

Problem C

10 Kinds of People


Problem ID: 10kindsofpeople

CPU Time limit: 1 second

Memory limit: 1024 MB

Author: David Sturgill

Source: Baylor Competitive Learning course

License: 

The world is made up of 10 kinds of people, those who understand binary and those who do not. These different kinds of people do not always get along so well. Bob might ask for a 10000 ounce coffee (meaning binary) and Alice might misinterpret his request as being in decimal and give him a 10011100010000 ounce coffee (binary). After Sue explains that this much coffee costs 100 dollars (decimal), Bob might assume he only has to pay 4 dollars (interpreting the price as being in binary). In response to these differences that are difficult to resolve, these two groups have divided the world into two regions, the binary-friendly zones and the decimal-friendly zones. They have even published a map like the following to help people keep up with where the areas are (they have used ones and zeros so nobody would have trouble reading it).

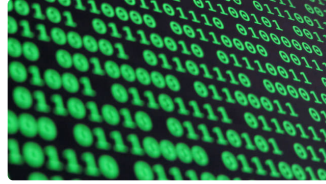


Image by Christiaan Colen

```
1111100000
1111000000
1110000011
0111100111
0011111111
```

Users of binary have to stay in the zones marked with a zero. Users of decimal have to stay in the zones marked with a one. You have to figure out if it is possible for either type of person to get between various locations of interest. People can move north, south, east or west, but cannot move diagonally.

Input

Input starts with a line containing two positive integers, $1 \leq r \leq 1\,000$ and $1 \leq c \leq 1\,000$. The next r input lines give the contents of the map, each line containing exactly c characters (which are all chosen from 0 or 1).

The next line has an integer $0 \leq n \leq 1\,000$. The following n lines each contain one query, given as four integers: r_1, c_1 and r_2, c_2 . These two pairs indicate two locations on the map, and their limits are $1 \leq r_1, r_2 \leq r$ and $1 \leq c_1, c_2 \leq c$.

Output

For each query, output *binary* if a binary user can start from location r_1, c_1 and move to location r_2, c_2 . Output *decimal* if a decimal user can move between the two locations. Otherwise, output *neither*.

Sample Input 1

```
1 4
1100
2
1 1 1 4
1 1 1 1
```

Sample Output 1

```
neither
decimal
```

Sample Input 2

```
10 20
11111111111111111111
11000000000000000101
111111111111111110000
111111111111111110000
11000000000000000111
00011111111111111111
00111111111111111111
10000000000000000111
11111111111111111111
11111111111111111111
3
2 3 8 16
8 1 7 3
1 1 10 20
```

Sample Output 2

```
binary
decimal
neither
```