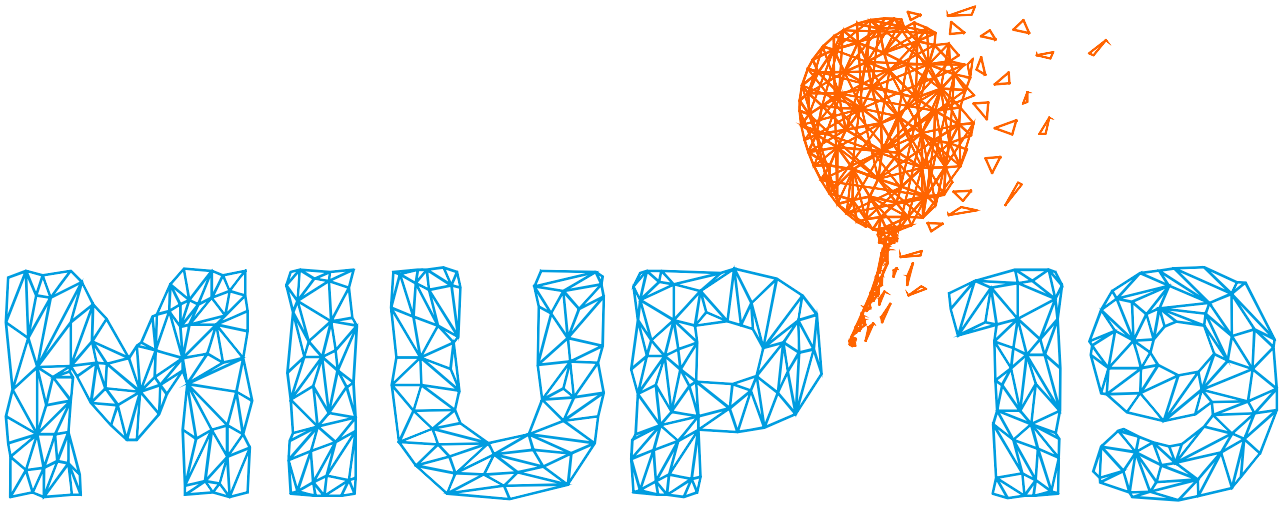


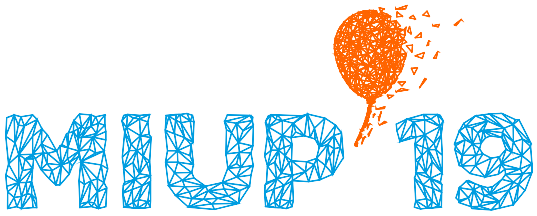
Maratona Inter-Universitária de Programação

Instituto Superior Técnico - 12 de outubro de 2019

TÉCNICO LISBOA

Microsoft

feedzai



Maratona Inter–Universitária de Programação 2019

Teams from the following universities

Contents

Scientific Committee

- André Restivo, FEUP / Universidade do Porto
- Fábio Marques, Universidade de Aveiro
- Filipe Araújo, Universidade de Coimbra
- João F. Ferreira, IST / Universidade de Lisboa
- Margarida Mamede, FCT / Universidade Nova de Lisboa
- Mário Pereira, LRI, Université Paris-Saclay
- Mikolas Janota, IST / Universidade de Lisboa
- Pedro Mariano, ISCTE
- Pedro Ribeiro, FCUP / Universidade do Porto
- Rui Maranhão, IST / Universidade de Lisboa
- Rui Mendes, Universidade do Minho
- Simão Melo de Sousa, Universidade da Beira Interior
- Vasco Pedro, Universidade de Évora

Local Organization Committee

- Alexandre Francisco, IST / Universidade de Lisboa
- Arlindo Oliveira, IST / Universidade de Lisboa
- Francisco Santos, IST / Universidade de Lisboa
- Francisco Miguel Dionísio, IST / Universidade de Lisboa
- José Carlos Monteiro, IST / Universidade de Lisboa
- Luís Guerra e Silva, IST / Universidade de Lisboa
- Luís Russo, IST / Universidade de Lisboa
- Vasco Manquinho, IST / Universidade de Lisboa

Languages and Compilers

1. Files `*.c` are assumed to be C code and are compiled with `gcc 8.3.0` with the command

   ```
   gcc -std=gnu11 -Wall name.c -lm
   ```

   and then executed with the command

   ```
   ./a.out
   ```

2. Files `*.cpp` are assumed to be C++ code and are compiled with `gcc 8.3.0` with the command

```
g++ -std=gnu++14 -Wall name.cpp -lm
```

   and then executed with the command

```
./a.out
```

3. Files `*.java` are assumed to be Java code and are compiled with `OpenJDK 11.0.4` with the command

```
javac -encoding utf8 name.java
```

   and then executed with the command

```
java -Xmx64M -Xss32M -classpath .  name
```

4. Files `*.py` are assumed to be Python 3 code and are executed with `Python 3.7.3` with the command

```
python3 name.py
```

The compilation process can take at most 60 seconds and the maximum source code size is 100 KB. Every program/solution must be submitted in a **single file**. For Java submissions, the file must have the same name as the class that contains the main method. There is no limit for the number of classes to be contained in that file.

Although the maximum run time is defined for each problem, the overall limit is 2 seconds.

Input/Output

All programs should read the input from the standard input, and write the output to the standard output. All lines (both in the input and output) should end with the newline character (`\n`). Except when explicitly stated, single space is used as a separator. No line starts or ends with any kind of white space.

Editors and IDEs

Among others, workstations have these editors and IDEs available:

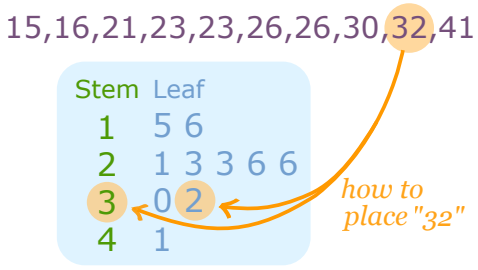
- Atom
- Eclipse
- Emacs
- Geany
- Gedit

- Gobby
- IntelliJ
- Kate
- Nano
- Visual Studio Code
- Vi/Vim
- Wing IDE

Documentation

Language documentation is available. Man pages are also available in your workstation as usual, just use the command `man`.

Problem A: Stem and Leaf Plot



Stem and leaf plots were invented by Tukey as a way of looking at data distribution in the eighties. Their attractiveness was that they could be printed in teletype and were visually pleasing. Nobody uses them today except third graders.

My daughter hates doing them. I decided to help her. Then I made a program. Then I realized they could be used to compare the distribution of two samples. Sorry about that.

A stem and leaf plot takes each number and divides it into the stem (the tenths) and leaf (the ones). For instance, 108 is divided into 10 for the stem and 8 for the leaf. Stems and leaves are ordered in ascending order. The figure above would be represented in text in the following manner:

```
1| 56
2| 13366
3| 02
4| 1
```

In our case, we are representing two samples at the same time. Suppose the second sample contains the following numbers: 5 27 31 32 32 33 35 37 40 41 41 42 43. The second sample is represented to the left side of the stem and its leaves are in descending order.

```
     5 |0|
       |1| 56
     7 |2| 13366
753221 |3| 02
 32110 |4| 1
```

Task

Your task is to build a stem and leaf plot for comparing two samples. You should use pretty printing as can be seen in the output examples. In order to better understand the output, pretty printing will be performed using the `'.'` character instead of spaces.

Input

The input consists of two lines, one for each sample. Each line starts with an integer N that indicates the number of integers V_i that follow, which are the numbers in the sample. Assume that at least one of the samples is not empty.

Constraints

$0 \leq N \leq 10,000$ Number of values in the sample

$1 \leq V_i \leq 1000$ A value of the sample

Output

The stems are ordered in strictly ascending order, are delimited by aligned pipes, and are right justified, possibly making use of dots, if needed. There is a dot after the right pipe and, after the dot, the corresponding leaves of the first sample are written in increasing order. In a symmetrical way, there is a dot before the left pipe, which is preceded by the corresponding leaves of the second sample. These leaves are right justified (possibly making use of dots, if needed) and are in decreasing order (when read from left to right).

Sample Input 1

```
11 29 19 154 17 23 19 49 17 12 41 18
14 34 129 123 24 35 121 12 33 125 51 121 37 32 31
```

Sample Output 1

```
.....2.|.1|.277899
.....4.|.2|.39
754321.|.3|.
.......|.4|.19
.....1.|.5|.
.95311.|12|.
.......|15|.4
```

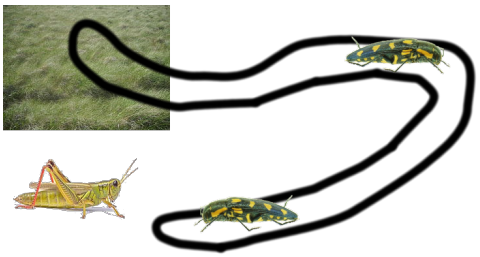
Sample Input 2

```
0
9 34 5 2 38 19 41 17 5 9
```

Sample Output 2

```
9552.|0|.
..97.|1|.
..84.|3|.
...1.|4|.
```

Problem B: GeeWee



GeeWee the grasshopper is going to travel to the land of Plenty where there are huge fields of corn. If he reaches Plenty all his fears of starvation will go away. Being a grasshopper, he uses the same amount of energy for every jump independently of the distance covered. The Waste lands are very windy, and this affects the distance covered by a jump. Whenever GeeWee jumps to one plant he will eat it all to replenish its energy. But the path to Plenty is through the scorched Waste lands, where there are few plants. There are also dragon beetles that will try to eat GeeWee. If GeeWee lands on a spot with a dragon beetle, he jumps backward.

Task

Your task is to compute the distance covered by GeeWee and the energy he consumed on his trip to the land of Plenty. The distance should include both the jumps forward and the jumps backward due to encounters with dragon beetles. To simulate the effect of wind on the distance covered by a jump, you are going to use a linear congruential generator, which uses the recurrence $X_{n+1} = (aX_n + b) \pmod{m}$. The distance covered by a jump varies uniformly between D_l and D_h. That is to say, the jump length at time step $n = 1, 2, \ldots$ is given by expression $D_l + (X_n \pmod{D_h - D_l + 1})$. Note that the seed, X_0, of the linear congruential generator is not used to compute a jump distance.

Whenever GeeWee jumps backward he never goes beyond the starting point. Whenever GeeWee jumps forward he never goes beyond the point just after the last point of the Waste lands. If GeeWee tries to jump but the energy he has left is less than that consumed by a jump, he ends up consuming his last energy and remains in the same spot. If GeeWee's energy reaches 0, he dies.

Input

The first line of the input contains 10 single space-separated integers representing, in this order:

- D_l, D_h — the minimum and maximum distances covered by a jump;
- X_0 — the seed of the pseudo random number generator;
- a, b, m — the parameters that govern the pseudo random number generator;
- E — the initial energy of GeeWee;
- j — the energy consumed by a jump;

- p — the energy stored in a plant;
- l — the length of the terrain.

The second line of the input contains the terrain that GeeWee is going to cross. It contains l characters, representing a spot that GeeWee can occupy. The characters are:

- `.` (period) — an empty place;
- `P` — a plant;
- `D` — a dragon beetle.

The first character is the starting point of GeeWee journey. It is an empty place.

Constraints

$1 \leq D_l, D_h \leq 1000$	Minimum an maximum distances covered by a jump
$1 \leq X_0, a < 2^{32}$	Pseudo random number
$0 \leq b < 2^{32} \quad 1 \leq m \leq 2^{32}$	generator parameters
$1 \leq E, j, p \leq 1000$	Initial energy, energy consumed by a jump, and plant energy
$10 \leq l \leq 1000000$	Terrain length

Output

The first line should contain the total distance covered by GeeWee (considering both backward and forward jumps), the second line the energy consumed, and the last line should contain the word `yes` if GeeWee reached the land of Plenty alive or `no` otherwise.

Sample Input 1

```
2 4 54123 1013904223 1664525 4294967296 100 3 7 30
.P.D.P.D.D.D.D.P.D.P.D.P...D..
```

Sample Output 1

```
36
33
yes
```

Sample Explanation

GeeWee makes one backward jump, but he is able to reach alive the land of Plenty. The following list of numbers contains the distance covered at time step $n = 1, 2, \ldots$, with a plus signal indicating a forward jump and minus signal representing a backward jump.

```
+2 +3 +3 +4 +4 +4 +4 +3 -3 +2 +4
```

Sample Input 2

```
5 7 795443 1013904223 1664525 4294967296 101 3 8 20
...PPPPPPPDDDDDDD...
```

Sample Output 2

```
303
149
no
```

Sample Explanation

There is a row of dragon beetles larger than the maximum GeeWee jump distance. Therefore he dies from all the backward jumps. Note that the plant that GeeWee eats disappears. The distance covered at time step $n = 1, 2, \ldots$ is:

```
+7 +7 -6 +5 -5 +5 -6 +5 -6 +7 -7 +7 -7 +7 -5 +5 -6 +6 -7 +6 -7 +7 -7 +5 -5 +7 -6
    +6 -7 +6 -5 +6 -5 +5 -7 +5 -7 +7 -7 +7 -7 +6 +5 -7 +7 -7 +7 -6 +5
```

Sample Input 3

```
1 7 2785443 1013904223 1664525 4294967296 35 3 9 30
.DDDDDD.PPPPP..PPPP...P...P.PP
```

Sample Output 3

```
45
44
no
```

Sample Explanation

The last jump of GeeWee takes him into the land of Plenty, unfortunately he spends his last energy in doing so. The distance covered at time step $n = 1, 2, \ldots$ is:

```
+1 -1 +6 -6 +2 -2 +2 -2 +7 +5 +1 +1 +7 +2
```

Sample Input 4

```
1 7 2785443 1013904223 1664525 4294967296 40 3 20 30
.DDDDDD.................P......
```

Sample Output 4

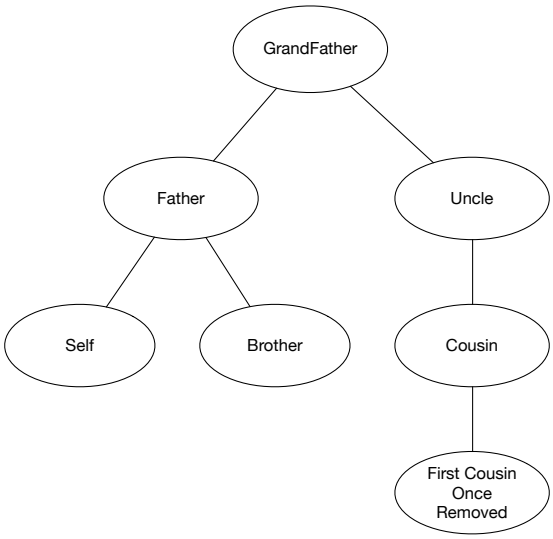
```
43
40
no
```

Sample Explanation

When GeeWee jumps to the single plant, he spends his last energy. Therefore he cannot eat the plant, although it would provide him with enough energy to reach the land of Plenty. Note that he spent 40 units of energy instead of $3 \cdot 14 = 42$ units of energy. When he is about to make the 14^{th} jump he only has one unit of energy left. The distance covered at time step $n = 1, 2, \ldots$ is:

```
+1 -1 +6 -6 +2 -2 +2 -2 +7 +5 +1 +1 +7
```

Problem C: Family Ties



In this close-by country there is a region, called Calci, where men play this very old form of football. Tradition goes back to the dawn of times and involves a fundamental rule saying that all players must descend from a common known ancestor. Unfortunately, in this town, called Ludopedi, the local team has been gradually losing ground to the rival adversary towns. Many think that misfortune comes from the pressure that families do in favor of including their own close members in the team (who could blame them?). The Mayor is very upset with the recent wave of defeats and is going to open an inquiry to determine whether or not nepotism is ruining Ludopedi's performance. Would you be so kind to help?

Task

Your task is to develop a program that receives the family tree starting on some old and distant grandfather and compute the number of players of that family that have another close member of the family in the team, such as a: *i*) brother, *ii*) father (and hence son), *iii*) uncle (nephew), *iv*) cousin, *v*) first cousin once removed (if the other is a son of a cousin), *vi*) grandfather (grandson). The relevant relations to the "self" are shown in the figure. You should count each player only once, even if the player is involved in multiple family relations.

Input

The input starts with the number of members of the family that have or had children in a single line. Then, each one of the following lines has one of the members of the family, with the name and the list of sons. Each of these descendants has a name and a 0 or a 1. A 0 means that such man never played on the team, a 1 means he did. The person starting the family (the oldest ancestor) played in the team. You may assume the following regarding the input:

- Names have no spaces.
- All the names are different.
- The first person in the input is always the oldest ancestor.
- The order of lines after the oldest ancestor is arbitrary.
- The list of descendants of a person is given in a single line.
- All the persons descend from the oldest ancestor (except himself!).
- No person in the input has more than one ancestor.

Constraints

$1 < N \leq 2000$ Number of family members.
$1 \leq L \leq 15$ Length of a name.
$1 \leq D \leq 10$ Number of descendants.

Output

Output a single number, which is the total number of members of the family that have or had close relatives playing in the team.

Sample Input 1

```
4
Mario Paulo 0
Paulo Rui 0 Artur 1
Rui Miguel 1
Artur Luis 0 Andre 0
```

Sample Output 1

```
3
```

Sample Explanation 1

Mario, Artur and Miguel are the only three family members who have played in the team. Artur and Mario have a grandfather/grandson relation; Miguel is Artur's nephew. Note that Mario is the grand-grandfather of Miguel, a relation that doesn't count.

Sample Input 2

```
4
Anibal Cesar 0 Pedro 1
Cesar Antonio 1
Pedro Afonso 0
Afonso Luis 0 Carlos 1
```

Sample Output 2

```
4
```

Sample Explanation 2

The following relations apply:

- Anibal is the grandfather of Antonio.
- Anibal is the father of Pedro.
- Pedro is an uncle of Antonio.
- Pedro is the grandfather of Carlos.
- Antonio is a first cousin once removed of Carlos.

This means that Anibal, Antonio, Pedro and Carlos must be counted.

Problem D: Alibis

When a crime takes place, forensics estimate a time interval $[T_s, T_e]$ and a duration C, in minutes, for the crime. This means that the crime started at or after T_s, ended at or before T_e, and took at least C minutes. Besides, the course of the investigation leads to some suspects. Witnesses, interrogations, surveillance cameras, and many other tools allow the police to know some activities of the suspects, particularly, when and where they took place. Do these activities provide alibis?

Suppose that a robbery took place between 14h00 and 17h30, and lasted for 40 minutes at location L.

- If suspect X has been from 10h00 to 15h00 at a location which is 2 hours away from L, X cannot be the perpetrator: even if X had arrived at 17h00 at the crime location, X could not have completed the robbery by 17h30.
- For suspect Y, four activities are known. One of them was at location L_y, from 15h05 to 18h10. It takes 30 minutes to travel between L and L_y. So, Y also has an alibi: although Y could have been at the crime location at 14h00, Y would have left the crime scene at most at 14h35.
- A third suspect, Z, has been from 12h30 to 14h15 at a location 1 hour away from L, and from 16h15 to 19h30 at another location, which is 20 minutes away from L. Therefore, Z does not have an alibi, as Z can have perpetrated the crime from 15h15 to 15h55.

Task

Write a program that, given the time interval and the duration of the crime, the time periods of the suspects' activities and the corresponding travel times, determines if each of the suspects has an alibi or not.

Input

The input first line has two times, T_s and T_e, and an integer, C. It represents that the crime took place on a given day and in a given place, started at or after T_s, ended at or before T_e, and took at least C minutes. A time is a sequence of the form `HH:MM`, respecting the usual rules ($\texttt{00} \leq \texttt{HH} \leq \texttt{23}$ and $\texttt{00} \leq \texttt{MM} \leq \texttt{59}$).

The second line consists of an integer, S, which is the number of suspects.

For each suspect there is a first line containing an integer, A, which is the number of the suspect's activities, taking place on the day of the crime but at different locations. Each of the following A lines corresponds to an activity. It has two times, T_j and T_k, and an integer, D, specifying that the suspect has been from time T_j to time T_k on the day of the crime at a location that is D minutes away from the crime location.

You may assume that the time interval for the crime has at least C minutes, any time interval for an activity has at least 1 minute (i.e. $T_j < T_k$), and no suspect has been at different locations at the same time.

Constraints

$1 \leq C \leq 600$ Crime duration (in minutes)

$1 \leq S \leq 400$ Number of suspects

$1 \leq A \leq 200$ Number of activities of a suspect

$1 \leq D \leq 720$ Distance between locations (in minutes)

Output

The output consists of S lines. The i^{th} line contains "`yes`", if the i^{th} suspect cannot have been the crime perpetrator (the suspect has an alibi), and "`no`", otherwise.

Sample Input

```
14:00 17:30 40
3
1
10:00 15:00 120
4
15:05 18:10 30
00:00 09:30 1
18:30 23:59 30
11:40 12:00 15
2
16:15 19:30 20
12:30 14:15 60
```

Sample Output

```
yes
yes
no
```

Problem E: Illicit Transactions

Task

Every so often, the police intercepts sequences of transactions between a group of criminals. The transactions consist of exchanging valuable tokens representing various monetary values. The tokens are placed and later retrieved from a secret location. The secret location works in a way that a token can be retrieved only if all the most recent ones were already retrieved. Tokens representing the same value are indistinguishable.

In some cases the police informants supply false information. This means that some sequences of transactions are invalid. Help the police to detect whether the given sequence is a valid sequence or not.

A valid sequence is characterized as follows.

- In a valid sequence a token with value t_1 may *not* be retrieved if there is a later token with value t_2, with $t_1 \neq t_2$, that has not yet been retrieved.
- In a valid sequence all the placed tokens must be eventually retrieved.

Input

The first line of the input is a positive integer m (line counter). The rest of the input consists of m lines, where each line is either a `P`-line or an `R`-line. Each `P`-line contains a positive integer n. A `P`-line represents that a token of value n was **p**laced at the secret location. Each `R`-line contains a negative integer n. An `R`-line represents that a token of value $|n|$ was **r**etrieved from the secret location.

The order of the lines corresponds to the order of how tokens were supposedly placed and retrieved. You can assume that the line counter m is always specified correctly.

Constraints

$1 \leq n \leq 10^4$	Token value
$1 \leq m \leq 10^6$	Number of input lines

Output

- `y` if the input sequence is a valid sequence
- `n` if the input sequence is an invalid sequence

Sample Input 1

```
8
2
2
-2
1
2
-2
-1
-2
```

Sample Output 1

```
y
```

Explanation The sequence is a valid one, where two tokens of value 2 are placed and one is immediately retrieved; then values 1 and 2 are placed, where the token 2 is again immediately retrieved; value 1 is retrieved and finally the last remaining token of value 2 is retrieved.

Sample Input 2

```
4
2
1
-2
-1
```

Sample Output 2

```
n
```

Explanation Produces `n` because 2 cannot be retrieved before 1 in this case.

Sample Input 3

```
3
1
-1
-2
```

Sample Output 3

```
n
```

Explanation Produces `n` because there is no token with value 2 that could be retrieved.

Sample Input 4

```
3
1
2
-2
```

Sample Output 4

```
n
```

Explanation Produces `n` because the token of value 1 remains to be retrieved at the end.

Sample Input 5

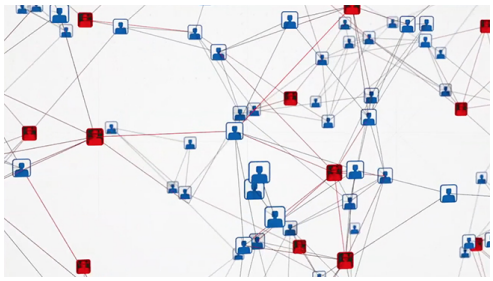
```
3
1
-1
-1
```

Sample Output 4

```
n
```

Explanation Produces `n` because there is only one token of value 1 and cannot therefore be retrieved the second time.

Problem F: Fake Accounts



There are users on a social network who tend to create multiple accounts in order to pretend to be multiple people. However, the rules of this particular social network dictate that a person should have a single account only. Each user account has a *username*, *email*, and *IP address*. Furthermore, *username*s are unique.

Task

Your task is to group those user accounts that might be the same person. In particular, when users have either the same email or IP address, they are likely the same person. As an example, if users John and Jane share IPs but Alice neither shares email nor IP, then your program should report that there exists 2 groups of accounts each representing 1 user.

Input

The input to your application is an integer N followed by N entries, one for each account. Each of the N entries contains the values that define a user, seperated by *one* whitespace. The provided *username*s, *email*s, and *IP address*s strings are no longer than 100 chars, and there is no need to check for their validity (e.g., *username*s are alphanumeric strings, *email*s are strings containing an '@', and *IP address*s are dotted-quad notation strings).

Constraints

$1 \leq N \leq 1000$ Number of accounts

Output

Your program should output the number of different accounts.

Sample Input

```
3
John john@miup.pt 192.128.10.1
Jane jane@miup.fr 192.128.10.1
Alice alice@miup.es 230.20.10.1
```

Sample Output

```
2
```

Problem G: Water Pipeline

Water is a precious and increasingly scarce resource. In a country not far away and in order to implement a country wide water supply policy, the State implemented a National Water Emergency and Safety Network that builds and monitors a water supply system for the whole country.

But, soon after that, problems occurred. One does not simply build and bury water pipelines with no caution.

The supply network was designed keeping in mind the mandatory requirement to optimize costs while ensuring the required service. This means that there is no redundancy in the pipelines between two cities: either there is a single path or there is no path. Hopefully each pair of cities are linked by the buried pipelines.

Hopefully...? but why? Because of rats, moles, fire-starters, anarchists and people who like to savagely build houses ....

As the pipelines are buried in a place, other are destroyed elsewhere. Since the network is huge, monitoring it is a total nightmare.

Task

Your task is to help the people that are building and controlling the water supply network.

The control center receives and sends three types of information:

- the order to build a direct pipeline between two points of interest in the network;
- the information that a pipeline (designated by the two points it connects) was destroyed;
- the query about the existence of a path between two points of interest. This information is very important to the control center to decide/prioritize future pipeline construction.

Input

Let N be the number of interest points in the water supply network (identified from 1 to N) and P the number of orders and queries processed by the control center.

These are given in the following format:

- `ADD a b`
- `DEL a b`
- `LINK a b`

with $a, b \in 1..N$ and, in case of DEL or ADD, $a \neq b$.

The input is then organized as:
The first line introduces N and P (separated by a single space).
The P following lines introduce the orders and queries to be processed.

Output

A sequence of answers ("`YES`" ou "`NO`") of the LINK queries. One line per answer.

The first line answers the first LINK query. The second line of output answers the second LINK query, and so on...

Constraints

$1 \leq N \leq 10\,000$
$1 \leq P \leq 100\,000$

Sample Input

```
5 11
LINK 1 5
ADD 1 2
ADD 1 3
ADD 3 4
ADD 5 4
LINK 1 5
DEL 4 5
LINK 1 5
DEL 3 4
ADD 3 5
LINK 1 5
```

Sample Output

```
NO
YES
NO
YES
```

Problem H: Theseus and the Minotaur

Allegedly, Theseus killed the Minotaur. Also allegedly, Ariadne assisted Theseus by giving him a ball of thread, to help him navigate his way back out of the Labyrinth.

But threads are notoriously unreliable contraptions and, in this day and age, it is possible to use more dependable methods for that purpose. Since the depth at which the labyrinth is located prevents accessing the GPS network, a solution is to use a device that records Theseus' steps on his way to meet the Minotaur, and which will then allow him to retrace them back to the entrance of the labyrinth.

Task

Given the sequence of steps taken by Theseus until he found the Minotaur, your task is to compute the minimum number of steps it will take Theseus to get back to the location from where he started. Note that, having drunk a full keg of hydromel to foster his courage before venturing into the Labyrinth, allegedly, Theseus began by losing the ball of thread and then wandered through the maze haphazardly, rather than systematically, possibly crossing his own path multiple times before coming face to face with the Minotaur.

Each of Theseus' steps is given by its direction, which may be one of North, East, South, and West. The four directions have their usual meaning and the geometry of the space is the normal one: North and South are opposite directions, as are East and West. If it is possible to move from one location to another by taking a step in some direction, then it is possible to move from the latter location to the former by taking a step in the opposite direction. All steps have the same length and if Theseus takes one step North, followed by one step East, one step South, and one step West, he ends up back in the location where he started.

Every step Theseus takes on his way back must correspond to a step he took on his way in, in either the same or the opposite direction.

Input

The input of the program consists of a line with a positive integer M, followed by M lines showing the K steps Theseus took from where he started to where he met the Minotaur, in order. The initial and final locations of his path may be any two locations in the labyrinth. Each step is identified by one of the characters `N`, `E`, `S`, or `W`, corresponding to the initial of the direction in which the step was taken.

All lines of input with Theseus' steps have length L except, possibly, the last line, which may have between 1 and L characters.

Constraints

$1 \leq K \leq 250\,000$	Number of steps
$L = 80$	Maximum number of steps on a line

Output

The output consists of a single line with the minimum number of steps it takes to go back from the location Theseus ended in to the location from where he started.

Sample Input 1

```
1
NNEEEENWNENNSEEEWWWSWSESWWWNNNNWWWSSNNNNEE
```

Sample Output 1

```
14
```

Sample Input 2

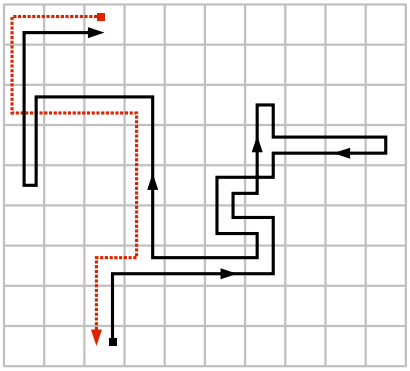
```
1
NEEENNWWWSSWNNEN
```

Sample Output 2

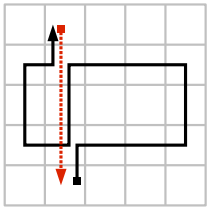
```
4
```

Explanation

Sample 1



Sample 2



Theseus' route.
Shortest path to starting location.

Problem I: Space Invaders

You are responsible for the defense of Algo, the capital city of planet MIUP. The city can be thought of as a long 1D segment with a set of N very tall and thin buildings, housing all the planetary administration. Each building can be identified by three integer values: its position X_i, its height Y_i and its shield generator power P_i (note that the width of all buildings is always one).

You know that the evil military forces of planet Complexity are planning an attack on your city and you need to be ready. The enemy will attack with spaceships coming from the sky that can fire lasers directly downwards towards the city.

You can build horizontal shields, that can be installed on the top of any building. A building i with a shield will protect itself and it will also protect all buildings j that are not higher ($Y_i \geq Y_j$), that are within a P_i horizontal distance ($|X_i - X_j| \leq P_i$) and such that there are no taller buildings between them (for all k between i and j, $Y_i \geq Y_k$).

The shields are however very expensive and you want to build the smallest possible number of shields, to save money, such that all the buildings are still protected.

Task

Given the position, height and power of each building in your capital city, your task is to compute the smallest number of shields that need to be installed such that all buildings are shielded and protected from the enemy spaceships.

Input

The first line contains one integer N, the number of buildings. N lines follow, containing the description of the buildings in your capital city. Each of these lines contains three integers X_i Y_i P_i describing one building, respectively its horizontal position, height and generator power. The buildings can come in any order and no two buildings will have the same horizontal position.

Output

A line with a single integer indicating the smallest possible number of shields that you need to build to protect all the buildings.

Constraints

$1 \leq N \leq 100\,000$	Number of buildings
$1 \leq X_i \leq 10^9$	Horizontal position of buildings
$1 \leq Y_i \leq 10^9$	Height of buildings
$0 \leq P_i \leq 10^9$	Power of buildings

Sample Input 1

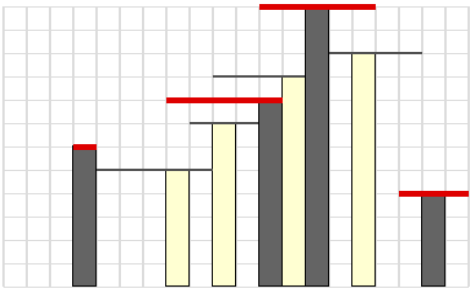
```
8
8 5 20
4 6 0
10 7 1
19 4 1
13 9 3
14 12 2
16 10 2
12 8 4
```

Sample Output 1

```
4
```

Sample Explanation 1

A possible allocation of 4 shields:



Sample Input 2

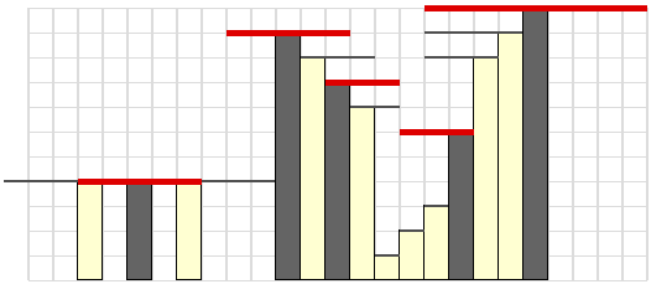
```
14
3 4 3
5 4 2
7 4 3
11 10 2
12 9 2
13 8 2
14 7 1
21 11 4
20 10 3
19 9 2
18 6 2
15 1 0
16 2 0
17 3 0
```

Sample Output 2

```
5
```

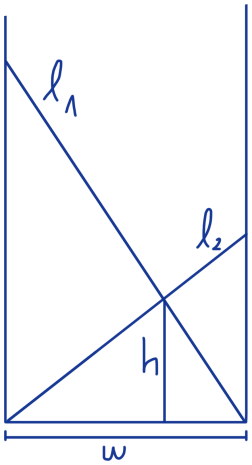
Sample Explanation 2

A possible allocation of 5 shields:



On the explanation figures, the thin black lines indicate the possible shields and the thick red lines indicate selected shields. Moreover, light yellow buildings are the ones protected by others, while darker grey buildings are the ones with shields built on top of it.

Problem J: Guess the Width



A painter takes two ladders, one 3 meters long and the other 2 meters long to paint a corridor and sets them up exactly as can be seen on the drawing. He has a ruler than is exactly 1 meter long and realizes that the intersection between the two ladders is exactly one meter from the ground. What is the width of the corridor?

Task

Please help me solve this problem. Give me the answer with 2 decimal places.

Input

The input consists of one line with the number of cases C. Each subsequent line of input contains three floats with the length of the first ladder l_1, the length of the second ladder l_2 and the height of the intersection of the two ladders h.

Constraints

$1 \leq C \leq 100$	Number of test cases
$0 < l1, l2 \leq 1000$	Length of the ladders
$0 < h < min(l1, l2)$	Height of the intersection of the two ladders

Output

The output should be the width of the corridor with a precision of at least two decimal places.

Sample Input 1

```
3
2 3 1
13 13 6
903.124 785.497 396.522
```

Sample Output 1

```
1.231185
5.0
275.3263
```