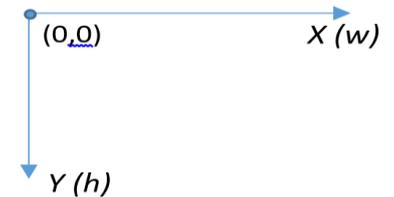


Snake Game

Food is randomly generated on map of $W \times H$ size: snake should eat as much as possible amount of food with limited moves. By the way, it should not hit walls or itself. At the beginning you're able to control snake, which has 3 sections of body (takes 3 cells on the map), and each food increases its length by 1 section.

Description of input/output data:

At the beginning of the game, your program gets world description. There are 2 numbers in first line: N and M – height and width respectively.



Then, your program gets the information about situation on the map via 2 sets of data on each step: data about food location and data about location and direction of snake's head:

$X Y$ – food coordinates;

$X Y D$ – snake's coordinates.

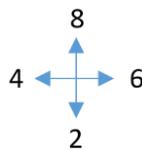
After data transferring, your program have to output one of the numbers (the answer should be due to map, not head):

8 — move up;

6 — move right;

2 — move down;

4 — move left;



Notes:

- Any hit with obstacle (map edge)/own body results snake death
Note: hit is considered to be a situation, when snake does move, after which head coordinates will be the same as obstacle one has.
- To eat food, snake's head should be on the same place, as food has.
Attention, next food will be generated when snake did one move after eating the previous one!
- Each game lasts not longer than N steps; amount of steps counts via formula: $N = W * H * 5$;
- Snake moves all the time.
- In case if you need to continue moving in same direction as head is, than you have to output the respctive move which is responsible for that direction.
- Snake can't move in reverse direction: an attempt to output reverse move to head direction will cause snake death.

N.B.: Do output with “new line” command («endl» in C/C++ and Console.WriteLine() in C#)