Special Hashing

Linear Probing is one of the most used Hashing techiniques. We define here a special hashing which is similar to linear probing.

The following operations are defined.

- Hash: Hash is defined for a number N to be N%k.
- Move forward : Move to the next number (the number connected by the forward link) . Initially, every number's forward link points to the itself. (currentIndex).
- Move backward : Move to the previous number (the number connected by the back link) . Initially, every number's back link points to the itself in the link (currentIndex).
- Insertion operation: Given a number N , find hash(N)=N%k where k is the size of the list. If the list[hash(N)] is empty the element is inserted at position hash(N) in the list and forward link is made to point at (currentIndex+1)%(size of list) and backward link is made to point at (currentIndex-1+sizeof list)%(size of list). If it is filled , we do move_forward/move_backward as specified and then the same process is again repeated.

  Note: Thus list is circular due to modulus property.

- Merge Operation: Let x be a index in the list which is not empty. Calculate xmin by doing a move_backward from index x till the previous index is empty . Similarily calculate xmax by doing a move_forward from index x till the next element is an empty space . Do the same for y to find out ymin and ymax. For a valid merge operation, the index x and y should not be empty and either xmax < ymin or ymax < xmin. Now, when merging x and y, if ymin > xmax, the forward link of xmax is made to point at ymin and the backward link of ymin is made to point to xmax. Same approch is applied in the other case.

  Note: for the merge operation take the min(b,c). The merge is only to be done from x(b)max to x(c)min if the merge was allowed.

Input

The first line of input contains a number representing the number of test cases. Each test case states with a line conataining two integers k(size of list) and C(operations to be applied).C lines follow. Each line contains a,b,c. a is 0 for merge operation followed by index b and c to be merged. a is 1 for insert operation and b is the element to be inserted and c is either 0 or 1(1 in case of left insertion and 0 in case right).

Output

For each operation in each test case,

Case 1: Insertion operation print the position of the hash(b). If the number cannot be inserted print the string "cannot insert element"

Case 2: Merge operation print "merge successful" if the merge was succesful and "cannot merge" if the merge operation failed.

Example

Input:
1
5 6
0 0 2
1 1 1
1 1 0
1 4 0
0 1 4
1 1 1

Output:

cannot merge
1
2
4
merge successful
0

Constraints

Dataset 1:T<25, k ≤=10000,C<=25000 Score: 100
Time limit: 5s Memory Limit: 128MB
Dataset 2:T<8, k ≤=400000,C<=800000 Score: 50
Time limit: 5s Memory Limit: 128MB