Do **NOT** read the problems before contest starts.

Contest URL: http://www.link.cs.cmu.edu/contest/current/.

2014 Carnegie Mellon University Programming Contest

and

ACM ICPC Team Selection Qualification

Round 2

6:35pm-10:05pm, Sept 17, 2014

PROBLEMS

Hints:

1. Problems are not sorted from easiest to hardest. It is important to solve the easiest problems first to minimize your penalty score. Usually, you can guess the difficulty of the problems by looking at the scoreboard.
2. Be careful about the efficiency of input/output. If the input size is larger than 10^5 bytes, try to use " `scanf / printf` " in C++ and " `BufferedReader / BufferedWriter` " in Java.
3. Do not access the internet, except for language support and our online judge for submissions.

Problem A
Easy task?

Last year there were a lot of complaints concerning the set of problems. Most contestants considered our problems to be too hard to solve. One reason for this is that the team members responsible for the problems are not able to evaluate properly whether a particular problem is easy or hard to solve. (We have created until now so many problems, that all seems quite easy.) Because we want our future contests to be better we would like to be able to evaluate the hardness of our problems after the contest, using the history of submissions.

There are a few statistics that we can use for evaluating the hardness of a particular problem: the number of accepted solutions of the problem, the average number of submissions of the problem and the average time consumed to solve it (as "General rules" of the contest state "the time consumed for a solved problem is the time elapsed from the beginning of the contest to the submittal of the accepted run"). For the latter two statistics we consider only the teams which solved this particular problem.

Needless to say we ask you to write a program that computes aforementioned statistics for all problems.

Task

Write a program that:

- reads a history of submissions during an ACM contest,
- computes for each problem the number of accepted solutions of the problem, the average number of submissions and the average time consumed to solve it,
- writes the result.

Input

The first line of the input contains one integer n ($1 \leq n \leq 2000$) being the number of submissions during the contest. Each of the next n lines describes one submission and contains a submission time (measured in seconds from the beginning of the contest), a team identifier, a problem identifier and a result of evaluating the submission, separated by single spaces. The submission time is a positive integer not greater than 18000. The team identifier is a non-empty string consisting of at most five small letters or digits. The problem identifier is a capital letter `A`, `B`, ..., or `I`. The result is a capital letter `A` (the submission is accepted) or `R` (the submission is rejected).

Submissions are given in nondecreasing order according to submission times and there are 60 teams competing.

Please note that if a problem is accepted all further submissions of this problem by the same team are possible but they should not be taken to the statistics.

Output

The output consists of nine lines. The first line corresponds to problem A, the second line to problem B, and so on. Each line should contain the problem identifier, the number of accepted solutions of the problem, the average number of submissions done by teams that solved that problem and the average time consumed to solve it, separated by single spaces. The latter two statistics should be printed only if there was at least one accepted solution of the given problem and should be rounded to two fractional digits (in particular 1.235 should be rounded to 1.24).

Example

For the input:

```
12
10 wawu1 B R
100 chau1 A A
2000 uwr2 B A
2010 wawu1 A R
2020 wawu1 A A
2020 wawu1 B A
4000 wawu2 C R
6000 chau1 A R
7000 chau1 A A
8000 pp1 A A
8000 zil2 B R
9000 zil2 B A
```

the correct answer is:

```
A 3 1.33 3373.33
B 3 1.67 4340.00
C 0
D 0
E 0
F 0
G 0
H 0
I 0
```

Problem B

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 specifications

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 specifications

Your output should consist of a single line saying "POSSIBLE" or "IMPOSSIBLE".

Sample input 1	Sample output 1

5 3
100
-75
-25
-42
42
0 1
1 2
3 4
	POSSIBLE

Sample input 2	Sample output 2

```
4 2
15
20
-10
-25
0 2
1 3
```
	`IMPOSSIBLE`

Problem C

Allergy Test

A test for allergy is conducted over the course of several days, and consists of exposing you to different substances (so called allergens). The goal is to decide exactly which of the allergens you are allergic to. Each allergen has a live duration D measured in whole days, indicating exactly how many days you will suffer from an allergic reaction *if* you are allergic to that particular substance. An allergic reaction starts to show almost immediately after you have been exposed to an allergen which you are allergic to. The test scheme has two action points per day:

I At 8 o'clock each morning, at most one of the allergens is applied to your body.

II At 8 o'clock each evening, you are examined for allergic reactions.

Thus an allergen with live duration D will affect exactly D allergic reaction examinations.

Of course, if you have two or more active allergens in your body at the time of an observed reaction, you cannot tell from that information only, which of the substances you are allergic to.

You want to find the shortest possible test scheme given the durations of the allergens you want to test. Furthermore, to allow simple large scale application the test scheme must be non-adaptive, i.e. the scheme should be fixed in advance. Thus you may not choose when to apply an allergen based on the outcome of previous allergic reaction examinations.

Input specifications

The first line of the input contains a single integer k ($1 \leq k \leq 20$) specifying the number of allergens being tested for. Then follow k lines each containing an integer D ($1 \leq D \leq 7$) specifying the live duration of each allergen.

Output specifications

The number of days of the shortest conclusive non-adaptive test scheme.

A scheme ends the morning when you no longer have active allergens in your body, thus a test scheme for a single allergen with live duration D takes D days.

Sample input 1	Sample output 1
3 2 2 2	5

Sample input 2	Sample output 2
5 1 4 2 5 2	10

Problem D

Rain Fall

Rainfall is measured in millimeters. The rain is collected in a vertical transparent tube with millimeter markings, and once the rain has stopped falling, one can check the height of the water in the tube.

In our problem, the tube unfortunately has a leak at height L millimeters (mm). If the water level is above the leak then water drains from the tube at a rate of K millimeters per hour (mm/h).

We want to figure out how much rain fell during a particular rainfall. *We assume that the tube is high enough that it does not overflow. We also assume that rain falls at an (unknown) uniform rate during a rainfall, and that water does not evaporate from the tube. The height of the leak itself is also negligible.*

Input specifications

The input is a line with five positive numbers: L K T_1 T_2 H where

L is where the leak is (mm)

K is the rate at which water leaks (mm/h)

T_1 is the duration of the rainfall (h)

T_2 is the time between the end of the rainfall and the observation of the water level (h)

H is the water level in the tube when we observe it (mm)

Each number is at least 0.01 and at most 1000.00, and each is given with two decimals.

Output specifications

One line with two floating point numbers F_1 F_2 where F_1 is the smallest rainfall in millimeters that would result in the given observation, and F_2 is the largest rainfall in millimeters that would result in the given observation. Values with either absolute or relative error smaller than 10^{-6} are acceptable.

Sample input 1	**Sample output 1**
`80.00 0.50 2.00 1.50 80.00`	`80.000000 80.759403`

Sample input 2	**Sample output 2**
`150.00 1.00 100.00 150.00 100.00`	`100.000000 100.000000`

Problem E

Speedy Escape

The Newton brothers are planning to rob a bank in the city of Alviso and want to figure out a way to escape the city's only police car. They know that their car is faster than the police car so if they could just reach one of the highways exiting the city they will be able to speed away from the police.

The police car has a maximum speed of 160 km/h. Luckily, the brothers know where the police car will start (it's parked at the police station). To be on the safe side they assume that the police car will start moving as soon as they leave the bank and start their car (this is when the alarm goes off).

The brothers want to find a fixed route that ensures that they are able to leave the city no matter what route the police car take and at what speed it drives. However, since the brothers are not very confident drivers they don't want to drive faster than necessary. Luckily they have recently invested in a new hi-tech in-car police escape system that *you* have constructed. This system will tell them what the minimal top speed needed to escape is (and probably other useful things like what route to take).

Let's turn the clock back a bit to the time when you were constructing the escape system and focused on finding the minimal required speed. Can you get it right?

You may treat all roads as infinitesimally narrow and both cars as point objects. If the brothers ever end up at the same point (on any road or intersection) at the same time as the police car they will be caught and by Murphy's law if there is any possibility of this happening it will happen. The two cars start simultaneously and can accelerate/decelerate instantaneously at any time to any speed below or equal to its maximum speed. They can also change roads at intersections or direction anywhere on a road instantaneously no matter what speed they are traveling at.

Input specifications

The first line of the input consists of three integers n, m and e, where $2 \leq n \leq 100$ describe the number of intersections, $1 \leq m \leq 5000$ describes the number of roads in the city and $1 \leq e \leq n$ describes the number of highway exits. Then follow m lines, each consisting of three integers a, b, l such that $1 \leq a < b \leq n$ and $1 \leq l \leq 100$ describing a road of length l hundred meters from intersection a to intersection b. Then follows a line of e integers, each one a number in $1, \ldots, n$ describing which intersections are connected to highway exits. Finally there is a line with two integers b and p ($1 \leq b, p \leq n$ and $b \neq p$) describing the intersections where the brothers and the police cars start, respectively.

It will always be possible to travel from any intersection to any other intersection. Roads are only connected at intersection points (although they may cross using bridges or tunnels at others points). Roads can be used in both directions but there cannot be more *than one road between two intersections.*

Output specifications

The minimal speed in km/h required to escape or the word `IMPOSSIBLE` if it is impossible. In the first case any answer with either absolute or relative error smaller than 10^{-6} is acceptable.

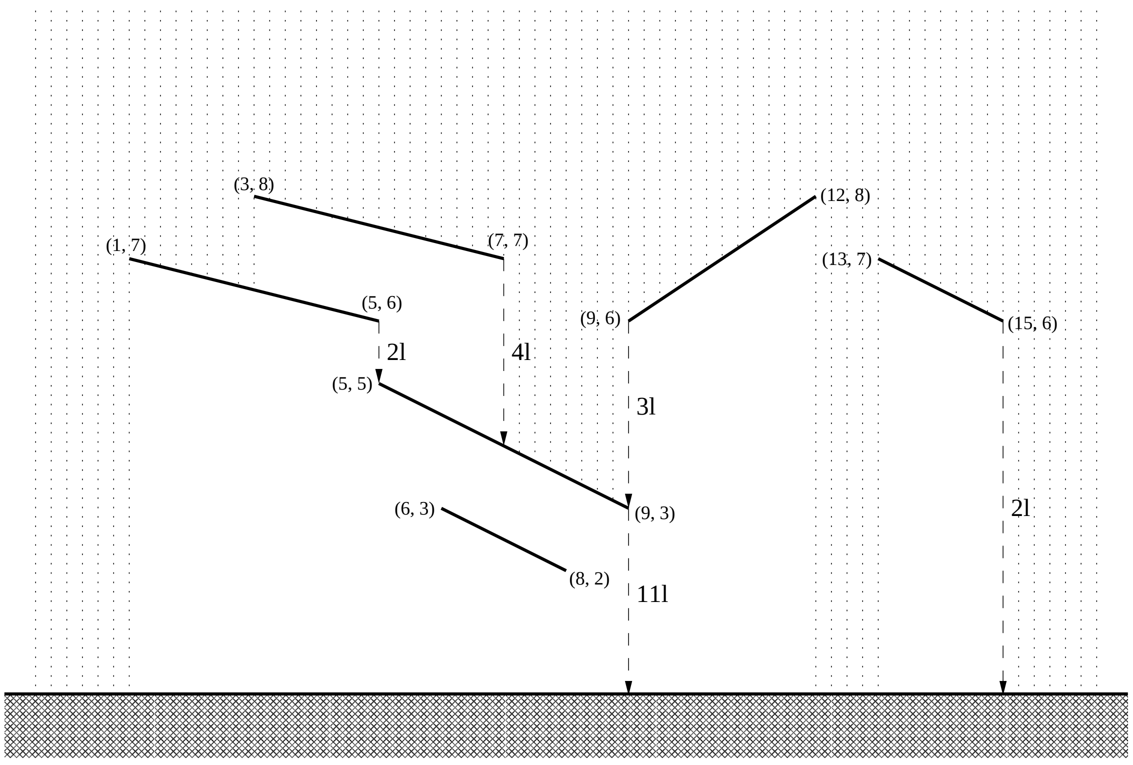
Sample input 1	**Sample output 1**
`3 2 1`	`IMPOSSIBLE`
`1 2 7`	
`2 3 8`	
`1`	
`3 2`	

Sample input 2	**Sample output 2**
`3 2 1`	`74.6666666667`
`1 2 7`	
`2 3 8`	
`1`	
`2 3`	

Sample input 3	**Sample output 3**
`4 4 2`	`137.142857143`
`1 4 1`	
`1 3 4`	
`3 4 10`	
`2 3 30`	
`1 2`	
`3 4`	

November rain

Contemporary buildings can have very complicated roofs. If we take a vertical section of such a roof it results in a number of sloping segments. When it is raining the drops are falling down on the roof straight from the sky above. Some segments are completely exposed to the rain but there may be some segments partially or even completely shielded by other segments. All the water falling onto a segment flows as a stream straight down from the lower end of the segment on the ground or possibly onto some other segment. In particular, if a stream of water is falling on an end of a segment then we consider it to be collected by this segment.



For the purpose of designing a piping system it is desired to compute how much water is flowing down from each segment of the roof. To be prepared for a heavy November rain you should count one liter of rain water falling on a meter of the horizontal plane during one second.

Task

Write a program that:

- reads the description of a roof,
- computes the amount of water flowing down in one second from each segment of the roof,
- writes the results.

Input

The first line of the input contains one integer n ($1 \leq n \leq 40\ 000$) being the number of segments of the roof. Each of the next n lines describes one segment of the roof and contains four integers x_1, y_1, x_2, y_2 ($0 \leq x_1, y_1, x_2, y_2 \leq 1\ 000\ 000$, $x_1 < x_2$, $y_1 \neq y_2$) separated by single spaces. Integers x_1, y_1 are the horizontal position and the height of the left end of the segment respectively. Integers x_2, y_2 are the horizontal position and the height of the right end of the segment respectively. The segments don't have common points and there are no horizontal segments. You can also assume that there are at most 25 segments placed above any point on the ground level.

Output

The output consists of n lines. The i-th line should contain the amount of water (in liters) flowing down from the i-th segment of the roof in one second.

Example

For the input:

```
6
13 7 15 6
3 8 7 7
1 7 5 6
5 5 9 3
6 3 8 2
9 6 12 8
```

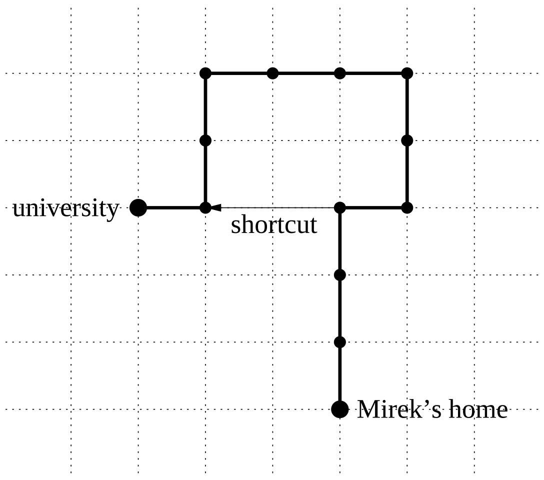
the correct answer is:

```
2
4
2
11
0
3
```

Shortcut

Mirek has a favourite way from home to the university that he traverses every working day. The route consists of sections and each section is a straight segment 10 meters long. Each section is either a straight ahead extension of the previous section or it is perpendicular to the previous section. After traversing each section Mirek takes a small break to admire the beauty of the nature. During his walk he never visits the same place twice.

Yesterday Mirek stayed up long in the night at the party and today he got up late from bed. He knows that he will miss the first lecture unless he changes his usual route. He plans to make one shortcut but he wants the shortcut to be as short as possible (well, we can tell you in secret that he doesn't want to be on time, he just wants to calm his conscience). The shortcut must be either a horizontal or vertical segment connecting two break points of Mirek's route.



Please help Mirek find the shortest shortcut.

Task

Write a program that:

- reads Mirek's route,
- computes the shortest shortcut on the route,
- writes the result.

Input

The first line of the input contains one integer n ($3 \leq n \leq 250\ 000$) being the number of sections of the route. The second line of the input contains a sequence of n characters `N`, `E`, `S` or `W` with no spaces in between. Each character is a description of one section of the route. Character `N`, `E`, `S` or `W` means that Mirek walks 10 meters north, east, south or west respectively. You may assume that at least one shortcut exists for the given route.

Output

The first and only line of the output contains integers l, b, e and character d separated by single spaces. Integer l is the length of the shortest shortcut (measured in 10 m segments). Integers b and e are the numbers of break points where the shortcut begins and ends respectively (we number break points with consecutive integers from 0 for Mirek's home to n for the university). Character d is the direction of the shortcut. If more than one shortcut of the minimal length exists you should output the one that begins earliest on the route. If more than one shortcut of the minimal length begins at the same break point you should output the one that ends furthest on the route.

Example

For the input:

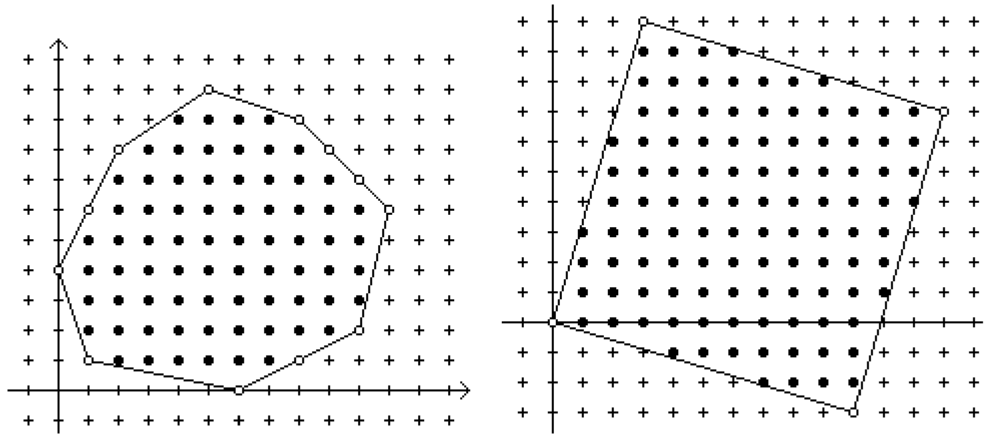
```
12
NNNENNWWWSSW
```

the correct answer is:

```
2 3 11 W
```

H • Interior Points of Lattice Polygons

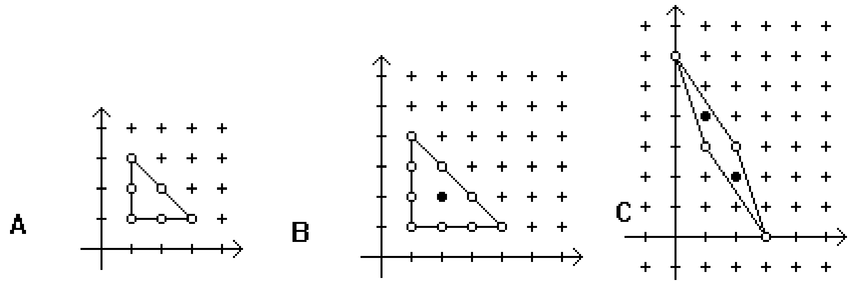
A **lattice point** is a point with **integer** coordinates. A **lattice polygon** is a polygon with all vertices lattice points.



The lattice points on the boundary of the polygon are **boundary points** (open dots in the figure above) and the points inside and not on the polygon are **interior points** (filled in dots in the figure above).

A polygon is **convex** if any line segment between two points of the polygon is inside (or on the boundary of) the polygon. Equivalently, the interior angle at each polygon vertex is less than 180 degrees. Note that any line between two points inside (and not on the boundary of) the polygon is entirely inside (and not on the boundary of) the polygon.

The interior points of a convex lattice polygon on any horizontal line form a single segment from a leftmost point to a rightmost point (which may be the same). Note that there may be no interior points (A), or only one (B), or isolated points (C) as shown in the figures below.



Write a program that reads the vertices of a convex lattice polygon in standard order and outputs the interior points as a list of horizontal line segments. The vertices of a lattice polygon are in *standard order* if:

a) The first vertex is the one with the largest ***y*** value. If two vertices have the same ***y*** value, the one with the smaller ***x*** value is the first.
b) Vertices are given in clockwise order around the polygon.

Input

The first line of input contains a single integer ***P***, (1 ≤ ***P*** ≤ 1000), which is the number of data sets that follow. The first line of each data set contains the data set number, followed by a space, followed by a decimal integer giving the number vertices ***N***, (3 ≤ ***N*** ≤ 50), of the polygon. The remaining lines in the data set contain the vertices, one per line in standard order. Each line contains the decimal integer ***x*** coordinate, a space and the decimal integer ***y*** coordinate.

Output

For each data set there are multiple lines of output. The first line contains a decimal integer giving the data set number followed by a single space, followed by a decimal integer giving the number of horizontal lines which contain interior points (this may be zero (0) or more). The lines of interior points, if any, follow, one per line in order of *decreasing* ***y*** value. Each line contains the decimal integer ***y*** coordinate, a single space and the decimal integer ***x*** coordinate of the left most point, a single space and the decimal integer ***x*** coordinate of the right most point.

Sample Input

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

Sample Output

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

It Can Be Arranged

Every year, several universities arrange inter-university national programming contests. ACM ICPC Dhaka site regional competition is held every year in Dhaka and one or two teams are chosen for ACM ICPC World Finals.

By observing these, MMR (Mission Maker Rahman) has made a plan to open a programming school. In that school, N courses are taught. Each course is taught every day (otherwise, programmers may forget DP while learning computational geometry!). You will be given the starting time A_i and finishing time B_i (inclusive) of each course i ($1 \leq i \leq N$). You will be also given the number of students registered for each course, S_i ($1 \leq i \leq N$). You can safely assume no student has registered to two different courses. MMR wants to hire some rooms of a building, named *Sentinel Tower*, for running that school. Each room of Sentinel Tower has a capacity to hold as much as M students. The programmers (students) are very restless and a little bit filthy! As a result, when $\texttt{course}_i$ is taken in a class room, after the class is finished, it takes $\texttt{clean}_{ij}$ time to clean the room to make it tidy for starting teaching $\texttt{course}_j$ immediately just after $\texttt{course}_i$ in the same room.

Your job is to help MMR to decide the minimum number of rooms need to be hired to run the programming school.

INPUT

Input starts with an integer T ($T \leq 100$) denoting the number of test cases. Each case starts with two integers N ($1 \leq N \leq 100$), number of courses and M ($1 \leq M \leq 10000$), capacity of a room. Next N lines will contain three integers A_i, B_i ($0 \leq A_i \leq B_i \leq 10000000$) and S_i ($1 \leq S_i \leq 10000$), starting and finishing time of a course. Next N lines will contain the clean time matrix, where the i^{th} row will contain N integers $\texttt{clean}_{ij}$ ($1 \leq i \leq N$, $1 \leq j \leq N$, $0 \leq \texttt{clean}_{ij} \leq 10000000$, $\texttt{clean}_{ii} = 0$).

OUTPUT

For each case, print the test case number, starting from 1, and the answer, minimum number of rooms needed to be hired.

SAMPLE INPUT

```
3
1 5
1 60 12
0
4 1
1 100 10
50 130 3
150 200 15
80 170 7
0 2 3 4
5 0 7 8
9 10 0 12
13 14 15 0
2 1
1 10 1
12 20 1
0 2
5 0
```

SAMPLE OUTPUT

```
Case 1: 3
Case 2: 22
Case 3: 2
```