

D - Elevator Simulation

Input: standard input

Output: standard output

You are to simulate the movements of an elevator, which works according to the following rules:

1. The elevator stays 3 seconds for customers to get in or get out. If some customers want to get in and some want to get out, the elevator stays 6 seconds.
2. It takes the elevator 2 seconds to go one story upwards or downwards.
3. When there is no request for service, the elevator stays where it is. Namely, it is idle.
4. When the elevator is idle, and a request comes from upwards or downwards, it moves upwards or downwards. If a request comes from the story where it stays, the elevator opens its door to serve these customers.
5. When the elevator goes upwards or downwards, it keeps its direction until there is no request from upwards or downwards. However, it breaks off to let customers get out or pick up customers if necessary.
6. If the elevator is to go upwards or downwards, it does not accept customers intending to go upwards or downwards.
7. When the elevator can act more than one way according to aforementioned rules, it chooses the action with the highest priority. The priorities for actions, from the highest to the lowest, are:
 - let customers get out
 - let customers get in
 - go downstairs
 - go upstairs

Initially (at time 0), the elevator is in the 0th story, and it is idle. Given the requests, you are to output how the elevator moves in a designated interval. Suppose that each request comes at an integer second.

Input

The input begins in its first line with the number of stories of a building, and other two numbers representing an interval. These three numbers are separated by a blank.

Each of the following lines in the input consists of three numbers, N_1 , N_2 , and N_3 , representing a request coming from N_2 th story to N_3 th story at time N_1 .

A line of three 0's separates two test cases. After the last test case, there is a line containing three 0's.

Output

You have to output what the elevator acts at each second in the designated interval. Behind each case should be one blank line.

Sample Input

```
5 9 67
10 2 1
13 2 4
21 4 2
26 4 3
40 1 2
45 0 3
46 0 2
0 0 0
0 0 0
```

Sample Output

```
9: Idle at story 0
10: Going up to 1
11: Going up to 1
12: Going up to 2
13: Going up to 2
14: Let upstairs-customers get in at story 2
15: Let upstairs-customers get in at story 2
16: Let upstairs-customers get in at story 2
17: Going up to 3
18: Going up to 3
19: Going up to 4
20: Going up to 4
21: Let customers get out at story 4
22: Let customers get out at story 4
23: Let customers get out at story 4
24: Let downstairs-customers get in at story 4
25: Let downstairs-customers get in at story 4
26: Let downstairs-customers get in at story 4
27: Going down to 3
28: Going down to 3
29: Let customers get out at story 3
30: Let customers get out at story 3
31: Let customers get out at story 3
32: Going down to 2
33: Going down to 2
34: Let customers get out at story 2
35: Let customers get out at story 2
36: Let customers get out at story 2
37: Let downstairs-customers get in at story 2
38: Let downstairs-customers get in at story 2
```

39: Let downstairs-customers get in at story 2
40: Going down to 1
41: Going down to 1
42: Let customers get out at story 1
43: Let customers get out at story 1
44: Let customers get out at story 1
45: Going down to 0
46: Going down to 0
47: Let upstairs-customers get in at story 0
48: Let upstairs-customers get in at story 0
49: Let upstairs-customers get in at story 0
50: Going up to 1
51: Going up to 1
52: Let upstairs-customers get in at story 1
53: Let upstairs-customers get in at story 1
54: Let upstairs-customers get in at story 1
55: Going up to 2
56: Going up to 2
57: Let customers get out at story 2
58: Let customers get out at story 2
59: Let customers get out at story 2
60: Going up to 3
61: Going up to 3
62: Let customers get out at story 3
63: Let customers get out at story 3
64: Let customers get out at story 3
65: Idle at story 3
66: Idle at story 3
67: Idle at story 3