1 The Whooziewhat Factory I

Problem Description

The Whooziewhat Factory is the largest producer of Whooziewhats in the country. As the Vice Chief Assistant to the Production Strategist Board Chair, you are in charge of determining the *production schedule*, or the number of whooziewhats produced each month. You must produce enough whooziewhats to satisfy the *demand schedule*, or the number of whooziewhats you need to deliver on the last day of each month. Of course, whooziewhat source materials have volatile prices, so the cost to produce a whooziewhat follows the *cost schedule*, which shows the cost to produce each whooziewhat. Given the *demand schedule* and the *cost schedule*, determine the minimum cost to deliver all the whooziewhats on time.

Input Description

The first line of input contains a single integer T, representing the number of test cases.

Each test case begins with a single integer n, the number of months in the schedules. The next line contains n integers, d_i, $1 \leq i \leq n$, representing the demand in month i. The next line contains n integers, c_i, $1 \leq i \leq n$, representing the cost to produce each whooziewhat in month i.

All data will be non-negative integers.

Output Description

For each test case, output "$P", the cost to produce all of the whooziewhats.

Sample Input

```
1
4
2 5 4 3
10 2 5 6
```

Sample Output

```
$42
```

Sample Explanation

The best way to produce whooziewhats is to produce the 12 whooziewhats needed for months 2, 3, and 4 all at once in month 2 when the price is only $2 per whooziewhat. You must produce both whooziewhats needed for month 1 in month 1, at a cost of $10 per whooziewhat. Total cost:

$$(2 \text{ whooziewhats}) * \$12 + (12 \text{ whooziewhats}) * \$2 = \$42$$

.

2 The Wikipedia Game

Problem Description

The Wikipedia Game is a game where a player begins on a randomly selected Wikipedia article and attempts to navigate to an unrelated article via clicking on links embedded in each article. The goals of the game is to traverse from the beginning page to the ending page in the fewest number of clicks. Given a map of Wikipedia, can you determine the optimal solution to The Wikipedia Game?

Input Description

The first line of input contains a single integer, n, representing the number of Wikipedia articles in the game.

For each of the n pages, the first line contains an integer m, representing the number of links on the page, followed by the article title. The next m lines contain the target pages.

Following next line after this data is an integer k, representing the number of data sets. Each data set is two lines, with the source and target page on separate lines, respectively.

Output Description

For each data set, output the list of articles in the shortest path from the source to the target. In the event of a tie, output the lexigraphically least ordering. Separate each data set with a blank line.

Sample Data

Please see the separate files for sample data.

3 The Whooziewhat Factory II

Problem Description

A horrible discovery has been made at the Whooziewhat Factory; turns out, Whooziewhats are perishable. To make matters worse, refrigeration costs are seasonal. From now on, you will need to take into account the *holding schedule*, which shows the cost of holding one whooziewhat from the current month to the following month.

Input Description

The first line of input contains a single integer T, representing the number of test cases.

Each test case begins with a single integer n, the number of months in the schedules.

- The next line contains n integers, d_i, $1 \leq i \leq n$, representing the demand in month i.
- The next line contains n integers, c_i, $1 \leq i \leq n$, representing the cost to produce each whooziewhat in month i.
- The next line contains $n - 1$ integers, h_i, $1 \leq i < n$, representing the holding cost from month i to month $i + 1$.

All data will be non-negative integers.

Output Description

For each test case, output "\$P", the cost to produce all of the whooziewhats.

Sample Input

```
1
4
2 5 4 3
10 2 5 6
3 1 7
```

Sample Output

```
$60
```

Sample Explanation

You still must produce the 2 whooziewhats for month 1 in month 1 at a cost of \$10 per whooziewhat. It is still cheapest to produce month 3 whooziewhats in month 2, but now the cost for those 4 whooziewhats is \$2 to produce plus \$1 to hold. Since the holding cost from month 3 to month 4 is so high, it is now cheaper to produce month 4 whooziewhats in month 4. Total cost:

$$(2 \text{ whooziewhats}) * \$10 + (5 \text{ whooziewhats}) * \$2 + (4 \text{ whooziewhats}) * (\$2 + \$1) + (3 \text{ whooziewhats}) * \$6 = \$60$$

.

4 Fake Friends

Problem Description

A friend is *fake* if they are a friend of an enemy. A friend of an enemy is an enemy. Note that an enemy of an enemy is not necessarily a friend; nor is a friend of a friend necessarily a friend. Also, relationships are symmetric; that is, if Alice is Bob's friend, then Bob is also Alice's friend, and identically for enemies.

A sign of a socially healthy, well-adjusted person is that they cut all of their fake friends out of their lives immediately. Given a list of friends and enemies, determine who your fake friends are so you can block them on all social media immediately.

Note: A person can be both a friend and an enemy. In fact, this is equivalent to being a fake friend.

Input Description

The first line of the input contains a single integer n, representing the number of test cases.

Each test case begins with a single integer k, the number of relationships. The next k lines contain 2 space-separated names, followed by the word "friends" or "enemies". (In this problem, names will not contain spaces.) At least one of the names is "You", representing you.

Output Description

For each test case, output a list of fake friends, separated by new lines. Separate test cases by a new line.

Sample Input

```
2
6
You Alice friends
You Bob friends
You Cathy enemies
Cathy David friends
Cathy Bob enemies
Alice David friends
7
You A friends
You B enemies
You C friends
You D friends
A B friends
A C friends
D E friends
```

Sample Output

```
David

A
C
```

5 The Whooziewhat Factory III

Problem Description

The Whooziewhat industry has experienced rapid growth and production has not kept up with demand. It is now possible to deliver Whooziewhats behind schedule; however, you must pay a fee for each whooziewhat per month that you do not deliver on time. The fee follows the *late fee schedule*, which shows the cost per whooziewhat of delaying delivery by one month.

Input Description

The first line of input contains a single integer T, representing the number of test cases.

Each test case begins with a single integer n, the number of months in the schedules.

- The next line contains n integers, d_i, $1 \leq i \leq n$, representing the demand in month i.
- The next line contains n integers, c_i, $1 \leq i \leq n$, representing the cost to produce each whooziewhat in month i.
- The next line contains $n-1$ integers, h_i, $1 \leq i < n$, representing the holding cost from month i to month $i+1$.
- The next line contains $n-1$ integers, h_i, $1 \leq i < n$, representing the late fee for delaying a whooziewhat delivery from month i to month $i+1$.

All data will be non-negative integers.

Output Description

For each test case, output “\$P”, the cost to produce all of the whooziewhats.

Sample Input

```
1
4
2 5 4 3
10 2 5 6
3 1 7
1 8 9
```

Sample Output

```
$60
```

Sample Explanation

Now, it is cheaper to delay production of month 1 whooziewhats to month 2, paying a \$1 late fee plus a \$5 production cost per whooziewhat. Total cost:

(2 whooziewhats)$*$(\$1+\$5)+(5 whooziewhats)$*$\$2+(4 whooziewhats)$*$(\$2+\$1)+(3 whooziewhats)$*$\$6 = \$52

.

6 Time Tables

Problem Description

You need to take the bus to make it where you're going. Unfortunately, there will be many bus transfers along the way, and the transfers don't always line up super well. Given a bus time table, determine if you can make it to your destination on time.

Each bus will have a time table showing how long from departing the starting station it takes to arrive at all other stations. All buses leave their starting stations at time $t = 0$, and will arrive back at their starting stations again at the end of their route (all routes are cycles). The bus then immediately departs the station again, and continues its loop. You are on a bus when it leaves at time $t = 0$.

Input Description

The first line of input is a single integer T, the number of test cases.

Each test case begins with a single integer n, the number of buses.

There are n bus schedules that follow. Each schedule begins with an integer k, the number of stops made by the bus, and a starting bus station (a string with no spaces). The next k lines contain the name of a bus stop (a string with no spaces) followed by a number of minutes it takes to arrive at that stop from when the bus left the previous stop.

The final line in each test case is two bus stops names, the starting and ending destination.

Output Description

For each test case, output the least number of minutes it takes to go from your starting bus stop to your ending bus stop.

Sample Input

```
1
2
3 A
B 2
A 2
3 B
C 2
D 3
B 2
```

Sample Output

```
12
```

Sample Explanation

There are 2 buses between 4 stops. The first bus travels between stops A and B, two minutes each. The second bus travels between B to C (2 minutes), then C to D (3 minutes), and then returs to B (2 more minutes). You take the first bus and arrive at stop B at $t = 2$. You then wait until the bus arrives again at $t = 7$. You then ride for two stops, a total of 5 more minutes. You therefore arrive at stop D at time $t = 12$.

7 The Whooziewhat Factory IV

Problem Description

A breakthrough in the Whooziewhat Production Laboratory has yielded a new way to deliver Whooziewhats: Time travel! From now on, you will have the ability to send a whooziewhat through time, from one month to another. Of course, variations in the local space-time field means that the cost to send a whooziewhat through time depends both on the source month and the destination month, so you will now have a two-dimensional *time travel schedule.*

Input Description

The first line of input contains a single integer T, representing the number of test cases.

Each test case begins with a single integer n, the number of months in the schedules.

- The next line contains n integers, d_i, $1 \leq i \leq n$, representing the demand in month i.
- The next line contains n integers, c_i, $1 \leq i \leq n$, representing the cost to produce each whooziewhat in month i.
- The next line contains $n - 1$ integers, h_i, $1 \leq i < n$, representing the holding cost from month i to month $i + 1$.
- The next line contains $n-1$ integers, h_i, $1 \leq i < n$, representing the late fee for delaying a whooziewhat delivery from month i to month $i + 1$.
- The next n lines contain n integers each. The jth integer in the ith row is t_{ij}, representing the cost to send a single whooziewhat through time, from month i to month j. For every i, t_{ii} will be zero (you will never time-travel to the present month).

All data will be non-negative integers.

Output Description

For each test case, output "\$P", the cost to produce all of the whooziewhats.

Sample Input

```
1
4
2 5 4 3
10 2 5 6
3 1 7
1 8 9
0 10 10 10
2 0 1 10
10 10 0 10
10 10 10 0
```

Sample Output

```
$48
```

Sample Explanation

The only difference here is that we want to produce month 4's Whooziewhats is month 2, time travel them to month 3, and then hold for one month.

Total cost:

$$(2\text{ whooziewhats})*(\$1+\$5)+(5\text{ whooziewhats})*\$2+(4\text{ whooziewhats})*(\$2+\$1)+(3\text{ whooziewhats})*(\$2+\$1+\$1) = \$48$$

.