

B - Dance Dance Revolution

Input: standard input

Output: standard output

The ultimate hyper dance contest™ is taking place every second decade in the Foowitta quadrant. It's so famous that some people are coming from the future (not to mention the ones coming from the past). No wonder they are: the first prize is the astounding Sundie-hippity device. With it, you can convert almost anything to wonderful blue oranges...

But that doesn't concern us here. What does is that a great deal of money can be made out in betting. However, there is no primitive kind of betting such as who will win, but one more intriguing. You will try to guess how many points will a dancer achieve. (As everybody prepares hard for the contest, you may assume that he will do his best.)

Now it is the time to read the rules. Thankfully, they're easy. There is one dancepad with A positions for feet and one middle position called Zero. The other positions are numbered from 1 to A . On each of the positions 1 to A there is just enough place for one foot. The position Zero is large enough for all of the dancer's feet.

The song lasts for some time. During this time the dancer is shown a sequence of required moves. Each move is a set of numbered positions (other than Zero) that have to be stamped on at a given time.

Now, how does moving one's feet work? The only style that is fast enough to match the devilish rhythm of the music works as follows: At any second, if the dancer is standing on the dancepad, he may decide to jump and move some of his feet to different positions. Note that he can also jump immediately after he stamped on the dancepad.

The time the jump takes depends on the number of feet the dancer moves: He will land after $D + \lfloor K/2 \rfloor$ seconds, where D is his delay time (how nimble he is) and K is the number of feet which have to change their position. At the moment he lands, all his feet stamp on their new positions. K may also be equal to zero he can jump on the spot and stamp on the same positions again.

(In the previous paragraph $\lfloor \cdot \rfloor$ is the floor function $\lfloor K/2 \rfloor$ is K divided by 2, rounded down.) If the dancer manages to jump on all the positions of the move at the given time, we say he "caught" the move. To catch the move, the dancer must stamp on all of the specified positions, just standing on them will not activate the sensors in the dancepad. Note that the dancer is not forbidden to stamp on any other positions at any time. E.g. if he is told to stamp on position 1 but in the given second he stamps on both 1 and 2, he still catches the move.

It is up to the contestant to find the best strategy and thus to catch as many moves as possible. If he wants to, he can stand still for some time (although doing so for too long would probably just enrage spectators and put him in a very bad situation like being served as the main dish for the R'tWXwieZy beast), or stand on the Zero position with as many feet as he needs.

At the beginning (in the second 0), the contestant may choose position of his feet on the dancepad. And one more small note: as the contest is intergalactical, there are going to be dancers with many

more than two legs...

Task specification For the dancer, dancepad, and the sequence of moves given, compute the maximum number of the moves he can catch.

Input

There are multiple test cases separated by blank lines. Each test case looks as follows: On the first line there are five integers A, L, D, T, M , where A is number of positions for feet on a dancepad, L is the number of legs (and thus also feet :) of the contestant, D is his delay time, T is the duration of the song and $M = T$ is the number of moves to perform. (The values A, L, D and T are positive, M is non-negative.) Next M lines are formatted in the following way: " $t, k, a_1, a_2 \dots a_k$ ", where t ($0 < t \leq T$) is the second, when the move should be performed, k ($0 < k \leq L$) is the number of prescribed positions and a_1 to a_k ($1 \leq a_i \leq A$) are their numbers.

The input is terminated by a test case where $A=0$. This test case should not be processed.

Output

Print a single line with the string " max_moves move(s) " for each test case.

Sample Input

```
2 1 1 4 3
1 1 1
2 1 1
3 1 1

4 2 2 10 6
3 2 2 4
2 2 3 1
6 1 1
8 1 3
10 2 1 3
7 1 2

0 12 3 29 3
1 1 1
1 1 1
1 1 1
```

Sample Output

```
3 move(s)
4 move(s)
```