

11690 Money Matters

Our sad tale begins with a tight clique of friends. Together they went on a trip to the picturesque country of Molvania. During their stay, various events which are too horrible to mention occurred. The net result was that the last evening of the trip ended with a momentous exchange of “I never want to see you again!”s. A quick calculation tells you it may have been said almost 50 million times!

Back home in Scandinavia, our group of ex-friends realize that they haven’t split the costs incurred during the trip evenly. Some people may be out several thousand crowns. Settling the debts turns out to be a bit more problematic than it ought to be, as many in the group no longer wish to speak to one another, and even less to give each other money.

Naturally, you want to help out, so you ask each person to tell you how much money she owes or is owed, and whom she is still friends with. Given this information, you’re sure you can figure out if it’s possible for everyone to get even, and with money only being given between persons who are still friends.

Input

The first line of the input file contains an integer N ($N \leq 20$) which denotes the total number of test cases. The description of each test case is given below:

The first line contains two integers, n ($2 \leq n \leq 10000$), and m ($0 \leq m \leq 50000$), the number of friends and the number of remaining friendships. Then n lines follow, each containing an integer o ($-10000 \leq o \leq 10000$) indicating how much each person owes (or is owed if $o < 0$). The sum of these values is zero. After this comes m lines giving the remaining friendships, each line containing two integers x, y ($0 \leq x < y \leq n - 1$) indicating that persons x and y are still friends.

Output

For each test case your output should consist of a single line saying ‘POSSIBLE’ or ‘IMPOSSIBLE’.

Sample Input

```
2
5 3
100
-75
-25
-42
42
0 1
1 2
3 4
4 2
15
20
-10
-25
0 2
1 3
```

Sample Output

POSSIBLE
IMPOSSIBLE

PROBLEMA B – Time limit: 3 sec

Factorial Factors

Arquivo: *factorial*.*[clcpp|java]*

The factorial of a number N , written $N!$, is the product of all integers between 1 and N , inclusively. For example, $5! = 120$.

Every integer greater than 1 can be written as the product of 1 or more prime numbers, some of which may repeat. For example, $120 = 2 * 2 * 2 * 3 * 5$.

For this problem, we are interested in the prime factorization of the factorial of a number. You will need to determine the number of total and distinct prime factors. For the example above, there are 5 total prime factors (2, 2, 2, 3, 5) and 3 distinct prime factors (2, 3, 5).

Input

The first line of input will contain the number of test cases, C ($1 \leq C \leq 50$). Each test case will consist of a single line containing an integer N ($2 \leq N \leq 100,000$).

Output

Each test case will result in a single line of output D T where D is the number of distinct prime factors of $N!$ and T is the total number of prime factors of $N!$.

Sample Input	Sample Output
2	3 5
5	3 7
6	

PROBLEMA C – Time limit: 3 sec

Palindrome Numbers

Arquivo: *palindrome.[c|cpp|java]*

A word is considered a palindrome if it is the same when read left to right or right to left. For this problem, a positive integer is considered a palindrome if its base 10 (decimal) representation is the same from left to right as right to left. For each number, determine the smallest palindrome by value that is greater than or equal to that number. Unlike with Nascar car numbers, leading zeroes are insignificant and do not appear in any number in the input or output of this problem.

Input

The first line of input will contain the number of test cases, T ($1 \leq T \leq 50$). Each of the following T lines contains a positive integer N that is no more than 80 digits in length.

Output

The output of each test case will be a single line containing the smallest palindrome that is greater than or equal to the input number.

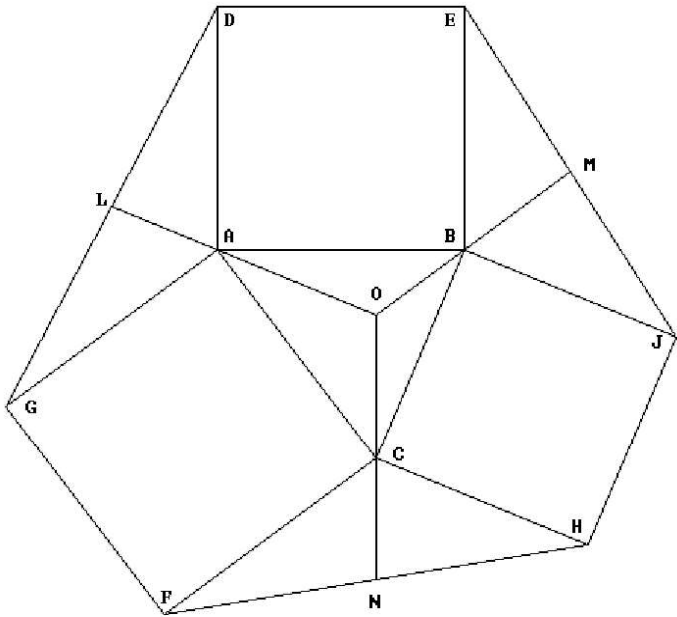
Sample Input	Sample Output
2	44
42	323
321	

Exocenter of a Triangle

Arquivo: *exocenter*.*[c|cpp|java]*

Given a triangle ABC, the Extriangles of ABC are constructed as follows:

On each side of ABC, construct a square (ABDE, BCHJ and ACFG in the figure below). Connect adjacent square corners to form the three Extriangles (AGD, BEJ and CFH in the figure). The Exomedians of ABC are the medians of the Extriangles, which pass through vertices of the original triangle, extended into the original triangle (LAO, MBO and NCO in the figure. As the figure indicates, the three Exomedians intersect at a common point called the Exocenter (point O in the figure). This problem is to write a program to compute the Exocenter of triangles.



Entrada

The first line of the input consists of a positive integer n , which is the number of datasets that follow. Each dataset consists of 3 lines; each line contains two floating point values which represent the (two -dimensional) coordinate of one vertex of a triangle. So, there are total of $(n*3) + 1$ lines of input. Note: All input triangles will be strongly non-degenerate in that no vertex will be within one unit of the line through the other two vertices.

Saída

For each dataset you must print out the coordinates of the Exocenter of the input triangle correct to four decimal places.

Entrada	Saída Correspondente
2 0.0 0.0 9.0 12.0 14.0 0.0 3.0 4.0 13.0 19.0 2.0 -10.0	9.0000 3.7500 -48.0040 23.3600

PROBLEMA E – Time Limit: 3 seg

Controle Financeiro

Arquivo: controle.[c|cpp|java]

Darlene graduou este ano e por seu excelente desempenho conseguiu um bom emprego. Ela agora ganha dinheiro, mas parece que nunca ter o suficiente. Darlene decidiu que ela precisa tomar controle de suas finanças, e resolver seus problemas financeiros. O primeiro passo é perceber o que ocorre com seu dinheiro. Ela tem o extrato de sua conta bancária, e deseja verificar quanto dinheiro possui. Ajude Darlene com a escrita de um programa para pegar seu fechamento bancário, dos últimos 12 meses, e calcular seu saldo médio.

Input

A entrada contém vários casos de teste, onde a primeira linha indica a quantidade de testes. Cada teste possui 12 linhas, e cada uma destas contém o fechamento bancário da conta para um mês. Cada número será positivo e serão incluídos os centavos, com duas casas decimais. O sinal de cifrão (\$) não será incluído. A entrada deve ser lida da entrada padrão.

Output

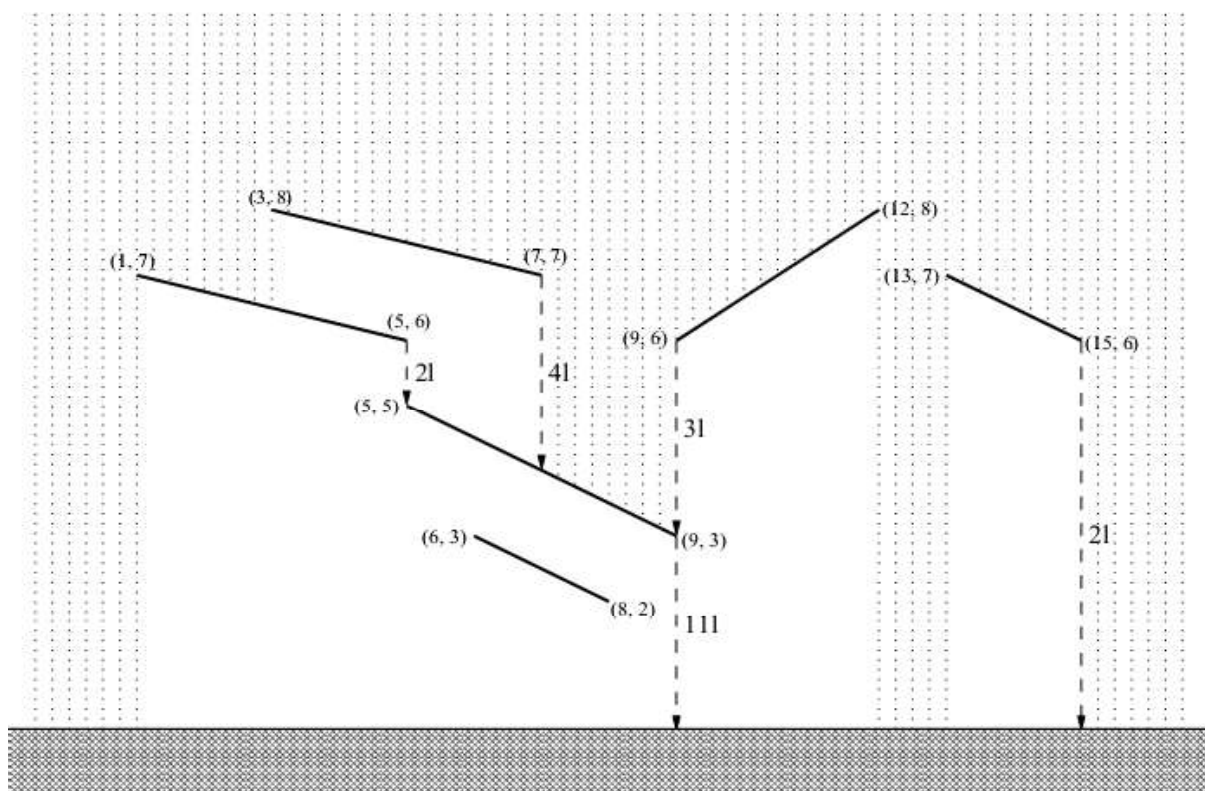
A saída será um número simples, com a média do fechamento bancário para os 12 meses. Este número será precedido pelo sinal de cifrão, terá o centavo mais próximo, e seguido por final de linha. Não haverá outros espaços ou caracteres na saída. A saída deve ser escrita na saída padrão.

Exemplo de Entrada	Saída Correspondente
1 100.00 489.12 12454.12 1234.10 823.05 109.20 5.27 1542.25 839.18 83.99 1295.01 1.75	\$1581.42

Chuva de Novembro

Arquivo: *chuva.[clcpp|java]*

Construções modernas podem ter telhados bastante complicados. Se tomarmos uma seção vertical de um destes telhados, teremos vários pequenos segmentos de rampas, como vistos na figura abaixo. Quando chove, as gotas caem sempre retas sobre os telhados. Alguns segmentos estão completamente expostos à chuva, enquanto alguns deles podem estar parcialmente ou completamente protegidos por outros. Toda a água que cai em um segmento escorre para baixo até a beirada do segmento e então cai sobre o chão ou algum outro segmento que se encontre no caminho. Em particular, caso um fluxo de água esteja caindo na extremidade de um segmento, consideramos que ele está sendo coletado por aquele segmento.



Para que seja possível projetar um bom sistema de calhas, é desejável computar a quantidade de água que está escorrendo por cada um dos segmentos do telhado. Para estar preparado para uma chuva intensa de novembro, você deve contar um litro de água da chuva caindo por metro do plano horizontal a cada segundo.

Escreva um programa que leia a descrição do telhado, calcule a quantidade de água escorrendo em um segundo de cada segmento do telhado e escreva os resultados.

Entrada

A primeira linha da entrada contém um inteiro n ($1 \leq n \leq 40000$) que representa a quantidade de segmentos do telhado. Cada uma das n linhas seguintes descreve um segmento de telhado e contém quatro inteiros x_1 , y_1 , x_2 e y_2 ($0 \leq x_1, y_1, x_2, y_2 \leq 1000000$, $x_1 < x_2$, $y_1 \neq y_2$) separados por espaços. Inteiros x_1 e y_1 são a posição horizontal e a altura da extremidade esquerda do segmento de telhado. Inteiros x_2 e y_2 são a posição horizontal e altura da extremidade

direita do mesmo segmento. Os segmentos não possuem pontos em comum e não há segmentos horizontais. Você pode, também, assumir que existam no máximo 25 segmentos colocados sobre qualquer ponto no nível do chão.

Saída

A saída consiste em n linhas. A i -ésima linha deve conter a quantidade de água (em litros) escoando do i -ésimo segmento de telhado em um segundo.

Exemplo de Entrada	Saída Correspondente
6	2
13 7 15 6	4
3 8 7 7	2
1 7 5 6	11
5 5 9 3	0
6 3 8 2	3
9 6 12 8	

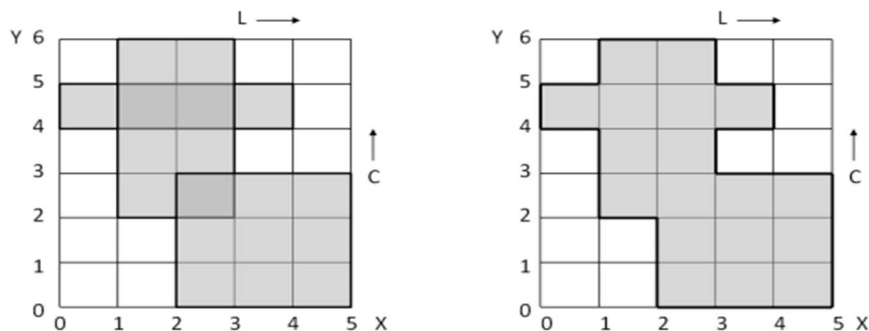
Fazenda

Arquivo: *fazenda.[c|cpp|java]*

Compadre Edmilson teve uma agradável surpresa: recebeu um telefonema de sua irmã dizendo que eles eram os únicos herdeiros das terras dos seus tios. Logo em seguida recebeu um fax com a descrição dos terrenos. Todos eram retangulares, e pela largura e comprimento de cada um concluiu que herdou uma grande área de terra em Piranguinho! Não tardou e Compadre Edmilson já estava lá para tomar posse dos terrenos.

Chegando lá, uma surpresa nada agradável... os terrenos eram sobrepostos!! Ninguém conseguiu explicar como isso aconteceu, mas boa parte das terras dos seus tios não pertenciam a apenas um, mas a dois ou mais! Isso nunca foi problema para eles, que cuidavam de tudo como uma grande família, sem se preocupar exatamente que pedaço era de quem. Tampouco é problema para Compadre Edmilson e sua irmã, que agora são os únicos donos. Mas é frustrante descobrir que o terreno não era tudo o que pensava...

A figura abaixo mostra um exemplo com 3 terrenos. Os terrenos são descritos com quatro números inteiros X, Y, L, C, indicando que o canto sudoeste do terreno (na figura, o canto inferior esquerdo) está na coordenada (X, Y) e ele tem largura L e comprimento C (todos os valores dados em Km). A descrição dos 3 terrenos mostrados é: Terreno A: 1 2 2 4; Terreno B: 2 0 3 3; Terreno C: 0 4 4 1. A figura da esquerda mostra os terrenos, e a da direita o terreno total resultado da sobreposição.

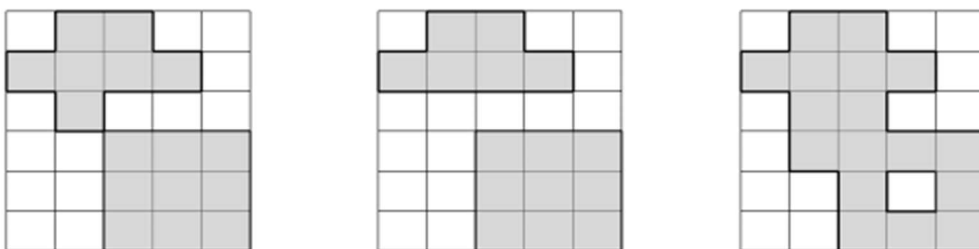


Quando Compadre Edmilson recebeu o fax, achava que esses terrenos tinham $2 \times 4 + 3 \times 3 + 4 \times 1 = 21 \text{Km}^2$ no total. Agora sabe que são 18Km^2 ...

Ele gostaria de saber quantos Km^2 os terrenos realmente contêm. E com as muitas sobreposições, precisa da sua ajuda. Além disso precisa cercar o terreno, então gostaria de saber também quantos Km de cerca são necessários. No exemplo acima a cerca tem 24Km de comprimento.

Observações

A junção dos terrenos nem sempre forma uma região contígua e sem “buracos”. Casos como os mostrados na figura abaixo podem acontecer. Então nem sempre será possível cercar o terreno herdado com uma única cerca contígua. Pode ser necessário usar mais de uma.



Entrada

Há vários casos de teste.

A primeira linha de um caso de teste contém um número inteiro N que é o número de terrenos herdados ($N \leq 20$). As N linhas seguintes contém cada uma a descrição de um terreno no formato X Y L C conforme descrito no enunciado ($0 \leq X$; $Y \leq 1000$, e $1 \leq L$; $C \leq 500$).

A entrada termina quando $N = 0$

Saída

Para cada caso de teste da entrada seu programa deve produzir uma linha na saída contendo dois valores A e P, que são a área (em Km^2) e o perímetro (em Km) total da união dos terrenos descritos no caso de teste.

Exemplo de Entrada	Saída Correspondente
3 1 2 2 4 2 0 3 3 0 4 4 1	18 24 23 36 12 24
5 1 2 1 1 1 2 5 1 2 5 4 2 4 0 1 6 1 0 2 4 4 0 0 1 4 0 3 4 1 3 0 1 4 1 0 2 1 0	

10608 Friends

There is a town with N citizens. It is known that some pairs of people are friends. According to the famous saying that “The friends of my friends are my friends, too” it follows that if A and B are friends and B and C are friends then A and C are friends, too.

Your task is to count how many people there are in the largest group of friends.

Input

Input consists of several datasets. The first line of the input consists of a line with the number of test cases to follow.

The first line of each dataset contains the numbers N and M , where N is the number of town's citizens ($1 \leq N \leq 30000$) and M is the number of pairs of people ($0 \leq M \leq 500000$), which are known to be friends. Each of the following M lines consists of two integers A and B ($1 \leq A \leq N$, $1 \leq B \leq N$, $A \neq B$) which describe that A and B are friends. There could be repetitions among the given pairs.

Output

The output for each test case should contain (on a line by itself) one number denoting how many people there are in the largest group of friends on a line by itself.

Sample Input

```
2
3 2
1 2
2 3
10 12
1 2
3 1
3 4
5 4
3 5
4 6
5 2
2 1
7 1
1 2
9 10
8 9
```

Sample Output

```
3
7
```