\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SILVER PROBLEMS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Three problems numbered 6 through 8

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Problem 6: Best Spot [Rob Kolstad, 2009]

Bessie, always wishing to optimize her life, has realized that she

really enjoys visiting F (1 <= F <= P) favorite pastures F\_i of the

P (1 <= P <= 500; 1 <= F\_i <= P) total pastures (conveniently

numbered 1..P) that compose Farmer John's holdings.

Bessie knows that she can navigate the C (1 <= C <= 8,000) bidirectional

cowpaths (conveniently numbered 1..C) that connect various pastures

to travel to any pasture on the entire farm. Associated with each

path P\_i is a time T\_i (1 <= T\_i <= 892) to traverse that path (in

either direction) and two path endpoints a\_i and b\_i (1 <= a\_i <=

P; 1 <= b\_i <= P).

Bessie wants to find the number of the best pasture to sleep in so

that when she awakes, the average time to travel to any of her F

favorite pastures is minimized.

By way of example, consider a farm laid out as the map below shows,

where \*'d pasture numbers are favorites. The bracketed numbers are

times to traverse the cowpaths.

1\*--[4]--2--[2]--3

| |

[3] [4]

| |

4--[3]--5--[1]---6---[6]---7--[7]--8\*

| | | |

[3] [2] [1] [3]

| | | |

13\* 9--[3]--10\*--[1]--11\*--[3]--12\*

The following table shows distances for potential 'best place' of

pastures 4, 5, 6, 7, 9, 10, 11, and 12:

\* \* \* \* \* \* Favorites \* \* \* \* \* \*

Potential Pasture Pasture Pasture Pasture Pasture Pasture Average

Best Pasture 1 8 10 11 12 13 Distance

------------ -- -- -- -- -- -- -----------

4 7 16 5 6 9 3 46/6 = 7.67

5 10 13 2 3 6 6 40/6 = 6.67

6 11 12 1 2 5 7 38/6 = 6.33

7 16 7 4 3 6 12 48/6 = 8.00

9 12 14 3 4 7 8 48/6 = 8.00

10 12 11 0 1 4 8 36/6 = 6.00 \*\* BEST

11 13 10 1 0 3 9 36/6 = 6.00

12 16 13 4 3 0 12 48/6 = 8.00

Thus, presuming these choices were the best ones (a program would

have to check all of them somehow), the best place to sleep is

pasture 10.

PROBLEM NAME: bestspot

INPUT FORMAT:

\* Line 1: Three space-separated integers: P, F, and C

\* Lines 2..F+1: Line i+2 contains a single integer: F\_i

\* Lines F+2..C+F+1: Line i+F+1 describes cowpath i with three

space-separated integers: a\_i, b\_i, and T\_i

SAMPLE INPUT (file bestspot.in):

13 6 15

11

13

10

12

8

1

2 4 3

7 11 3

10 11 1

4 13 3

9 10 3

2 3 2

3 5 4

5 9 2

6 7 6

5 6 1

1 2 4

4 5 3

11 12 3

6 10 1

7 8 7

INPUT DETAILS:

As the problem statement

OUTPUT FORMAT:

\* Line 1: A single line with a single integer that is the best pasture

in which to sleep. If more than one pasture is best, choose

the smallest one.

SAMPLE OUTPUT (file bestspot.out):

10

OUTPUT DETAILS:

As the problem statement.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Problem 7: Total Flow [Rob Kolstad, 2008]

Farmer John always wants his cows to have enough water and thus has

made a map of the N (1 <= N <= 700) water pipes on the farm that

connect the well to the barn. He was surprised to find a wild mess

of different size pipes connected in an apparently haphazard way.

He wants to calculate the flow through the pipes.

Two pipes connected in a row allow water flow that is the minimum

of the values of the two pipe's flow values. The example of a pipe

with flow capacity 5 connecting to a pipe of flow capacity 3 can

be reduced logically to a single pipe of flow capacity 3:

+---5---+---3---+ -> +---3---+

Similarly, pipes in parallel let through water that is the sum of

their flow capacities:

+---5---+

---+ +--- -> +---8---+

+---3---+

Finally, a pipe that connects to nothing else can be removed; it

contributes no flow to the final overall capacity:

+---5---+

---+ -> +---3---+

+---3---+--

All the pipes in the many mazes of plumbing can be reduced using

these ideas into a single total flow capacity.

Given a map of the pipes, determine the flow capacity between the

well (A) and the barn (Z).

Consider this example where node names are labeled with letters:

+-----------6-----------+

A+---3---+B +Z

+---3---+---5---+---4---+

C D

Pipe BC and CD can be combined:

+-----------6-----------+

A+---3---+B +Z

+-----3-----+-----4-----+

D

Then BD and DZ can be combined:

+-----------6-----------+

A+---3---+B +Z

+-----------3-----------+

Then two legs of BZ can be combined:

B

A+---3---+---9---+Z

Then AB and BZ can be combined to yield a net capacity of 3:

A+---3---+Z

Write a program to read in a set of pipes described as two endpoints

and then calculate the net flow capacity from 'A' to 'Z'. All

networks in the test data can be reduced using the rules here.

Pipe i connects two different nodes a\_i and b\_i (a\_i in range

'A-Za-z'; b\_i in range 'A-Za-z') and has flow F\_i (1 <= F\_i <=

1,000). Note that lower- and upper-case node names are intended

to be treated as different.

The system will provide extra test case feedback for your first 50

submissions.

PROBLEM NAME: flow

INPUT FORMAT:

\* Line 1: A single integer: N

\* Lines 2..N + 1: Line i+1 describes pipe i with two letters and an

integer, all space-separated: a\_i, b\_i, and F\_i

SAMPLE INPUT (file flow.in):

5

A B 3

B C 3

C D 5

D Z 4

B Z 6

OUTPUT FORMAT:

\* Line 1: A single integer that the maximum flow from the well ('A')

to the barn ('Z')

SAMPLE OUTPUT (file flow.out):

3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Problem 8: Laserphones [Rob Kolstad, 2008]

The cows have a new laser-based system so they can have casual

conversations while out in the pasture which is modeled as a W x H

grid of points (1 <= W <= 100; 1 <= H <= 100).

The system requires a sort of line-of-sight connectivity in order

to sustain communication. The pasture, of course, has rocks and

trees that disrupt the communication but the cows have purchased

diagonal mirrors ('/' and '\' below) that deflect the laser beam

through a 90 degree turn. Below is a map that illustrates the

problem.

H is 8 and W is 7 for this map. The two communicating cows are

notated as 'C's; rocks and other blocking elements are notated as

'\*'s:

7 . . . . . . . 7 . . . . . . .

6 . . . . . . C 6 . . . . . /-C

5 . . . . . . \* 5 . . . . . | \*

4 \* \* \* \* \* . \* 4 \* \* \* \* \* | \*

3 . . . . \* . . 3 . . . . \* | .

2 . . . . \* . . 2 . . . . \* | .

1 . C . . \* . . 1 . C . . \* | .

0 . . . . . . . 0 . \-------/ .

0 1 2 3 4 5 6 0 1 2 3 4 5 6

Determine the minimum number of mirrors M that must be installed

to maintain laser communication between the two cows, a feat which

is always possible in the given test data.

PROBLEM NAME: lphone

INPUT FORMAT:

\* Line 1: Two space separated integers: W and H

\* Lines 2..H+1: The entire pasture.

SAMPLE INPUT (file lphone.in):

7 8

.......

......C

......\*

\*\*\*\*\*.\*

....\*..

....\*..

.C..\*..

.......

OUTPUT FORMAT:

\* Line 1: A single integer: M

SAMPLE OUTPUT (file lphone.out):

3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*