\ x_1)\ or\ x_2)$ is an expression.
Each expression has a value in an environment.(for example expression $((x_0\ xor\ x_1)\ or\ x_2)$ has value 0 in environment $\{x_0 = 1, x_1 = 1, x_2 = 0\}$ but it has value 1 in enviroment $\{x_0 = 0, x_1 = 0, x_2 = 1\}$).
You are given n environments $env_1, env_2, ..., env_n$ and their expected value $y_1, y_2, ..., y_n$ (the value that the expressions should has in them), find an expression which satisfy enviroments as much as possible and uses operations (and, or, not, xor) as least as possible, in other words for each of 20 tests your score will be:

- Let exp be your expression.
- Let cnt be number of environments like env_i that $exp\ in\ env_i = y_i$.
- $score = \frac{e^{(\frac{cnt}{n})^2}-1}{e-1} \times (1 - \frac{number\ of\ used\ operations}{1.5\times n\times v}) \times 5$

Input
First line contains two integers v and n separated by space.
Each of the following n lines contains env_i and y_i separated by space, env_i is a string containing 0 and 1s of length v, jth character in env_i is value of v_j in env_i.

- $1 \le n \le 100$
- In 7 tests: $7 \le v \le 10$
- In 3 tests: $10 < v \le 20$
- In other tests: $30 \le v \le 50$

Output
Print a single line the expression.

Notes:

- The expression should not contain any space.
- Print i instead of x_i.

- Print following characters instead of operations:
    - & for *and*.
    - |for *or*.
    - ! for *not*.
    - ˆfor *xor*.
- For example $((x_2\ \mathit{xor}\ (\mathit{not}\ x_1))\ \mathit{or}\ x_0)$ should be printed as ((2 ˆ(!1))|0).
- You can use at most $1.5 \times n \times v$ operations.
- The expression's length can be at most 3×10^4.