Description | Hints | Submissions | Discussions | Notes

Disjoint Union

[illegible]	0.2 sec	256000KB	100
Difficulty	Time Limit	Memory	Score

80/80 XP | 30/30

Description

Write a program that manipulates a disjoint set $S = S_1, S_2, ..., S_k$.

First of all, the program should read an integer *n*, then make a disjoint set where each element consists of *0, 1, ..., n-1* respectively.

Next, the program should read an integer *q* and manipulate the set of *q* queries. There are two kinds of queries for different operations:

1. ***unite(x, y)***: unite sets that contain *x* and *y*, say S_x and S_y, into a new set.
2. ***same(x, y)***: determine whether *x* and *y* are in the same set.

Input Format

The first line contains two space-separated integers *n* and *q*, ($1 \le n \le 10^4$, $1 \le q \le 10^5$).

Then, *q* queries are given in the form ***com x y***, where *com* represents the type of queries. *'0'* denotes ***unite*** and *'1'* denotes ***same*** operation (*x != y*).

Output Format

For each same operation, print 1 if *x* and *y* are in the same set, otherwise 0, in a line.

Constraints

Sample Input 1

Copy

```
5 12
0 1 4
0 2 3
1 1 2
```

C++14 | 00:00:00 | 12 px

```
12          return x;
13      }
14      else
15      {
16          return parent[x] = rep(parent[x]);
17      }
18  }
19  void add(int a, int b)
20  {
21      if(rep(a)==rep(b)){
22          return;
23      }
24      a=rep(a);
25      b=rep(b);
26      if (component_size[a] >= component_size[b])
27      {
28          parent[b] = a;
29          component_size[a] += component_size[b];
30      }
31      else
32      {
33          parent[a] = b;
34          component_size[b] += component_size[a];
35      }
36  }
37  int main()
38  {
39      ios base::sync with stdio(0);
```

Sample Tests | Manual Tests

Test Case 1

Console | Run on Sample