2014 FHSPS Playoff

March 15, 2014

Timber Creek High School

Problem/Filename	Problem Name
a	Dallas Buyers' Club
b	Building Olaf
c	The Great Gatsby
d	Hunger Games
e	Seating Arrangements
f	Movie Trip
g	Twelve Years a Slave
h	The Wolf of Wall Street

Problem A: Dallas Buyers' Club

Filename: a

You are an FBI Agent investigating Ron Woodruff and the Dallas Buyers' Club. You have recovered some data of customers, but much of the data has been lost, due to some really bad handwriting. Your goal is to find evidence of the members of the club. You have a list of people you believe to be in the club and you want to see for each handwritten name if it may match one of the people on your list. For each of the handwritten names, you can only make out some letters and can't tell the number of letters in between the ones you can read. Thus, for each of these names, you simply know a list of letters that are a subsequence of the full name. For example, the list 'l', 'j', 'n', and 'a' is a subsequence found in "alejandra", but 'j', 'e', 'r', and 'a' is NOT a subsequence found in "alejandra", because no 'e' appears after the first 'j' in "alejandra."

The Problem

Given a list of names, and several subsequences of letters, for each subsequence, determine if it belongs to any of the names on the given list.

The Input

The first line of the input file will contain a single positive integer, n $(n \leq 10)$, representing the test cases to process. Each of the test cases will follow. The first line of each test case will contain a single positive integer, k $(k \leq 1000)$, representing the number of names on the list. The following k lines of each test case will contain the complete names on the list. Each name will be comprised of lower case letters only and be in between 1 and 20 letters long, inclusive. This will be followed by a line with a positive integer, m $(m \leq 1000)$, representing the number of handwritten names to match. The following m lines of each test case will contain strings of lowercase letters only representing subsequences of letters found in each handwritten name. Each of these strings will also contain in between 1 and 20 lowercase letters, inclusive.

The Output

For each test case, print out a header line with the following format:

```
List #i:
```

where i is the list number starting at 1. For each name to check on the list, output a single line with the input subsequence, followed by a colon, followed by a space, followed by either `"FOUND"` or `"NOT FOUND"`, depending on whether any name on the original list is consistent with that subsequence of letters.

Follow the output for each case with a blank line.

Sample Input

```
2
3
john
amanda
evelyn
4
jhn
mdaa
e
lyn
5
asha
amelia
jack
manny
marty
3
jacky
jack
m
```

Sample Output

```
List #1:
jhn: FOUND
mdaa: NOT FOUND
e: FOUND
lyn: FOUND

List #2:
jacky: NOT FOUND
jack: FOUND
m: FOUND
```

Explanation of Sample Data

In the first case, "john" is consistent with the first subsequence, none of the names is consistent with the second subsequence, and "evelyn" is consistent with both the third and fourth subsequences. The "closest" match to "mdaa" is "amanda", but after the first 'd' in "amanda", there is only one 'a' left, not two.

In the second case, "jacky" is longer than the name on the list, "jack", so there is no match. The name exactly matches, so the subsequence "jack" matches the name "jack". Finally, "amelia", "manny" and "marty" match the subsequence "m".

Problem B: Building Olaf

Filename: b

Princess Elsa and her sister Anna used to make snowmen. In time, they realized that for particular purposes, different ratios for the radii of the three spheres that formed their snowmen were desirable. Since they know of your superior mathematics skills, they have asked you to help them determine the radii of each of the spheres of the snowman they intend to build.

Please use the following value for PI in your calculation: 3.141592653590.

The Problem

Given the total volume of snow (in cm^3) that is available to Elsa and Anna, as well as the integer ratios of the radii of the three spheres to make the desired snowman, determine the radius of the smallest sphere.

The Input

The first line of the input file will contain a single positive integer, n $(n \le 100)$, representing the number snowmen to build. Each of the test cases will follow, one per line. Each test case will have four positive integers, v $(v \le 1000000)$, s, m $(s \le m)$, l $(m \le l \le 100)$, separated by spaces, representing the volume in cubic centimeters of the snowman, and the integer ratios of the small, medium and large sphere radii to build that snowman.

The Output

For each test case, print out the radius, rounded to two decimal places, in centimeters, of the smallest sphere of the snowman.

Sample Input

```
2
10000 1 2 3
65412 1 3 6
```

Sample Output

```
4.05
4.00
```

Problem C: The Great Gatsby

Filename: c

You've been hired by the eccentric Jay Gatsby to help him in his affair with Daisy Buchanan. In particular, Jay needs to make sure that whenever he sees Daisy, her husband Tom isn't around. Using your various private eye tools, you've learned to predict perfectly both Daisy and Tom's whereabouts. Jay would like for you to determine the maximum number of minutes he can see Daisy on a given day. You may assume that Jay is so wily that he gets to Daisy as soon as Tom leaves and he can leave right when Tom returns to the same location as Daisy.

The Problem

Given both Daisy and Tom's Schedules for a full day (starting at midnight and ending the next midnight), determine the number of minutes Jay can see Daisy that day.

The Input

The first line of the input file will contain a single positive integer, n $(n \leq 1000)$, representing the number of days for which you've been asked to help Jay out with his calculation. The first line of each test case will contain a single positive integer, d $(d \leq 20)$, representing the number of locations Daisy visits during the day. This is followed by d lines, each of which contains a pair of positive integers, d_l $(d_l \leq 100)$, and d_m $(d_m \leq 1440)$, representing the location number and number of minutes, respectively, that Daisy spent at that location. The next line of each test case will contain a single positive integer t $(t \leq 20)$, representing the number of locations Tom visits during the day. This is followed by t lines, each of which contains a pair of positive integers, t_l $(t_l \leq 100)$, and t_m $(t_m \leq 1440)$, representing the location number and number of minutes, respectively, that Tom spent at that location. The lists of the times for both individuals are in chronological order, starting at midnight. The sum of the number of minutes at all locations throughout the day will be exactly 1440 for both individuals and consecutive locations on each list will be distinct. The same location number may be seen multiples times on each list, however, since people can visit the same place more than once in a day.

The Output

For each test case, print out the maximum number of minutes Jay can see Daisy for that day on a line by itself.

Sample Input

```
2
3
1 500
2 440
1 500
2
1 800
2 640
2
1 501
2 939
2
2 500
1 940
```

Sample Output

```
800
1439
```

Explanation of Sample Cases

In the first test case, Tom and Daisy are in location number 1 for the first 500 minutes. For the next 300 minutes, Daisy is in location 2 while Tom is in location 1, so Gatsby can see Daisy for these 300 minutes. For the following 140 minutes, both Tom and Daisy are in location 2. For the final 500 minutes, Daisy is in location 1 and Tom remains in location 2, so Gatsby can see Daisy for these last 500 minutes, for a total of 800 minutes.

In the second case, Tom and Daisy are together for one minute of the whole day, the 501st minute of the day in location 1. Thus, Gatsby can see Daisy the whole day except for that one minute, which is 1439 minutes total.

Problem D: Hunger Games

Filename: d

Due to popular demand, the Hunger Games continue. The new arena is in the shape of a triangle. You must help train your district's competitor. You've determined that the safest place for a competitor in the arena is the single point that is equidistant from all three sides of the triangular arena. Your job will be to automate the process of calculating this point, given various different arena designs.

The Problem

Given the Cartesian coordinates of three points that define a triangle with positive area, determine the point inside of the triangle that is equidistant from all three sides of the triangle.

The Input

The first line of the input file will contain a single positive integer, n $(n \leq 100)$, representing the number triangles to process. Each of the triangles will follow, one per line. Each test case will consist of 6 space separated integers, x_1, y_1, x_2, y_2, x_3 and y_3, respectively denoting the Cartesian coordinates of the three points that form the triangle of the arena, which is guaranteed to have positive area. All coordinates will have an absolute value less than or equal to 10000.

The Output

For each arena, print out the x and y coordinate, rounded to two decimal places, separated by a space on a line by itself of the safest point for that arena.

Sample Input

```
1
0 0 10 0 0 10
```

Sample Output

```
2.93 2.93
```

Problem E: Seating Arrangements

Filename: e

You and your significant other enjoy going to the movies with your other couple friends. However, you've noticed that if any couple sits adjacent to each other, the corresponding public display of affection is not necessarily savory for the rest of the group. Thus, you want to ensure that whenever you go to the movies with your group of couple friends, the seating arrangements are such that no two couples are sitting next to each other but everyone is in one contiguous segment of seats, on a single row. To prove to the others that this is a good idea, you've decided to count the number of possible sets of seats the couples can arrange themselves adhering to this restriction. (You hope that by showing them that there are plenty of valid arrangements with the new restriction, the other couples will go along with your rule.)

For the purposes of this problem, the couples are indistinguishable from each other, as are the two members of each couple. Assume the seats are numbered 1, 2, ..., 2*n*, from left to right, where *n* is in the number of couples attending the movie. For example, if there are two couples, only one possible set of seats works: (1, 3), (2, 4). (Thus, we're NOT counting an arrangement where the couple in seats 1 and 3 swaps with the couple in seats 2 and 4, AND we're not counting differently if the couple themselves swap seats.) If there are three couples, the following five sets of seats works:

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

The Problem
Given the number of couples attending a movie, determine the number of sets of seating arrangements on a contiguous segment of a single row that are valid such that no two couples are sitting next to one another.

The Input
The first line of the input file will contain a single positive integer, n $(n \leq 9)$, representing the number of input cases to consider. Each of the test cases will follow, one per line. Each test case will have a single positive integer, c $(c \leq 9)$, on a line by itself.

The Output
For each test case, print out the number of valid seating arrangements for that case.

Sample Input	Sample Output
3	0
1	1
2	5
3	

Problem F: Movie Trip

Filename: f

All year you've been trying to get an edge on the other competitors in FHSPS. After several contests you realize that all of the questions have a movie theme. Familiarity always helps with problem solving so you decide that you are going to watch more movies, in the hopes that the questions in the Playoff Contest are about some of the movies you watch! But, watching movies is lonely to do alone, so you want to bring some friends and family. Though watching these movies may give you an edge in the contest, it may be costly. You'd like to figure out how much it'll cost to go to the movies!

The Problem

Given the price of an adult ticket, a child ticket, the number of adult tickets to purchase and the number of child tickets to purchase, determine the total cost of the group going to the movies. Assume that both prices include tax.

The Input

The first line of the input contains a single positive integer, n $(n \leq 100)$, representing the number of test cases in the input file. Each test case, representing a single trip to the movies, follows, one per line. On each of these lines will be four values, a $(a \leq 20.00)$, c $(c \leq 12.00)$, t_a $(t_a \leq 100)$, and t_c $(t_c \leq 100)$, representing the price of an adult ticket in dollars, the price of a child ticket in dollars, the number of adult tickets bought and the number of child tickets bought, respectively, separated by spaces. The first two values on each line will always be positive values given to two decimal places and the last two values will both be non-negative integers.

The Output

For each test case, print out the price of the corresponding trip to the movies, to exactly two decimal places.

Sample Input

```
2
12.99 7.99 3 2
15.00 10.00 4 0
```

Sample Output

```
54.95
60.00
```

Problem G: Twelve Years a Slave

Filename: g

You and your classmates are currently taking AP American History. In order to diversify your experience in the class, your teacher has required for you and all of your classmates to watch the 2014 Oscar winning film, "12 Years a Slave." You have a very busy life and would like to fit the movie in to your plans without wasting too much time, as is the case with many of your classmates. Your plan is to swing by a movie theater in between an already scheduled trip. You already know that the movie takes exactly 134 minutes. We assume due to your alacrity that it takes no time to park, get to the movie, and leave the theater. We also assume that fortuitously, the movie begins exactly whenever you get to any theater, no matter when that might be. Write a program that will calculate the number of different theaters you can watch the movie at in the middle of a given planned trip and still arrive at your destination on time. You are considered to be on time if you take the exact number of minutes allowed for the whole trip or fewer minutes, you will only watch the movie once on your trip, even if your path takes you through multiple movie theaters, and you may visit as many intermediate locations as necessary to minimize your trip time.

The Problem

Given a list of all locations in your city, some of which will be movie theaters, as well as several trips you are taking and the total amount of time you have to take each trip, determine the number of different theaters you could theoretically stop by to watch 12 Years a Slave for each specific trip and still make it to your destination on time.

The Input

The first line of the input file will contain a single positive integer, n $(n \leq 300)$, representing the number of locations in the city. Assume that the locations are numbered 0 through $n-1$. The following n lines will each contain n space separated non-negative integers. The j^{th} value (with these values starting at 0) on the i^{th} line (with line numbers starting at 0) will represent the number of minutes it will take to travel from location i to location j. Assume that the movie theaters are located at the odd indexed locations and that all even indexed locations are not movie theaters. Note that it may take a different number of minutes to travel from city i to city j than to travel from city j to city i and it will always take 0 minutes to travel from city i to city i. The next line of input contains a single positive integer, p, representing the number of travel paths to evaluate. Each of the travel paths will follow, one per line. Each travel path will be described with three space separated integers, s, e and t. The first integer, s, represents the starting location of travel and is guaranteed to be an even integer less than n. The second integer, e, represents the ending location of travel and is guaranteed to be an even integer less than n. The last integer, t $(0 \leq t \leq 1440)$, represents the number of minutes you are given to make the trip from s to e. You are guaranteed that there will exist some path between s and e that takes t minutes or less.

The Output

For each travel path, output on a line by itself, the number of different movie theaters at which you could watch, "12 Years a Slave" and still make it to your destination on time.

Sample Input

```
5
0 10 20 30 3
15 0 5 25 35
40 30 0 15 20
35 35 10 0 10
40 3 20 5 0
3
0 2 145
4 0 300
2 0 140
```

Sample Output

```
1
2
0
```

Explanation of Sample Cases

For the first trip, we can go from location 0 to location 4 to location 1 in 6 minutes to arrive at a movie theater, since each leg of the trip is 3 minutes long. From there, we watch the movie in 134 minutes. Then we can travel directly from location 1 to location 2 in 5 minutes, taking a total of 145 minutes. There is no way to go from location 0 to location 3 to location 2 AND watch the movie in 145 minutes.

There is ample time to swing by either location 1 or location 3 to catch the movie and get to location 0 afterwards in the second trip.

In the third trip, we can't watch the movie at either location 1 or location 3. The minimum time spent to swing through location 1 is 172 minutes (location 2 to location 4 to location 1 to location 0) and the minimum time spent to swing thorugh location 3 is 177 minutes (location 2 to location 3 to location 4 to location 1 to location 0).

Problem H: The Wolf of Wall Street

Filename: wolf

On the side, Jordan Belfort uses his stock market skills to make a few wagers on sporting events. He has a set of lucky numbers and has predetermined the different amounts he wants to wager, but hasn't determined which games he should wager each amount. Since he's not so good at math, he's hired you to determine how much money to wager on each game, so as to maximize his expected profit.

The Problem

Given the amount of money wagered on each of a set of games, as well as the probability of winning each wager, determine the maximum amount of the expected profit.

The Input

The first line of the input file will contain a single positive integer, n $(n \leq 100)$, representing the number of weeks to calculate wagers for. Each of the test cases will follow. The first line of each test case will contain a single positive integer, g $(g \leq 10)$, representing the number of games Jordan would like to make wagers for the week. The second line of each test case includes g space-separated positive integers, representing the dollar amounts Jordan would like to wager of the g games. All of these values will be no more than 1000000. The third line of each test case includes g space-separated real numbers in between 0 and 1, inclusive, representing the probabilities of winning each of the g wagers. Each probability will be expressed to 3 or fewer decimal places.

The Output

For each test case, output the maximum expected profit, in dollars, for arranging the wager amounts appropriately. (Note: This value may be negative.)

Sample Input

```
2
3
10 20 30
.5 .5 .6
3
100 200 300
.5 .55 .45
```

Sample Output

```
6.00
20.00
```