IDI Open
Programming Contest
Warmup Round
April 22nd, 2009

The Problem set

Tips

- Tear the problem set apart and share the problems among you.
- Problems are not ordered by difficulty.
- Try solving the easy problems first. Two problems in this set are tagged with "easy" to help you getting started.
- If you get "incorrect answer" on a problem, you can print your program and debug it on paper while you let someone else work on a different problem on the computer.
- Contact Truls or Nils if you need help.

Rules

- Each team consists of one to three contestants.
- One computer is used per team.
- You may not cooperate with persons not on your team.
- You may print your programs on paper to debug them.
- What you may bring to the contest:
  - Any written material (Books, manuals, handwritten notes, printed notes, etc).
  - Pens, pencils, blank paper, stapler and other useful non-electronic office equipment.
  - NO material in electronic form (CDs, USB pen and so on).
  - NO electronic devices (PDAs and so on).
- The only electronic content you may consult during the content is that specified by the organiser (see the web-page). You may not copy source code from web pages, etc.
- Your programs should read from standard in and write to standard out. Writing to standard error will result in a failed submission. C programs should return 0 from `main()`.
- Your program may use at most 100MB of memory.
- Your programs may not:
  - access the network,
  - read or write files on the system,
  - talk to other processes,
  - fork,
  - or similar stuff.
  - If you try, your program will hang or crash. If it hangs, it will take a couple of minutes before others will be able to run their programs. And please do not crack somebody who uses their spare time trying to give you something valuable.
- Show common sense and good sportsmanship.

Problem A

Addition (Easy)

Your little brother is having a hard time learning to add. As an effort to ensure he never learns this, you decide to write a program that adds two numbers for him.

Input specifications

The first line of input contains a single number T , the number of test cases. The following T lines contain two numbers each, A and B, the numbers you need to add.

Output specifications

For each test case, print a single line containing the number that equals A + B.

Notes and Constraints

- $0 < T \leq 100$
- $-1000 \leq A, B \leq 1000$

Sample input

```
2
1 3
-7 0
```

Output for sample input

```

4
-7
```

Problem B

Virtual Friends

These days, you can do all sorts of things online. For example, you can use various websites to make virtual friends. For some people, growing their social network (their friends, their friends' friends, their friends' friends' friends, and so on), has become an addictive hobby. Just as some people collect stamps, other people collect virtual friends.

Your task is to observe the interactions on such a website and keep track of the size of each person's network.

Assume that every friendship is mutual. If Fred is Barney's friend, then Barney is also Fred's friend.

Input specifications

The first line of input contains a single number T , the number of test cases. Each test case begins with a line containing an integer F, the number of friendships formed. Each of the following F lines contains the names of two people who have just become friends, separated by a space.

Output specifications

Whenever a friendship is formed, print a line containing one integer, the number of people in the social network of the two people who have just become friends.

Notes and Constraints

- $0 < T \leq 100$
- $0 < F \leq 100000$
- A name is a string of 1 to 20 letters (uppercase or lowercase).

Sample input

```
1
3
Fred Barney
Barney Betty
Betty Wilma
```

Output for sample input

```
2
3
4
```

Problem C

Prison Rearrangement

In order to lower the risk of riots and escape attempts, the boards of two nearby prisons of equal prisoner capacity, have decided to rearrange their prisoners among themselves. They want to exchange half of the prisoners of one prison, for half of the prisoners of the other. However, from the archived information of the prisoners' crime history, they know that some pairs of prisoners are dangerous to keep in the same prison, and that is why they are separated today, i.e. for every such pair of prisoners, one prisoners serves time in the first prison, and the other in the second one. The boards agree on the importance of keeping these pairs split between the prisons, which makes their rearrangement task a bit tricky. In fact, they soon find out that sometimes it is impossible to fulfil their wish of swapping half of the prisoners. Whenever this is the case, they have to settle for exchanging as close to one half of the prisoners as possible.

Input specifications

On the first line of the input is a single positive integer T, telling the number of test scenarios to follow. Each scenario begins with a line containing two non-negative integers M, the number of prisoners in each of the two prisons, and R, the number of dangerous pairs among the prisoners. Then follow R lines each containing a pair $x_i y_i$ of integers, which means that prisoner x_i of the first prison must not be placed in the same prison as prisoner y_i of the second prison.

Output specifications

For each test scenario, output one line containing the largest integer K, $K \leq M/2$, such that it is possible to exchange K prisoners of the first prison for K prisoners of the second prison without getting two prisoners of any dangerous pair in the same prison.

Notes and Constraints

- $0 < T \leq 100$
- $0 < M \leq 200$
- $0 < x_i, y_i \leq M$

Sample input

```
3
101 0
3 3
1 2
1 3
1 1
8 12
1 1
1 2
1 3
1 4
2 5
3 5
4 5
5 5
6 6
7 6
8 7
8 8
```

Output for sample input

```
50
0
3
```

Problem D

Scrolling Sign

Electric scrolling signs are often used for advertising. A given sign displays exactly k characters. When the sign is switched on, all of the character positions are initially empty (showing spaces). In each time interval, all of the characters on the sign are shifted to the left by one position, and a new character is added at the right-most position. The character that was in the left-most position moves off the sign.

For certain sequences of words, it is possible to reuse characters from one word to form a subsequent word. For example, on a sign with three character positions, the sign can display the message 'CAT ATE TED' by scrolling in the five characters 'CATED'.

The advertiser has a specific message to show using the sign. The faster the message is displayed, the more people will be able to see the whole message. Therefore, your job is to find a way to display all the words of the message by scrolling in the smallest number of letters. In between showing the words of the message, the sign may display other words that are not considered part of the message. However, the words of the message must be shown in the order in which they are given.

Input specifications

The first line of input contains a single number T , the number of test cases. Each test case starts with a line containing a two integers, K, the number of character positions on the sign, and W, the number of words in the message. The following w lines each contain a word of the message comprising exactly K letters.

Output specifications

For each test case, output a line containing a single integer, the minimum number of letters that must be scrolled into the sign so that it displays all the words of the message.

Notes and Constraints

- $0 < T \leq 100$
- $0 < K, W \leq 100$
- All words consists of only uppercase letters ('A'-'Z').

Sample input

```
2
3 2
CAT
TED
3 3
CAT
ATE
TEA
```

Output for sample input

```
5
5
```

Problem E

Holes

You may have seen a mechanic typewriter - such devices were widespread just 15 years ago, before computers replaced them. It is a very simple thing. You strike a key on the typewriter keyboard, the corresponding type bar rises, and the metallic letter molded into the type bar strikes the paper. The art of typewriter typing, however, is more complicated than the art of computer writing. You should strike keys with some force otherwise the prints will not be dark enough. Also you should not overdo it otherwise the paper will be damaged.

Imagine a typewriter with very sharp letters, which cut the paper instead of printing. It is clear that the digit 0 being typed on the typewritermakes a nice hole in the paper and you recieve a small paper oval as a bonus. The same happens with some other digits: 4, 6, 9 produce one hole, and 8 produces two (touching) holes. The remaining digits just cut the paper without making holes.

The Jury thinks about some exhibition devoted to the upcoming national day of Norway. One of the ideas is to make an art installation, consisting of an empty sheet of paperwith exactly H holes made by typing a non-negative integer number on teh cutting typewriter described above. The number must be as small as possible and should not have any leading zeros. Unluckily we are to busy with preparing IDI Open, so we need your help and ask you to write a computer program to generate the required number.

Input specifications

The first line of input contains a single number T , the number of test cases. Each test case consists of the number H on a line by itself.

Output specifications

For each test case, print a single line containing the number to be printed on the paper.

Notes and Constraints

- $0 < T \leq 100$
- $0 < H \leq 510$

Sample input

```
4
0
1
15
70
```

Output for sample input

```

1
0
48888888
88888888888888888888888888888888888
```

Problem F

Nasty Hacks (Easy)

You are the CEO of Nasty Hacks Inc., a company that creates small pieces of malicious software which teenagers may use to fool their friends. The company has just finished their first product and it is time to sell it. You want to make as much money as possible and consider advertising in order to increase sales. You get an analyst to predict the expected revenue, both with and without advertising. You now want to make a decision as to whether you should advertise or not, given the expected revenues.

Input specifications

The input consists of T cases, and the first line consists of one positive integer giving T. The next T lines each contain 3 integers, R, E and C. The first, R, is the expected revenue if you do not advertise, the second, E, is the expected revenue if you do advertise, and the third, C, is the cost of advertising.

Output specifications

Output one line for each test case: "advertise", "do not advertise" or "does not matter", presenting whether it is most profitable to advertise or not, or whether it does not make any difference.

Notes and Constraints

- $0 < T \leq 100$
- $-10^6 \leq R, E \leq 10^6$
- $0 \leq C \leq 10^6$

Sample input

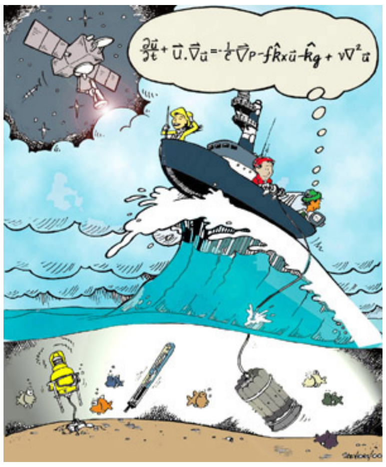
```
3
0 100 70
100 130 30
-100 -70 40
```

Output for sample input

```
advertise
does not matter
do not advertise
```

Problem G

Ocean Currents


$\frac{\partial\vec{u}}{\partial t} + \vec{u}.\vec{\nabla}\vec{u} = -\frac{1}{\varrho}\vec{\nabla}P - f\hat{k}\times\vec{u} - \hat{k}g + \nu\nabla^2\vec{u}$

For a boat on a large body of water, strong currents can be dangerous, but with careful planning, they can be harnessed to help the boat reach its destination. Your job is to help in that planning.

At each location, the current flows in some direction. The captain can choose to either go with the flow of the current, using no energy, or to move one square in any other direction, at the cost of one energy unit. The boat always moves in one of the following eight directions: north, south, east, west, northeast, northwest, southeast, southwest. The boat cannot leave the boundary of the lake. You are to help him devise a strategy to reach the destination with the minimum energy consumption.

For the purposes of this problem, we represent the lake as a rectangular grid.

Input specifications

The first line of input contains a single number T , the number of test cases. Each test case starts with a line containing two integers R and C, the number of rows and columns in the grid. Each of the following R lines contains exactly C characters, each a digit from 0 to 7, inclusive. The character 0 means the current flows north (i.e. up in the grid, in the direction of decreasing row number), 1 means it flows northeast, 2 means east (i.e. in the direction of increasing column number), 3 means southeast, and so on in a clockwise manner, so that 7 means northwest.

The line after the grid contains a single integer N, the number of trips to be made. Each of the following N lines describes a trip using four integers r_s, c_s, r_d, c_d, giving the row and column of the starting point and the destination of the trip. Rows and columns are numbered starting with 1.

Output specifications

For each trip, output a line containing a single integer, the minimum number of energy units needed to get from the starting point to the destination.

Notes and Constraints

- $0 < T \leq 10$
- $1 < R, C \leq 1000$
- $0 < N \leq 50$
- $0 < r_s, r_d \leq R$
- $0 < c_s, c_d \leq C$

Sample input

```
5 5
04125
03355
64734
72377
02062
3
4 2 4 2
4 5 1 4
5 3 3 4
```

Output for sample input

```
0
2
1
```