

E. Parking Lot

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

A parking lot in the City consists of n parking spaces, standing in a line. The parking spaces are numbered from 1 to n from left to right.

When a car arrives at the lot, the operator determines an empty parking space for it. For the safety's sake the chosen place should be located as far from the already occupied places as possible. That is, the closest occupied parking space must be as far away as possible. If there are several such places, then the operator chooses the place with the minimum index from them. If all parking lot places are empty, then the car gets place number 1.

We consider the distance between the i -th and the j -th parking spaces equal to $4 \cdot |i - j|$ meters.

You are given the parking lot records of arriving and departing cars in the chronological order. For each record of an arriving car print the number of the parking lot that was given to this car.

Input

The first line contains two space-separated integers n and m ($1 \leq n, m \leq 2 \cdot 10^5$) — the number of parking places and the number of records correspondingly.

Next m lines contain the descriptions of the records, one per line. The i -th line contains numbers t_i, id_i ($1 \leq t_i \leq 2$; $1 \leq id_i \leq 10^6$). If t_i equals 1, then the corresponding record says that the car number id_i arrived at the parking lot. If t_i equals 2, then the corresponding record says that the car number id_i departed from the parking lot.

Records about arriving to the parking lot and departing from the parking lot are given chronologically. All events occurred consecutively, no two events occurred simultaneously.

It is guaranteed that all entries are correct:

- each car arrived at the parking lot at most once and departed from the parking lot at most once,
- there is no record of a departing car if it didn't arrive at the parking lot earlier,
- there are no more than n cars on the parking lot at any moment.

You can consider the cars arbitrarily numbered from 1 to 10^6 , all numbers are distinct. Initially all places in the parking lot are empty.

Output

For each entry of an arriving car print the number of its parking space. Print the numbers of the spaces in the order, in which the cars arrive to the parking lot.

Examples

input

```
7 11
1 15
1 123123
1 3
1 5
2 123123
2 15
1 21
2 3
1 6
1 7
1 8
```

output

1
7
4
2
7
4
1
3