ACM International Collegiate Programming Contest 97/98

Sponsored by IBM

Central European Regional Contest

Problem B: King

Input file: `king.in`
Output file: `king.out`
Program file: `king.pas` or `king.cpp`

Once, in one kingdom, there was a queen and that queen was expecting a baby. The queen prayed: "If my child was a son and if only he was a sound king." After nine months her child was born, and indeed, she gave birth to a nice son.

Unfortunately, as it used to happen in royal families, the son was a little retarded. After many years of study he was able just to add integer numbers and to compare whether the result is greater or less than a given integer number. In addition, the numbers had to be written in a sequence and he was able to sum just continuous subsequences of the sequence.

The old king was very unhappy of his son. But he was ready to make everything to enable his son to govern the kingdom after his death. With regards to his son's skills he decided that every problem the king had to decide about had to be presented in a form of a finite sequence of integer numbers and the decision about it would be done by stating an integer constraint (i.e. an upper or lower limit) for the sum of that sequence. In this way there was at least some hope that his son would be able to make some decisions.

After the old king died, the young king began to reign. But very soon, a lot of people became very unsatisfied with his decisions and decided to dethrone him. They tried to do it by proving that his decisions were wrong.

Therefore some conspirators presented to the young king a set of problems that he had to decide about. The set of problems was in the form of subsequences $S_i = \{a_{s_i}, a_{s_i+1}, \ldots, a_{s_i+n_i}\}$ of a sequence $S = \{a_1, a_2, \ldots, a_n\}$. The king thought a minute and then decided, i.e. he set for the sum $a_{s_i}+a_{s_i+1}+\ldots+a_{s_i+n_i}$ of each subsequence S_i an integer constraint k_i (i.e. $a_{s_i}+a_{s_i+1}+\ldots+a_{s_i+n_i} < k_i$ or $a_{s_i}+a_{s_i+1}+\ldots+a_{s_i+n_i} > k_i$ resp.) and declared these constraints as his decisions.

After a while he realized that some of his decisions were wrong. He could not revoke the declared constraints but trying to save himself he decided to fake the sequence that he was given. He ordered to his advisors to find such a sequence S that would satisfy the constraints he set. Help the advisors of the king and write a program that decides whether such a sequence exists or not.

Input

The input file consists of blocks of lines. Each block except the last corresponds to one set of problems and king's decisions about them. In the first line of the block there are integers n, and m, where $0 < n \leq 100$ is length of the sequence S and $0 < m \leq 100$ is the number of subsequences S_i. Next m lines contain particular decisions coded in the form of quadruples s_i, n_i, o_i, k_i, where o_i represents operator $>$ (coded as `gt`) or operator $<$ (coded as `lt`) respectively. The symbols s_i, n_i, and k_i have the meaning described above. The last block consists of just one line containing `0`.

Output

The output file contains the lines corresponding to the blocks in the input file. A line contains text `successful conspiracy` when such a sequence does not exist. Otherwise it contains text `lamentable kingdom`. There is no line in the output file corresponding to the last "null" block of the input file.

Example

Input file:
```
4 2
1 2 gt 0
2 2 lt 2
```

```
1 2
1 0 gt 0
1 0 lt 0
0
```

Output file:

```
lamentable kingdom
successful conspiracy
```