PLU February 2017 Programming Contest

Advanced Division

I. General Notes

1. Do the problems in any order you like. They do not have to be done in order from 1 to 12.

2. Problems will have either no input or will read input from standard input (stdin, cin, System.in -- the keyboard). All output should be to standard output (the monitor).

3. All input is as specified in the problem. Unless specified by the problem, integer inputs will not have leading zeros.

4. Your program should not print extraneous output. Follow the form exactly as given in the problem.

II. Names of Problems

Number	Name
Problem 1	Moo
Problem 2	Bunnies
Problem 3	Shipping
Problem 4	Animal Conference
Problem 5	Scarecrows
Problem 6	Seed Bags
Problem 7	Cowspeak
Problem 8	Bloom
Problem 9	Field Navigation
Problem 10	Square Pen
Problem 11	Seed Purchasing
Problem 12	Histogram Fencing

1. Moo

You’ve decided to buy a farm and start a new life. To pass some time while you wait for the title of the land to go through, produce the following ascii cow.

Input

There is no input data for this problem.

Output

```
12345678901
2(___)
3(o o)____/
4 @@      \
5  \ ____,/
6  //   //
7 ^^   ^^
```

Output to Screen

```
(___)
(o o)____/
 @@      \
  \ ____,/
  //   //
 ^^   ^^
```

2. Bunnies

You’re going to raise farm animals and you decided to start with bunnies, the easiest of animals. To your surprise they are breeding like rabbits, so much so that you’re unable to count them accurately. However, you know that rabbits’ breeding patterns always follow the Fibonacci sequence. The Fibonacci sequence is defined as follows:

F(0) = 1, F(1) = 1, F(N) = F(N-1)+F(N-2)

Given the number of months the rabbits have been breeding, use the Fibonacci sequence to determine the number of rabbits you should have.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will start with a single integer `x` denoting the number of months that have passed since you bought your initial pair of rabbits. 0<=x<=45

Output

For each test case, output the expected number of rabbits after `x` months.

Sample Input

```
3
0
5
45
```

Sample Output

```
1
8
1836311903
```

3. Shipping

Your animals have begun producing products, and you're honestly a little strapped for cash. Since you have far more animal byproducts than you know what to do with, you've decided to begin shipping them for extra money. Given the information in your shipping ledger, determine how much money you can expect to make.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will start with a single integer `x` denoting how many items follow. The next `x` lines consist of a string, and integer, and a floating point number to two decimal places, representing the name of what was sold, the quantity, and the unit price of each item.

Output

For each test case, output the amount of money you expect to make with all of the goods you sold, rounded to two decimal places.

Sample Input

```
2
3
Eggs 12 0.75
Milk 1 2.00
Feathers 50 .02
2
Cow 1 100.00
Cheese 3 0.54
```

Sample Output

```
$12.00
$101.62
```

4. Animal Conference

Perhaps farming isn't the healthiest of occupations. You've grown suspicious of your animals. Upon examining them, you've noticed that some animals tend to stay oddly close to one another. They're likely conferring against you. In order to know which two animals to keep the closest eye on, create a program that finds the closest two animals to each other.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will start with a single integer `z` denoting how many animals are under examination. The next `z` lines will consist of two non negative integers `x` and `y`, denoting the x and y coordinate of each animal.

Output

For each test case, output the locations of the two closest animals, sorted by x, and then by y in the case of a tie. If there are two such closest pairs of points, choose the smallest sorted pair, i.e., with respect to the first points x, then first point's y, then second point's x, and finally second point's y.

Sample Input

```
2
11
90 60
80 30
12 94
13 43
24 40
26 86
30 60
32 63
70 50
60 96
34 20
5
2 0
1 0
0 0
1 1
0 1
```

Sample Output

```
30 60 32 63
0 0 0 1
```

5. Scarecrows

After having your field picked apart by crows time and time again, you've decided to employ a system of scarecrows. In order to determine how effective your scarecrow network is, and which areas need more protection, create a graph to display which areas of the field are protected. Each of your scarecrows has a unique range in which it works. Any location within P distance from a scarecrow with a range of P will be protected.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will start with a single integer `w`, `h`, and `q`, denoting the width and height of your field, and the number of scarecrows you have. the next `q` lines consist of 3 space separated integers, `r`, `c`, and `p`, representing the horizontal and vertical position of each scarecrow and its range. 0<=r<h, 0<=c<w

Output

For each test case, output h lines of w characters each, where character w of line h is an '*' if point (w,h) is protected, an 'x' if it is a scarecrow, and a '.' Otherwise. Leave a blank line between test cases.

Sample Input

```
2
5 5 1
2 2 2
5 5 2
1 1 1
3 3 1
```

Sample Output

```
..*..
.***.
**x**
.***.
..*..

.*...
*x*..
.*.*.
..*x*
...*.
```

6. Seed Bags

Looking at the space you've allotted for your garden, you begin to wonder how many bags of seeds you'll need to buy in order to fill it. It's a little complicated though, as you plant seeds in a unique way. To plant your seeds, you simply throw the bag in the air, and the seeds land around you in a 3x3 square. Given the layout of your garden, and in which areas you would like crops to grow, determine how many bags of seeds are required. If some seeds land where you don't want them to grow, no worries. You just won't water those. It's also acceptable for an area to be overlapped with multiple bags of seeds.

Input

The first line will contain a single integer n that indicates the number of data sets that follow. Each data set will start with a single integer s (1<=s<=8) denoting the side length of your square garden. The next s lines will represent the layout of your garden. A '.' will represent a spot that does not need to be seeded, while an 'x' will represent an area which you want to plant seeds in.

Output

For each test case, output the minimum number of seed bags you need to plant your garden.

Sample Input

```
2
5
xxxxx
.....
xxxxx
.....
xxxxx
8
.x.xx.x.
x..x..xx
..x.xx.x
.x.x...x
..xx...x
x.x..x.x
..x..x..
.x..x...
```

Sample Output

```
4
7
```

7. Cowspeak

The animals are conspiring again. The cows are mooing very strangely and you're certain of it. Your theory is that the length of each "MM" sound and the length of the "OO" sound can be translated to hexadecimal values and then converted to ascii. Create a program to translate cowspeak to English so you can find out what they're up to.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will consist of one line containing many space separated strings representing the cows' moos.

Output

For each test case, output the decoded cowspeak

Sample Input

```
2
MMMMOOOOOOOOOOOOO MMMMOOOOOOOOOOOOOOO MMMMOOOOOOOOOOOOOOO
MMMMMMOOOOOOOO MMMMMMOOOOOOOOO
```

Sample Output

```
MOO
hi
```

8. Bloom

Your garden is coming along very nicely, and you've grown curious as to when all of your different plants will be blooming. In order to ease your curiosity, you've pulled the seed bags out of the trash and found each plants' growth cycle on the back of their bag. Given the growth cycle of everything you planted, determine how many plants will be blooming on each day.

Plants grow in different stages, and a plant's growth cycle is simply a list of how many days each of these stages take to complete. For example, a plant with a growth cycle of 1 2 4 3 5 would bloom on day 15. Note, the first number in the list is stage 0 (zero).

One added complication is that some plants can be harvested more than once. For example, a plant with a growth cycle of 1 2 4 3 5 that returns to stage 2 after harvest would bloom on days 15, 27, 39, and so on.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will start with a single integer `x` denoting how many plants there are. The next `x` lines will each consist of one plant's growth cycle, denoted by an unknown number of integers. The last integer on each line will represent which stage in growth a plant returns to after blooming, or will be negative one if the plant can only be harvested once. The final line of each test case will consist of one integer, representing the day in question.

Output

For each test case, output the number of plants blooming on the given day.

Sample Input

```
2
3
1 2 3 4 5 0
1 2 3 2
5 5 5 5 5 5 -1
30
3
1 2 3 4 5 0
1 2 3 2
5 5 5 5 5 5 -1
15
```

Sample Output

```
3
2
```

9. Field Navigation

Your field has become quite cluttered, and you just realized you left your best hat out in the middle of the pasture. You have to go retrieve it before it stays out any longer than it already has, but you're also lazy. And there are many obstacles in the way. You have a sickle to cut tall grass, and an axe to chop up any wood in the way, as well as a hammer to destroy rocks you can't get through, but you don't want to have to carry them all. Given the location of your hat and the layout of the field and all the obstacles, determine the minimum number of tools you'll have to bring to reach your prized hat.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will start with a single integer `s` denoting the side length of your square field. The next `s` lines will consist of a map of your field, with '.' denoting open space, 'X' denoting your hat, 'S' denoting your starting position leaving your house, 'R' denoting a rock, 'B' denoting a branch, and 'G' denoting tall grass. You can only move up, down, left, and right. No diagonals.

Output

For each test case, output the minimum number of tools needed to retrieve your hat.

Sample Input

```
3
5
S....
GGGG.
.....
.RRRR
....X
7
BRBRGGG
RGGGGRG
BGBBRRG
RGGGGRG
BBRBGBG
GGRBGRG
SGGGGRX
4
GGGX
RRRG
BBRG
SBRG
```

Sample Output

```
0
1
3
```

10. Square Pen

You want to create a huge square pen for your animals in one of your fields. That way they can enjoy the sunlight instead of staying pent up in their dusty barn all year. Your field is still messy though. You don’t want to clean it, you just want to build the biggest possible square pen without any junk in it. Given the layout of your field, determine the area of the largest square pen you can make.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will start with a single integer `s` denoting the side length of your square field. The next `s` lines will consist of the layout of your field, with “`#`” denoting an obstacle, and ‘`.`’ denoting a free space.

Output

For each test case, output the area of the largest square pen you can make without it having any obstacles inside of it.

Sample Input

```
2
5
...##
##...
.....
#....
#..##
6
...##.
##..##
..###.
#.....
#..##.
..##..
```

Sample Output

```
9
4
```

11. Seed Purchasing

The new season has begun, and it is time to purchase all new seeds. But there are so many choices! And so many variables. Every bag of seeds costs a certain amount, and each bag also produces a certain number of crops, which each sells for a certain amount of money. It's awful to keep track of, and you simply can't do all the math in your head. Create a program to determine the maximum amount of money you can earn on crops this year given the amount you have available to spend on seeds.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will start with a single integer `x` denoting how many different seed choices there are, followed by a floating point number `y` denoting how much you have to spend. The next `x` lines will consist of a floating point number `c`, an integer `m`, and a floating point number `p`, representing the cost of each bag, the number of crops it produces, and the selling price of each crop the bag produces. The store has an unlimited supply of each variety of seeds.

Output

For each test case, output the maximum amount of money you can make selling crops this season. Do not include the change from your seed purchases in the answer.

Sample Input

```
1
4 15.00
2.00 7 1.00
1.00 4 1.00
11.00 10 5.00
5.00 1 5.00
```

Sample Output

```
$51.00
```

12. Histogram Fencing

Your next-door neighbor has asked to borrow some fencing to put around all of his land. Being the kind neighbor that you are, you've decided to lend him your fencing. But being the stingy farmer that you are, you've also decided to give him exactly enough, and no more. An interesting quality of your neighbor's farm is that it's shaped like a histogram. Given the width and height of each column of the histogram, determine the perimeter of your neighbor's land.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will start with a single integer `x` denoting the number of 'bars' in your neighbor's histogram shaped plot of land. The next two lines consists of `x` space separated integers each, representing the widths and heights respectively of each bar of land

Output

For each test case, output the amount of fencing needed to surround the perimeter of your neighbors histogram shaped plot of land.

Sample Input

```
2
7
1 1 1 1 1 1 1
1 3 6 3 3 3 2
3
1 2 3
1 2 3
```

Sample Output

```
26
18
```