Problem 1: Counting Haybales

Farmer John has just arranged his N haybales (1≤N≤100,000) at various points along the one-dimensional road running across his farm. To make sure they are spaced out appropriately, please help him answer Q queries (1≤Q≤100,000), each asking for the number of haybales within a specific interval along the road.

INPUT FORMAT (file haybales.in):

The first line contains N and Q.

The next line contains N distinct integers, each in the range 0…1,000,000,000, indicating that there is a haybale at each of those locations.

Each of the next Q lines contains two integers A and B (0≤A≤B≤1,000,000,000) giving a query for the number of haybales between AA and BB, inclusive.

OUTPUT FORMAT (file haybales.out):

You should write QQ lines of output. For each query, output the number of haybales in its respective interval.

SAMPLE INPUT:

4 6
3 2 7 5
2 3
2 4
2 5
2 7
4 6
8 10

SAMPLE OUTPUT:

2
2
3
4
1
0

Problem credits: Nick Wu

Problem 2: Cities and States

To keep his cows intellectually stimulated, Farmer John has placed a large map of the USA on the wall of his barn. Since the cows spend many hours in the barn staring at this map, they start to notice several curious patterns. For example, the cities of Flint, MI and Miami, FL share a very special relationship: the first two letters of "Flint" give the state code ("FL") for Miami, and the first two letters of "Miami" give the state code ("MI") for Flint.

Let us say that two cities are a "special pair" if they satisfy this property and come from different states. The cows are wondering how many special pairs of cities exist. Please help them solve this amusing geographical puzzle!

INPUT FORMAT (file citystate.in):

The first line of input contains NN ($1 \le N \le 200,000$ $1 \le N \le 200,000$), the number of cities on the map. The next NN lines each contain two strings: the name of a city (a string of at least 2 and at most 10 uppercase letters), and its two-letter state code (a string of 2 uppercase letters). Note that the state code may be something like ZQ, which is not an actual USA state. Multiple cities with the same name can exist, but they will be in different states.

OUTPUT FORMAT (file citystate.out):

Please output the number of special pairs of cities.

SAMPLE INPUT:

6
MIAMI FL
DALLAS TX
FLINT MI
CLEMSON SC
BOSTON MA
ORLANDO FL

SAMPLE OUTPUT:

1

Problem credits: Brian Dean

Problem 3: Moocast

Farmer John's NN cows (1≤N≤2001≤N≤200) want to organize an emergency "moo-cast" system for broadcasting important messages among themselves.

Instead of mooing at each-other over long distances, the cows decide to equip themselves with walkie-talkies, one for each cow. These walkie-talkies each have a limited transmission radius -- a walkie-talkie of power PP can only transmit to other cows up to a distance of PP away (note that cow A might be able to transmit to cow B even if cow B cannot transmit back, due to cow A's power being larger than that of cow B). Fortunately, cows can relay messages to one-another along a path consisting of several hops, so it is not necessary for every cow to be able to transmit directly to every other cow.

Due to the asymmetrical nature of the walkie-talkie transmission, broadcasts from some cows may be more effective than from other cows in their ability to reach large numbers of recipients (taking relaying into account). Please help the cows determine the maximum number of cows that can be reached by a broadcast originating from a single cow.

INPUT FORMAT (file moocast.in):

The first line of input contains NN.

The next NN lines each contain the xx and yy coordinates of a single cow (integers in the range 0…25,0000…25,000) followed by pp, the power of the walkie-talkie held by this cow.

OUTPUT FORMAT (file moocast.out):

Write a single line of output containing the maximum number of cows a broadcast from a single cow can reach. The originating cow is included in this number.

SAMPLE INPUT:

4
1 3 5
5 4 3
7 2 1
6 1 1

SAMPLE OUTPUT:

3

In the example above, a broadcast from cow 1 can reach 3 total cows, including cow 1.

Problem credits: Brian Dean