Fourteenth Quinnipiac University Programming Competition for High School Students

Quinnipiac University

April 27, 2019

Instructions

1. Program submissions are done using the Programming Contest Control System (PC^2). See the separate sheet for instructions on submitting your programs.
2. You may assume the input is formatted as shown in the problem, and your program must work on such input. You may assume there are no tabs in the input or trailing spaces.
3. Your output must match the specified output *exactly*. This includes not having any trailing spaces.
4. If your program crashes, produces incorrect output, or times out, no information will be given as to the input that caused the problem.

1 Rotational Equivalence

Two numbers are rotationally equivalent if they have the same number of digits and you can get from one number to the other by rotating the digits moving the last digit to the front. Your task is to take two numbers and determine if they are rotationally equivalent.
Here are some example numbers to illustrate:

- 12345 and 34512 are rotationally equivalent
- 12 and 21 are rotationally equivalent
- 123 and 123 are rotationally equivalent
- 12304 and 30412 are rotationally equivalent
- 12345 and 2345 are not rotationally equivalent (different lengths)
- 12340 and 1234 are not rotationally equivalent (different lengths)
- 12345 and 54321 are not rotationally equivalent
- 12345 and 23415 are not rotationally equivalent
- 12304 and 40123 are not rotationally equivalent

Input

The first line contains the number N of cases with $0 < N \leq 100$. This is then followed by N cases. Each case consists of two numbers A and B on a single line separated by space with $0 < A, B \leq 10^{10}$. *No number will start with a 0.*

Output

The output consists of one line for each case, which is either YES or NO depending on if the two are rotationally equivalent or not.

Input

```
8
12345 34512
12 21
123 123
12304 30412
12345 2345
12345 54321
12345 23415
12304 40123
```

Output

```
YES
YES
YES
YES
NO
NO
NO
NO
```

2 Fibonacci Sum

The Fibonacci numbers are a sequence of numbers $1, 1, 2, 3, 5, \ldots$ formed by adding the previous two values to form the next value. That is, $F_1 = 1, F_2 = 1, F_i = F_{i-1} + F_{i-2}$. Suppose you are given a number V, determine a sequence of k Fibonacci numbers that add up to V. When choosing among multiple sequences, use the one with the fewest numbers (smallest value of k). (There will always be at least one sequence.)
For example, 7 can be formed with $2 + 5$, $1 + 1 + 5$, or $1 + 1 + 2 + 3$. So, we would say 7 is formed using $2 + 5$ (since that uses just two numbers). If the number is a Fibonacci number, then the sequence is just that number!
The numbers don't repeat. So, $10 = 2 + 8$ not $5 + 5$.

Input

The first line contains the number N of cases with $0 < N \leq 100$. This is then followed by N cases. Each case is just a single number M with $0 < V \leq 1000000$.

Output

The output consists of one line for each case, which is just the shortest sequence of Fibonacci numbers that add up to V (listed in increasing order).

Input

```
5
7
10
5
456
100
123456
```

Output

```
2 5
2 8
5
3 21 55 377
3 8 89
```

3 Elevator Ride

Boomer recently made a recent trip to the Big Apple to visit his pal Jay, who was staying in an extremely tall (and deep) luxury building (with an immeasurable number of floors above and below ground!). Jay had decided to play a game with Boomer by not telling him the actual floor he was staying on but instead gave him some instructions consisting of a set of floors to go up or down on the elevator starting on floor 0 (ground floor). The ending location would be the floor he needed to get off on.
For example, $+2, -1, +4$ would mean to go up 2 floors, then down a floor, and then up four floors ending up on the 5th floor. Boomer was planning on riding the elevator up and down until he ended up on the final floor until his buddy Sebastian suggested that he write a program to solve it. Help Boomer meet up with Jay.

Input

The first line contains the number N of cases with $0 < N \leq 100$. This is then followed by N cases. Each case is given on a single line with the first number K being the number of steps followed by K numbers $S_1, S_2, \ldots, S_k$ representing the steps to take. In all cases, $0 < K \leq 100$ and $-100 \leq S_i \leq +100$.

Output

The output consists of one line for each case, where the line is the ending floor for that case.

Input

```
4
3 2 -1 4
5 -2 -3 0 4 -1
4 100 200 300 -2
8 1 -2 3 -4 5 -6 7 -8
```

Output

```
5
-2
598
-4
```

4 Elevator Ride Revisited

After getting off the elevator at the correct floor according to the instructions, he finds a note from Jay: "Ha! You didn't think it would be that easy." Jay's note continues to explain that Boomer has to continue to repeat the instructions over and over until he reaches a floor that he visits twice.
For example, suppose the instructions were $500, -100, 500, 300, -900$. Following these instructions, Boomer's trip would be:

1. Start on floor 0
2. Go up 500 floors to floor 500
3. Go down 100 floors to floor 400
4. Go up 500 floors to floor 900
5. Go up 300 floors to floor 1200
6. Go down 900 floors to floor 300
7. At this point, the instructions start over again from the current floor.
8. Go up 500 floors to floor 800
9. Go down 100 floors to floor 700
10. Go up 500 floors to floor 1200
11. At this point, Boomer has been to floor 1200 before so this is his stop.

Help Boomer revise his program to handle larger values *and* recognize that repeat floor so he can identify the correct floor.

Input

The first line contains the number N of cases with $0 < N \leq 100$. This is then followed by N cases. Each case is given on a single line with the first number K being the number of steps followed by K numbers $S_1, S_2, \ldots, S_k$ representing the steps to take. In all cases, $0 < K \leq 100$ and $-10^7 \leq S_i \leq 10^7$ and at no point will the current floor fall out of the range $(-10^8, 10^8)$.

Output

The output consists of one line for each case, where the line is the ending floor for that case.

Input

```
6
5 500 -100 500 300 -900
5 -500 100 -500 -300 900
2 1 -2
4 25 -14 27 -40
4 650 550 -300 -897
4 25 -15 8 -15
```

Output

```
1200
-1200
0
11
1200
25
```

5 Text Editor

Boomer received an "encoded" note from his pal Jay. After having met in the Big Apple, they decided on the following encoding scheme. Suppose you have a string consisting of characters <, >, and A-Z. The string hides an actual message which can be found by doing the following:

1. Go across each character in the input string.
   (a) If the character is a < then delete the first character in the string up to this point (unless there are none left)
   (b) If the character is a > then delete the last character in the string up to this current position (unless there are none left)
2. The resulting string at the end is the final message.

For example, suppose the input string A<>BCD<EFG><. We would transform the string as follows:

1. First, we see an A
2. Then, we encounter < so we delete the first character A leaving an empty string
3. Encountering a > we attempt to delete the last character, but there is no character left (so we move on)
4. The next three symbols create the string BCD
5. We then encounter a < so we delete the first character B, leaving CD
6. The next three symbols EFG leaves us with the string CDEFG
7. The next symbol > transforms the string to CDEF
8. Finally, the symbol < transforms the string to DEF (our final string)

Help Boomer write a program to decode the messages he receives from Jay.

Input

The first line contains the number N of cases with $0 < N \leq 100$. This is then followed by N cases. Each case is a single string of characters of length 1 to 100 containing only the symbols A-Z, <, and >.

Output

The output consists of one line for each case, where the ouput per case is the transformed string.

Input

```
5
A<>BCD<EFG><
<<<ABC<D<EFGH>>
ABCDE>>><><>F<>
AHAH<>
TAKING<F>OOD<<<JUST>>><ON>BUS>>
```

Output

```
DEF
CDEF

HA
GOODJOB
```

6 Text Editor Revisited

Sebastian decided to join in the party and sent a message. But he didn't follow one key rule. He made an extremely long message. Help Boomer revise the previous program so that it handles input strings as large as 1,000,000 characters long. Everything else remains the same.

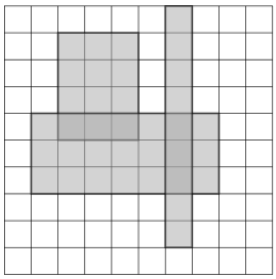
Perhaps your program already works, but if not efficiency is probably going to be important here!

7 Arena Seating

After a recent hockey game, the packed crowd left in such a rush that every single seat in the arena was littered with trash. Boomer and his buddies including Jay and Sebastian volunteered to help and clean up the area. Each of them chose a region to clean up and naturally they did not discuss beforehand so there was lots of overlap. Your task is to determine how many seats at the end were not cleaned up.

Fortunately for us the seating in the arena is described as a standard rectangular grid and Boomer and his buddies all chose rectangular regions to clean.

For example, suppose that the arena was 10 rows by 10 columns which we number as $(0, 0)$ in the upper left and $(9, 9)$ in the lower right. Further suppose that Boomer chose to clean the seats from row 1, column 2 to row 4, column 4; Sebastian cleaned from row 4, column 1 to row 6, column 7; and Jay cleaned from row 0, column 6 to row 8, column 6. Looking at the figure below, you can see that in the end they cleaned 36 seats (6 seats were overlapped), leaving 64 seats untouched.



Input

The first line contains the number N of cases with $0 < N \leq 100$. This is then followed by N cases. Each case consists of a line of numbers (separated by spaces). The first two numbers represent the dimensions of the arena R by C. The next number is the number of workers W. This is then followed by $4W$ numbers. The first four of these numbers represent the starting row S_R, starting column S_C, ending row E_R and ending column E_C of the first worker, followed by the 4 numbers for the second worker, and so on.

Here are some assumptions that you can make:

- The dimensions of the arena will be between 1 and 1000: $0 < R, C \leq 1000$.
- The number of workers is no more than 10 (and at least 1): $0 < W \leq 10$.
- All regions selected by the workers will fall inside the arena dimensions. (They aren't cleaning the ice rink).
- All regions selected by the workers will have the start row and column values be no larger than the end row and column. That is, $0 \leq S_R \leq E_R < R$ and $0 \leq S_C \leq E_C < R$.

Output

The output consists of one line for each case, which is simply the number of uncleaned seats.

Input

```
4
10 10 3 1 2 4 4 4 1 6 7 0 6 8 6
20 20 4 0 4 19 7 0 12 19 15 4 0 7 19 12 0 15 19
15 15 4 7 0 7 14 3 3 11 11 0 7 14 7 13 13 13 13
10 10 4 1 1 1 1 2 2 2 2 3 3 3 3 2 2 9 9
```

Output

```
64
144
131
35
```

8 Stadium Seating

Boomer and his pals were so thrilled with their stellar job cleaning the hockey arena that they decided to up their game and volunteer at the Big Apple's new mega-stadium. The stadium layout was the same. Unfortunately, the stadium's size was a bit bigger than they expected. The problem is identical, the only assumption that has changed is:

- The dimensions of the stadium will be between 1 and 1000000: $0 < R, C \leq 1000000$.

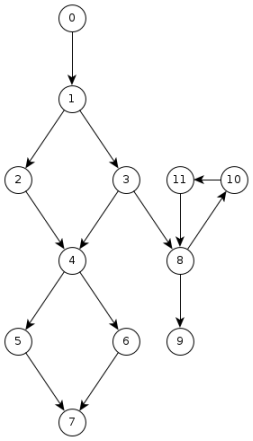
Will your previous solution still work or will you have to improve your approach to handle this larger problem more efficiently?

9 Route Counter

Boomer is visiting his favorite ski resort in Vermont. He has a map of all the possible trails he can take which he has converted to a nice directed graph. The graph has a collection of vertices that represent starting and ending points or intersection points on the slope and edges that represent directed paths from one vertex to another (downward skiing only). Boomer has visited this resort plenty of times and has done many different routes along these fabled slopes. But today he wanted to ride every possible route on the slopes from one starting point s to one ending point t. But before he does this task, he needs to know how many actual routes there would be. That is where you shall come in. Help Boomer decide if it is worth pursuing this monumental task.
For example, in the graph below, there are 4 possible routes from vertex 1 to vertex 7.

$$(1, 2, 4, 5, 7), (1, 2, 4, 6, 7), (1, 3, 4, 5, 7), (1, 3, 4, 6, 7)$$

Note, since all of the slopes are downward. This map will not have any loops. So, the input graph will not contain a cycle like $(8, 10, 11)$ instead the edge $(8, 10)$ (or one of the other edges in that cycle) would not be present.



Input

The first line contains the number N of cases with $0 < N \leq 100$. This is then followed by N cases. The input for each case consists of a directed graph G followed by two vertices s and t (on the same line). The first line of input for each graph consists of two integers V and E representing the number of vertices and edges respectively. This is then followed by E lines of the form `A B` where A and B are two integers representing an edge from vertex A to vertex B. After those E lines, there is one final line `S T` indicating the start and end vertices s and t. The vertices are numbered from 0 to $V - 1$.

Output

The output consists of one line for each case, which is just the number of routes for that case (which will not be larger than 10^7 but could be 0).
Here are some assumptions that you can make:

- The number of vertices is more than 0 and no more than 30.
- The graph provided is valid and in the valid format (no incorrect input format to check).
- There are no duplicate edges (two edges from a to b) or simple loops (an edge from a to a).
- The number of possible routes is no more than 10^7.
- The graph does not have any loops (simple or otherwise).

Input

```
2
12 13
0 1
1 2
1 3
2 4
3 4
4 5
4 6
5 7
6 7
3 8
8 10
10 11
8 9
1 7
6 15
1 0
2 1
2 0
3 2
3 1
3 0
4 3
4 2
4 1
4 0
5 4
5 3
5 2
5 1
5 0
5 0
```

Output

```
4
16
```

10 Route Counter Revisited

Boomer was so happy with his recent excursion that he convinced Jay and Sebastian to fly out with him to the Rockies to try out his routine on his favorite ski resort in Colorado. This time the resort is a bit bigger. The maps are different of course but the only two assumptions that have changed are:

1. The number of vertices is more than 0 and no more than 500.
2. The number of possible routes is no more than 2^{62}.

Will your previous solution still work, or will you have to improve your approach to handle this larger problem?