FAU Winter Contest 2013

February 02, 2013

The Problem Set

No	Title
A	A clear shot
B	Army
C	Bus Tickets
D	Card count
E	Cave
F	Dressed for success
G	Hack the Reactor
H	Moving Target
I	Password hints
J	Priorities
K	Time Defusal
L	Tube Marathon

Good luck and have fun!

hosted by

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

sponsored by

accenture
High performance. Delivered.

Problem A

A clear shot

James is fighting a bad guy on top of a moving train. You are his assisting field agent and M has ordered you to shoot the bad guy. You've positioned yourself along a stretch of railway track in between tunnels to make the shot. Unfortunately, there is not just the difficulty of trying to hit the bad guy while James is fighting him; there are also obstacles obstructing your clear view of the full track length in between the tunnels. To optimally prepare for taking the shot, you'd like to determine the longest consecutive interval of train track to which you have a clear view for making the shot.

The train track and obstacles are described in reference coordinates around you. That is, your position is at the origin $(0, 0)$. The train track is always a straight, horizontal line segment that lies North of you ($y > 0$). For simplicity all obstacles are considered to be circles, and no obstacle overlaps either you or the railway track; obstacles are allowed to overlap each other. All obstacles lie completely in the half plane given by $y > 0$.

Input

The first line contains the number of test cases t ($1 \leq t \leq 100$). Then follows for each test case:

- a line containing three integers x_a, x_b, y ($-10\,000 \leq x_a < x_b \leq 10\,000, 1 \leq y \leq 10\,000$), the coordinates (x_a, y) and (x_b, y) of the railway track endpoints;
- a line containing an integer n ($0 \leq n \leq 1\,000$), the number of obstacles;
- for each obstacle a line containing three integers x_i, y_i, r_i ($-10\,000 \leq x_i, y_i \leq 10\,000, 1 \leq r_i \leq 1\,000$), the circle center (x_i, y_i) and its radius r_i.

Output

For each test case, print on a separate line the maximum consecutive track length onto which you have a clear view. If you cannot see any of the track, then print "`IMPOSSIBLE`".

Your answer should have an (either relative or absolute) precision of 10^{-6}. If you do have a clear shot, then it will be for at least a length of 10^{-4}.

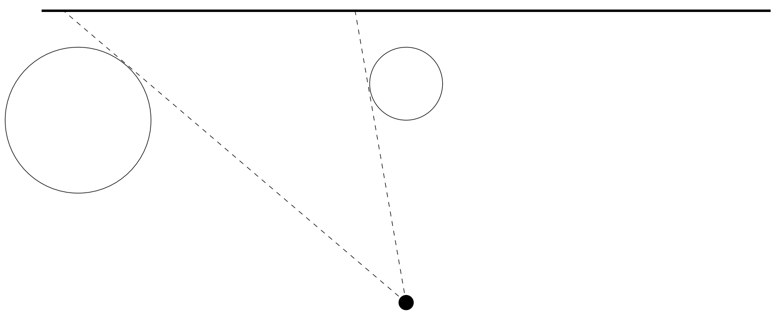

Figure 1 – the first example; the longest visible track is right of the object in the middle.

Sample Input	Sample Output
3	8.64775
-10 10 8	IMPOSSIBLE
2	10
-9 5 2	
0 6 1	
0 8 10	
1	
1 5 4	
-10 10 8	
2	
-1 2 1	
10 10 1	

This page is intentionally left (almost) blank.

Problem B

Army

The MI6 fears that one soldier in the British army is a spy. They (even James Bond himself) tried several times to find the spy — they know what he looks like. But it seems he is always hiding among the other soldiers. Luckily, next week there will be a military parade in honour of the Queen's birthday.

You might not know, but the British army has a funny tradition at these parades: they always form a rectangle with all soldiers, i.e. the soldiers line up in rows and columns so that the number of soldiers in each row is the same (and of course in each column respectively).

The MI6 wants to avoid that the spy is able to hide somewhere in the group again. So, not only James Bond is going undercover as a soldier to the parade but also some other agents (if necessary). It should be simple to find the spy, if the army can only form line ups with a *single* row or a *single* column according to the rules above. The soldiers always look in the same direction and we cannot recognize the spy from his back, so two rows/columns are not an option.

Input

The first line contains the number of test cases t ($1 \leq t \leq 100$). Each case is given as one integer a on a line ($1 < a \leq 10^5$), where a is the number of soldiers in the army without undercover agents.

Output

For each test case, print one line containing the number of necessary undercover agents including James Bond who will always "join" the parade.

Sample Input	Sample Output
5	4
7	3
8	1
42	3
14	10
999	

This page is intentionally left (almost) blank.

Problem C
Bus Tickets

James Bond is currently on a secret mission in Absurdistan. As you might have heard, Absurdistan is a very strange country. For example, they only have one road in the country and all cities lie on this road and are numbered $1, \ldots, n$ from west to east.
There are several bus lines on this road, some go from west to east and some from east to west. In this problem, James is trying to get from city 1 to city n, so we only consider buses going from west to east.
Each bus line has a starting city s and an ending city e. The state-run bus company wants to make the life of the bus drivers easier, so they stop in all cities $s, s+1, \ldots, e$ and thus don't have to remember where to stop the bus — they simply stop in each city. To make their lives even easier, a ticket for each bus line has a fixed price, no matter where you start or how many stops you go. You might think this is absurd, but in Absurdistan they prioritize absurdity over logic.
James is now in city 1 but needs to get to city n to fulfill his mission. How can he get there in the cheapest possible way?

Input

The first line contains the number of test cases t ($1 \leq t \leq 100$). Then follows for each test case:

- a line containing an integer n ($1 \leq n \leq 2\,000$), the number of cities in Absurdistan.
- $n-1$ lines, i-th containing integers $p_{i,i+1}, p_{i,i+2}, \ldots, p_{i,n}$. Number $p_{i,j}$ is the price of a ticket for a bus line going from city i to city j. If $p_{i,j} = -1$, then there is no bus line starting in i and ending in j. You can assume that all prices are positive integers and at most $200\,000$.

Warning: The input is large, make sure you use a fast input routine.

Output

For each test case, print on a separate line the minimum cost of tickets that allow James to travel from the first city 1 to the last one n. If there are no such tickets, print "`IMPOSSIBLE`".

Sample Input

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

Sample Output

```
5
8
```

This page is intentionally left (almost) blank.

Problem D
Card count

James Bond loves to go to the casino, and he has lately discovered that Auric Goldfinger cannot stand to lose when playing cards and might have been cheating, so he decides that would be a good idea to have a cheating detector device to expose the dirty guy. 007 is an expert on shooting, girls and drinking martinis, but he has no idea of programming. This is why he wants to hire you as a developer, but obviously you have to pass the interview first. To test your programming and engineering skills, he asks you to develop a card counting program. He tells you that they always represent the value of a card with a character v ($v \in \{\texttt{2}, ..., \texttt{9}, \texttt{T}, \texttt{J}, \texttt{Q}, \texttt{K}, \texttt{A}\}$), where the special characters $\texttt{T}, \texttt{J}, \texttt{Q}, \texttt{K}, \texttt{A}$ represent a card with a value of 10, 11, 12, 13, and 14, respectively. He also tells you that the suit (Diamonds, Clubs, Hearts or Spades) of a card is given by a character s ($s \in \{\texttt{D}, \texttt{C}, \texttt{H}, \texttt{S}\}$). Your card counting program should be able to tell the highest card that have not yet been drawn from a deck with 52 cards, given that the ordering of the cards is first by value and then by suit, where $D < C < H < S$.

Input

The first line contains the number of test cases t, ($1 \leq t \leq 100$). Then follows for each test case:

- A line containing an integer n ($1 \leq n \leq 50$), the number of cards that have already been drawn.
- Then follows n lines that represent the cards that have already been drawn. Each line contains a character v for the card value and a character s for the suit.

Only uppercase letters will appear in the file.

Output

For each test case, print on a separate line the highest card that have not yet been drawn using the same representation for the value and suit as in the input file.

Sample Input

```
2
15
K C
Q S
A D
7 C
T H
A C
K D
K H
A S
5 H
A H
4 C
3 H
8 D
K S
4
T C
Q C
3 H
4 S
```

Sample Output

```
Q H
A S
```

This page is intentionally left (almost) blank.

Problem E
Cave

James Bond is on a space mission to the moon. According to current MI6 information, the next generation of terrorist superstars shelter in a cave labyrinth on the backside of the moon while planning attacks on the United Kingdom. The villains are known to defend themselves by simply flooding the whole cave complex with a superjellygoo to drown all intruders. The goo is produced by Dr. No & Sons Inc. and behaves like water on planet earth. Despite this sophisticated defense mechanism, James must hunt them down. He obtained some pretty reliable maps from the lunar Bond girl. In such a map, an underscore `_` denotes a cave and a hash `#` represents a wall. Using these maps, James wants to assure that he is able to enter the caves, and survive in a safe spot, even when the villains flood it. The flooding starts from all boundary positions of the map. Once a cave is full, its neighbouring caves (left, right, up, and down) will also be flooded whenever the neighbour is no wall. The flooding does not occur diagonally.
James got a special gizmo for this mission that can drill through walls and afterwards reseal them. Hence he simply has to determine whether there exists at least one safe spot after the cave is completely flooded. His gizmo and his superagenty senses ensure that James will always reach such a place whenever it exists.

Input

Input starts with one line, containing an integer n, denoting the number of test cases (at most 100). Each test case starts with one line, containing two integers w and h ($2 \leq w, h \leq 100$), describing the width and height of the cave system. After that follow h lines, containing w characters, which are either cave (denoted by "`_`", there is at least one cave on every map) or wall (denoted by "`#`").

Output

For each test case, print one line, containing "`SURVIVE`" if there is at least one remaining safe space, or, if James Bond will drown, print "`RIP`".

Sample Input

```
4
3 3
_#_
_#_
___
5 5
__##_
_###_
_#_#_
_###_
_____
5 5
_____
_####
_#___
_####
_#_#_
3 7
###
###
#_#
###
###
###
#_#
```

Sample Output

```
RIP
SURVIVE
RIP
SURVIVE
```

This page is intentionally left (almost) blank.

Problem F

Dressed for success

James Bond was invited to a very exclusive party. He got very upset with the last Bond girl who served him a stirred martini, so he has to find a new girl to accompany him and be the next Bond girl. He asks his loyal housekeeper and friend May Maxwell to prepare a casting where the candidate girls will have the opportunity to spend some time with 007 himself. There is, obviously, a lot of interest among the local girls to participate and it is decided that they will be casted one at a time. On each candidate's turn, May, who knows James' taste to perfection, assigns the girl a score s (0 = No way of being chosen, 1 = The perfect Bond girl) and gives her a dress to put on and some make-up. She finally asks the candidate to say the most probable time t_d (in minutes) it is going to take her to get ready, because an old statistician friend of hers who has 12 sisters told her that the time t it takes to a woman to get ready follows a continuous Erlang-k distribution with parameters λ (rate) and k (shape), where both parameters are related by $t_d = \frac{k-1}{\lambda}$. The probability density function (*) for this distribution is given by

$$f(t, k, \lambda) = \frac{\lambda^k t^{k-1} e^{-\lambda t}}{(k-1)!}$$

You can assume that May can draw upon her friend to know the exact value of k for each girl, and that the score given by May is independent of t_d. We also know that James is very picky. He will refuse to accept a candidate if she takes less than t_{min} minutes to get ready, because he will think that she put too little effort in doing so. He will also refuse a girl who takes more than t_{max} minutes, because his time is precious. If a girl is on time, she will be picked with a probability given by her score. But if she is not on time, then she will not be picked, even if she is a perfect Bond girl.

(*) *If a random variable T follows a distribution with probability density function $f(x)$ then the probability $P(x \leq T < x+h)$ is given (approximately) by $f(x) \cdot h$, where h is an infinitesimally small positive number ($h \approx 0$).*

Input

- A line containing two integers t_{min}, t_{max} ($5 \leq t_{min} < t_{max} \leq 60$), the minimum and maximum time in minutes 007 expects a girl to take to be ready, respectively.
- A line containing an integer n ($1 \leq n \leq 25$), the number of girls in the casting.
- Then follow $3n$ lines, where every line group corresponds to
  - A line containing a floating point number s ($0 \leq s \leq 1$), the score for the girl.
  - A line containing an integer k ($k \in \{2, 3, 4, 5\}$), the Erlang-k distribution shape parameter.
  - A line containing an integer t_d ($15 \leq t_d \leq 50$), the most probable time in minutes for the girl to get ready.

Output

Print on n separate lines the *individual* probability for each of the girls to be accepted as the next Bond girl, i.e., assuming that this probability is independent of James having seen any other candidate. Your answer should have an (either relative or absolute) precision of 10^{-6}.

Sample Input

```
10 45
4
0.342
5
32
0
3
47
0.9419
4
15
0.7
2
26
```

Sample Output

```
0.2231553
0
0.7873315
0.3211423
```

Problem G
Hack the Reactor

Darn! Dr. No's son Dr. Nono harasses America again by radio jamming from a new location. Because Dr. Nono is smarter than Dr. No, he secured the reactor against sabotage with a unbelievably strong code. James wants to stop Dr. Nono's evil plans, and managed to steal a program which can calculate the code needed to deactivate the reactor which serves the jammer with energy. Unfortunately, 007 has no interpreter which can execute the program and Q is on vacation. You are the last hope to help James with an interpreter and stopping the plans of Dr. Nono.
The program contains multiple (at least two) instructions and there are two types of instructions:

- Read Number, Format: `R <nr>`
  This instruction reads one integer nr ($0 \leq nr \leq 10^9$) into the memory. It starts with the opcode `R` followed by one space and the number.
- Print Median, Format: `M`
  This instruction prints the median of all numbers currently held in memory on a single line. If the count of numbers is odd it prints one median, else it prints the lower and upper median in ascending order separated by a single space.

The code to deactivate the reactor is the output of the program if it is executed correctly. You may safely assume that each program starts with a read instruction and ends with a median instruction. Needless to say that James needs your help as fast as possible.

Input

The first line contains a single integer n ($1 < n \leq 10^6$) denoting the number of instructions of the stolen program. Then, n lines follow with one instruction per line.

Output

Print the code for the reactor after a correct execution of the stolen program, i.e., the output of the program.

Sample Input

```
7
R 3
R 1
M
R 6
M
R 2
M
```

Sample Output

```
1 3
3
2 3
```

This page is intentionally left (almost) blank.

Problem H
Moving Target

Q always creates fancy gadgets, which then will be used by James Bond either to spy, to chase or even to kill his enemies (but times are over where Q creates bombs in pens). To be honest, Q does not do all the work on his own, he rather presents *your* work. So Q has a new job for you: you should create a program that keeps track of a moving target.
All you know is that the target moves according to a polynomial with degree three (or less). You observe the target for some time and write down its coordinate at time stamp 0, 1, 2, and 3. Can you compute the target's coordinate at time stamp t with that information?

Input

The first line contains the number of test cases C ($1 \leq C \leq 100$). Each case is specified in a single line with five values that give the coordinate of the target at the times 0, 1, 2, and 3 followed by the time stamp t. Both the coordinates and the time stamp are non-negative integers not larger than 15 000.

Output

For each test case, print one line containing the coordinate of the target at time stamp t.
Another team inside Q's section already has proven that the result is always an integer and the absolute value is smaller than 2^{58} (thus, use a 64-bit integer data type for your calculations such as `long long` in C/C++ or `long` in Java).

Sample Input

```
4
0 11111 0 11111 11111
0 3 14 39 4
1 42 165 370 10
47 11 8 15 23
```

Sample Output

```
10156531557679871
84
4101
-33165
```

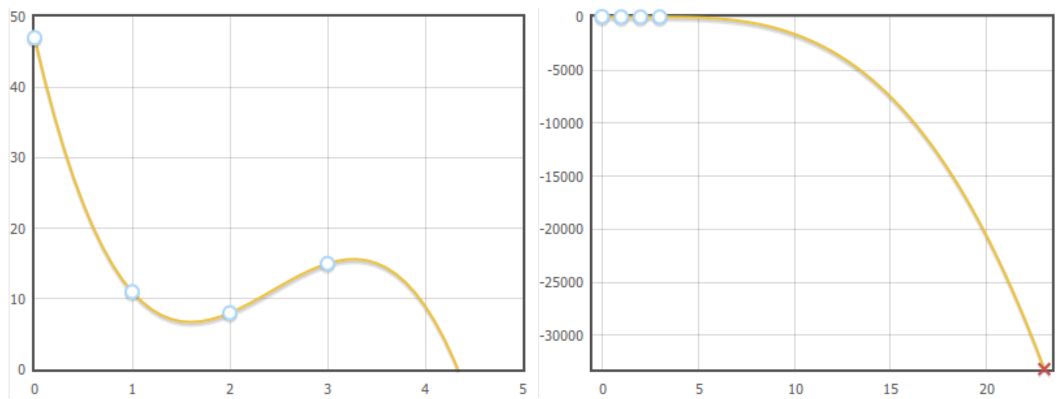

Figure 2 – Illustration of the last polynomial in the sample: $-\frac{23}{6}t^3 + 28t^2 - 60\frac{1}{6}t + 47$.
The first three sample cases describe the polynomials $7407\frac{1}{3}t^3 - 33333t^2 + 37036\frac{2}{3}t$, $t^3 + t^2 + t$, and $41t^2 + 1$.

This page is intentionally left (almost) blank.

Problem I
Password hints

The MI6 (secret intelligence service) has several sections that do criminal investigations, e.g. for special agents like James Bond. The current target is an arms factory, or more precisely, the password of its main server. The password is long and complicated; to be able to reconstruct the password once forgotten, there are a lot of password hints in the factory on various objects. One section of MI6 hopefully found all of these hints. The job of your section is to resolve them.
Each hint is an equation of integers that looks like $a + b = c$ or $a - b = c$, which can be wrong. Now the equation is interpreted with matches as shown in the pictures (without leading zeros).

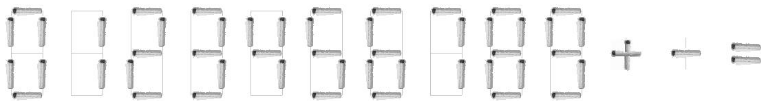

Figure 3 – Numbers and symbols represented by matches.

You have to remove a given number of R matches and you have to insert a given number of I matches. It is allowed to remove or add complete digits on the left side of the numbers (leading zeros are still not allowed). It is not allowed to insert matches at positions where matches are removed. After removal and insertion of the matches the equation has to be correct. This means that the equation has the shape $a+b = c$ or $a-b = c$ and the represented numbers are positive integers less than 1 000 and the operation leads to the result on the right side of the equation.
If you supply us the needed information, then a third section of MI6 can reconstruct the actual correct password. For security reasons, we are not allowed to tell you how this is done, but for sure your help is necessary to attack the target.

Input

The first line of input contains a single integer H: the number of hints ($0 < H \leq 20$). The following H lines each contain an equation and the two integers R and I as specified above. The equations contain no spaces. a, b and c are positive integers less than 1 000, while R and I are non-negative integers less than 100.

Output

For each hint in the input produce two lines of output. The first line should contain the number of different correct equations that can be produced by the current hint. The second line should contain the smallest equation that can be generated by the current hint, or "`IMPOSSIBLE`" if no equation fulfills the given conditions. The smallest equation is the equation which has the smallest first number (in case of a tie the smallest second and third number, respectively).

Sample Input

```
5
1+1=2 0 0
1+2=1 0 0
2+1=1 1 0
1+1=8 0 1
11-11=112 2 3
```

Sample Output

```
1
1+1=2
0
IMPOSSIBLE
1
2-1=1
2
1+7=8
2
1+111=112
```

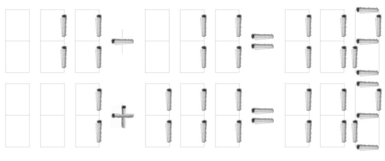

Figure 4 – Last password hint (input and correct output).

This page is intentionally left (almost) blank.

Problem J
Priorities

So far James had some good shootings, had some martini's, rescued the world and got the girl in every mission. But nowadays even the MI6 has to save money, so the time for a mission is limited and the agents can't always fulfill all planned tasks. James gets offered some missions and has to decide if he wants to take them or reject them. He wants you to judge if he could fulfill all mission tasks. If not, you should give hints if he just can't make all secondary goals (having some martini's or having a good shooting), or if one of the primary goals may fail (rescuing the world or getting the girl). He has made a plan with tasks for every mission and tells you how long a task takes, when it must be finished, and for what goal it is necessary.

Input

The input starts with the number of test cases (at most 20). Each test case starts with the number of mission tasks (at most 100 000). Every task is described in one line consisting of the time d it takes to fulfill this task, the due time t for fulfilling this task ($0 < d, t \leq 1\,000\,000$) and a string s describing for which goals this task is important. s contains between 1 and 4 letters (`g`, `m`, `s` or `w`), meaning this task is necessary for getting the (`g`)irl, getting some (`m`)artini's, having a good (`s`)hooting or rescuing the (`w`)orld. To fulfill one goal, James has to fulfill all tasks marked with the corresponding letter.
Warning: The input is large, make sure you use a fast input routine.

Output

For every test case output one line:

- `CINEMATIC` If James can fulfill all tasks.
- `GIRL+WORLD` If James can fulfill the tasks needed to get the girl and the tasks needed to rescue the world, but has to give up some secondary goals.
- `GIRL` If James can fulfill the tasks to get the girl, but not the tasks to rescue the world.
- `WORLD` If James can fulfill the tasks to rescue the world, but not the tasks to get the girl.
- `HMM` If James can fulfill either all tasks to rescue the world or all tasks to get the girl.
- `NOTHING` If James cannot fulfill all tasks to rescue the world nor all tasks to get the girl. (He may still get some martini's.)

Sample Input

```
2
4
1 25 m
10 10 gw
10 20 gm
10 30 sw
3
10 10 gw
10 20 gm
10 20 sw
```

Sample Output

```
GIRL+WORLD
HMM
```

This page is intentionally left (almost) blank.

Problem K
Time Defusal

Red alert for the City of London! The new evil overlord Marina — yes, a female overlord — isn't satisfied with blowing up MI6-HQ, but wants to eliminate the entire capital with one gigantic blast. Because she knows of the misfortune of her predecessors — their crucial bombs were all defused — she constructed a very tricky time fuse to set off the bomb.
It consists of two rotating rods which are attached to the ground at one end. The circles described by the rotating end meet at exactly one point — and you guessed right: the fuse triggers if the ends meet.
By interrogating an affiliate of Marina's, 007 received the intelligence where to find the bomb and got there right at the moment as the fuse is armed. He catches the weak point of the device at once: the connection between fuse and bomb itself. Knowing where to attack, can you tell Bond how long he has to disarm the bomb?

Input

On the first line the number of test cases N ($1 \leq N \leq 1\,000$). Then for each test case two lines containing three integers; c_i the circumference, v_i the velocity of the rotating end and d_i the length of the segment of the circle between the initial point of the rotor to the point where the circles intersect (in the direction of movement), where $0 < c_i, v_i \leq 55\,000$, $0 \leq d_i < c_i$ and $i \in \{1, 2\}$.

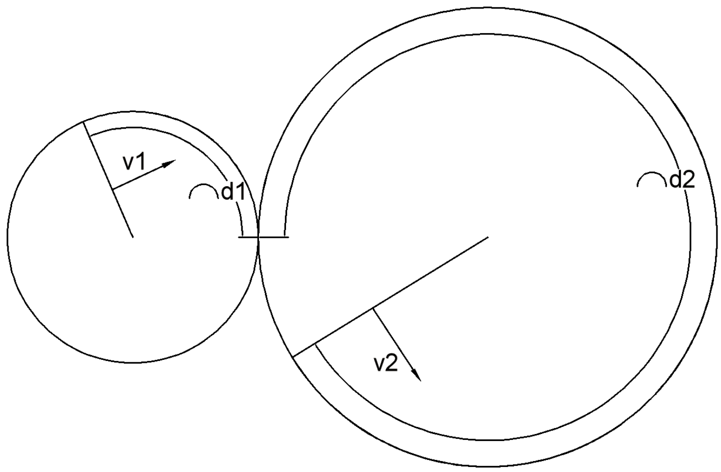

Figure 5 – Visualisation of d_i

Output

For each test case a single line containing the time Bond has to disarm the bomb with relative or absolute error $\leq 10^{-5}$. If the bomb never triggers print "`oo`" (two small 'O').

Sample Input

```
4
23 2 1
42 14 35
23 2 1
42 14 34
23 3 1
42 14 35
50 4 2
23 234 0
```

Sample Output

```
23.5
oo
oo
138
```

This page is intentionally left (almost) blank.

Problem L
Tube Marathon

James is on vacation and bored. He thus decided to participate in the famous London tube marathon. Rules of this marathon are simple: given a tube map participants are required to visit all tube stations in lexicographical order as fast as possible. Next to several different tube lines participants are also allowed to use other means of public transport. However, James is reluctant to use any cycle hire service and thus only uses buses as alternative to the tube service. He thinks it looks ridiculous him riding bikes. Of course he wants to win and thus asks you to determine a strategy resulting in the fastest possible time to complete the marathon.

Further rules of the competition are as following. Once participants arrive at the next tube station to visit they are required to get a stamp at a checkpoint. While you can assume that getting the stamp takes zero time, it takes C_u minutes to get from any tube platform to the checkpoint and C_c minutes to get there from a bus station. Times for getting from the checkpoint to a tube platform or to a bus station, analogously. Checkpoints exist only at stations that are part of the tube system. The competition starts and ends at the checkpoints of the (lexicographically) first and last tube stop respectively. The competition starts at time 0 and ends at time E, finishing at E is fine. For simplicity departure times are given with respect to the starting time of the competition. Times are given in minutes. Participants are allowed to pass/ride through any stations in order to visit the next station on the list.

You are given an underground map consisting of U different tube lines with $N_i^u, i \in \{1..U\}$ tube stations each. You are also given a bus map consisting of B different bus lines with $N_i^b, i \in \{1..B\}$ bus stations each. For each tube/bus line you are given the departure time F_i of the first train/bus, the interval I_i between departing trains/buses and the time L_i of the last train/bus. Trains and buses always depart in opposite directions both at the first and last station of a line on the map. The journey time between tube stations is given as T_u and between bus stations as T_b. Journey times between stations are the same for all lines.
Changing between tube services, bus services or between tube and bus service is possible, but only at stops having an identical name. Changing platforms between tube lines takes C_u minutes, between bus services takes C_b minutes and between corresponding tube and bus stops C_c minutes.

Input

The first line contains the number of test cases T ($1 \leq T \leq 100$). Then follows for each test case:

- A line containing five integers: the end time E ($1 \leq E \leq 1\,440$), the number of tube lines U ($1 \leq U \leq 10$), the number of bus lines B ($0 \leq B \leq 10$), the journey time between tube stations T_u ($1 \leq T_u \leq 10$), and the journey time between bus stops T_b ($1 \leq T_b \leq 10$).
- A line containing three integers: the transfer time between tube platforms C_u ($1 \leq C_u \leq 25$), between bus lines C_b ($1 \leq C_b \leq 25$) and between tube platform and bus stop C_c ($1 \leq C_c \leq 25$). It holds $max(C_b, C_u) \leq C_c$.
- Then follow $2U$ lines, each pair of lines describing one of the tube lines. The first of each pair of lines contains four integers: the departure time of the first train F_i ($0 \leq F_i \leq 1\,440$), the interval between trains I_i ($1 \leq I_i \leq 1\,440$), the departure time of last train L_i ($1 \leq L_i \leq 1\,440$) and the number of stops on this tube line N_i^u ($1 \leq N_i^u \leq 20$). It holds $L_i \geq F_i$ and $L_i - F_i$ is a multiple of I_i. The second line contains N_i^u distinct station names. Each station name consists of between 1 and 25 lower case characters.
- Then follow $2B$ lines describing each of the bus lines. The input format is equivalent to the format of the tube lines.

Output

For each test case output a single line with the minimum time in minutes to visit all tube stations on the map in lexicographic order. If it is not possible to visit all stations within the given time E output "`IMPOSSIBLE`".

Sample Input

```
4
100 1 0 9 9
6 7 8
10 7 80 3
a c b
563 2 0 10 7
3 2 5
100 20 600 5
a b c d e
100 20 600 5
aa bb c dd ee
500 2 1 5 5
5 5 5
0 10 100 10
a b c d e f g h i j
0 10 300 10
k l m n o p q r s t
100 10 200 4
j jj kk k
1000 2 0 5 5
5 5 5
0 10 200 2
a b
0 10 200 2
c d
```

Sample Output

```
60
563
310
IMPOSSIBLE
```