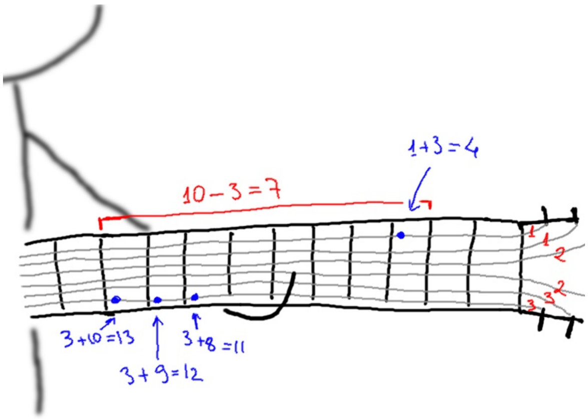
A. Perform Easily

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

After battling Shikamaru, Tayuya decided that her flute is too predictable, and replaced it with a guitar. The guitar has 6 strings and an infinite number of frets numbered from 1. Fretting the fret number j on the i-th string produces the note $a_i + j$.

Tayuya wants to play a melody of n notes. Each note can be played on different string-fret combination. The easiness of performance depends on the difference between the maximal and the minimal indices of used frets. The less this difference is, the easier it is to perform the technique. Please determine the minimal possible difference.

For example, if $a = [1, 1, 2, 2, 3, 3]$, and the sequence of notes is $4, 11, 11, 12, 12, 13, 13$ (corresponding to the second example), we can play the first note on the first string, and all the other notes on the sixth string. Then the maximal fret will be 10, the minimal one will be 3, and the answer is $10 - 3 = 7$, as shown on the picture.



Input

The first line contains 6 space-separated numbers a_1, a_2, ..., a_6 ($1 \leq a_i \leq 10^9$) which describe the Tayuya's strings.

The second line contains the only integer n ($1 \leq n \leq 100\,000$) standing for the number of notes in the melody.

The third line consists of n integers b_1, b_2, ..., b_n ($1 \leq b_i \leq 10^9$), separated by space. They describe the notes to be played. It's guaranteed that $b_i > a_j$ for all $1 \leq i \leq n$ and $1 \leq j \leq 6$, in other words, you can play each note on any string.

Output

Print the minimal possible difference of the maximal and the minimal indices of used frets.

Examples

input

```
1 4 100 10 30 5
6
101 104 105 110 130 200
```

output

```
0
```

input

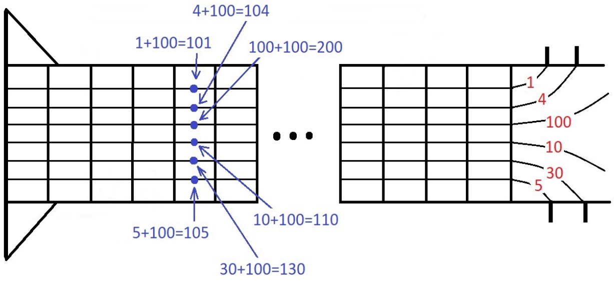
```
1 1 2 2 3 3
7
13 4 11 12 11 13 12
```

output

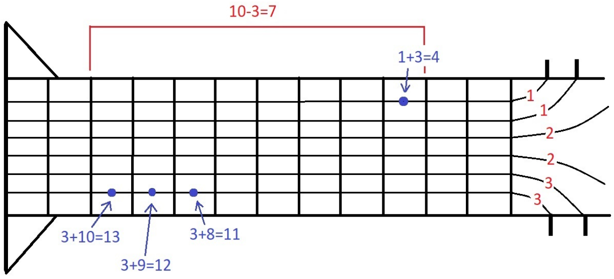
```
7
```

Note

In the first sample test it is optimal to play the first note on the first string, the second note on the second string, the third note on the sixth string, the fourth note on the fourth string, the fifth note on the fifth string, and the sixth note on the third string. In this case the 100-th fret is used each time, so the difference is $100 - 100 = 0$.



In the second test it's optimal, for example, to play the second note on the first string, and all the other notes on the sixth string. Then the maximal fret will be 10, the minimal one will be 3, and the answer is $10 - 3 = 7$.



B. Shurikens

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Tenten runs a weapon shop for ninjas. Today she is willing to sell n shurikens which cost 1, 2, ..., n ryo (local currency). During a day, Tenten will place the shurikens onto the showcase, which is empty at the beginning of the day. Her job is fairly simple: sometimes Tenten places another shuriken (from the available shurikens) on the showcase, and sometimes a ninja comes in and buys a shuriken from the showcase. Since ninjas are thrifty, they always buy the **cheapest** shuriken from the showcase.

Tenten keeps a record for all events, and she ends up with a list of the following types of records:

- + means that she placed another shuriken on the showcase;
- - x means that the shuriken of price x was bought.

Today was a lucky day, and all shurikens were bought. Now Tenten wonders if her list is consistent, and what could be a possible order of placing the shurikens on the showcase. Help her to find this out!

Input

The first line contains the only integer n ($1 \leq n \leq 10^5$) standing for the number of shurikens.

The following $2n$ lines describe the events in the format described above. It's guaranteed that there are exactly n events of the first type, and each price from 1 to n occurs exactly once in the events of the second type.

Output

If the list is consistent, print "YES". Otherwise (that is, if the list is contradictory and there is no valid order of shurikens placement), print "NO".

In the first case the second line must contain n space-separated integers denoting the prices of shurikens in order they were placed. If there are multiple answers, print any.

Examples

input

```
4
+
+
- 2
+
- 3
+
- 1
- 4
```

output

```
YES
4 2 3 1
```

input

```
1
- 1
+
```

output

```
NO
```

input

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

output

```
NO
```

Note

In the first example Tenten first placed shurikens with prices 4 and 2. After this a customer came in and bought the cheapest shuriken which costed 2. Next, Tenten added a shuriken with price 3 on the showcase to the already placed 4-ryo. Then a new customer bought this 3-ryo shuriken. After this she added a 1-ryo shuriken. Finally, the last two customers bought shurikens 1 and 4, respectively. Note that the order $[2, 4, 3, 1]$ is also valid.

In the second example the first customer bought a shuriken before anything was placed, which is clearly impossible.

In the third example Tenten put all her shurikens onto the showcase, after which a customer came in and bought a shuriken with price 2. This is impossible since the shuriken was not the cheapest, we know that the 1-ryo shuriken was also there.

C. Solo mid Oracle

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Meka-Naruto plays a computer game. His character has the following ability: given an enemy hero, deal a instant damage to him, and then heal that enemy b health points at the end of every second, for exactly c seconds, starting one second after the ability is used. That means that if the ability is used at time t, the enemy's health decreases by a at time t, and then increases by b at time points $t+1$, $t+2$, ..., $t+c$ due to this ability.

The ability has a cooldown of d seconds, i. e. if Meka-Naruto uses it at time moment t, next time he can use it is the time $t+d$. Please note that he can only use the ability at integer points in time, so all changes to the enemy's health also occur at integer times only.

The effects from different uses of the ability may stack with each other; that is, the enemy which is currently under k spells gets $k \cdot b$ amount of heal this time. Also, if several health changes occur at the same moment, they are all counted at once.

Now Meka-Naruto wonders if he can kill the enemy by just using the ability each time he can (that is, every d seconds). The enemy is killed if their health points become 0 or less. Assume that the enemy's health is not affected in any way other than by Meka-Naruto's character ability. What is the maximal number of health points the enemy can have so that Meka-Naruto is able to kill them?

Input

The first line contains an integer t ($1 \le t \le 10^5$) standing for the number of testcases.

Each test case is described with one line containing four numbers a, b, c and d ($1 \le a, b, c, d \le 10^6$) denoting the amount of instant damage, the amount of heal per second, the number of heals and the ability cooldown, respectively.

Output

For each testcase in a separate line print -1 if the skill can kill an enemy hero with an arbitrary number of health points, otherwise print the maximal number of health points of the enemy that can be killed.

Example

input

```
7
1 1 1 1
2 2 2 2
1 2 3 4
4 3 2 1
228 21 11 3
239 21 11 3
1000000 1 1000000 1
```

output

```
1
```

```
2
1
5
534
-1
500000500000
```

Note

In the first test case of the example each unit of damage is cancelled in a second, so Meka-Naruto cannot deal more than 1 damage.

In the fourth test case of the example the enemy gets:

- 4 damage (1-st spell cast) at time 0;
- 4 damage (2-nd spell cast) and 3 heal (1-st spell cast) at time 1 (the total of 5 damage to the initial health);
- 4 damage (3-nd spell cast) and 6 heal (1-st and 2-nd spell casts) at time 2 (the total of 3 damage to the initial health);
- and so on.

One can prove that there is no time where the enemy gets the total of 6 damage or more, so the answer is 5. Please note how the health is recalculated: for example, 8-health enemy would **not** die at time 1, as if we first subtracted 4 damage from his health and then considered him dead, before adding 3 heal.

In the sixth test case an arbitrarily healthy enemy can be killed in a sufficient amount of time.

In the seventh test case the answer does not fit into a 32-bit integer type.

D. Roads and Ramen

time limit per test: 5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

In the Land of Fire there are n villages and $n-1$ bidirectional road, and there is a path between any pair of villages by roads. There are only two types of roads: stone ones and sand ones. Since the Land of Fire is constantly renovating, every morning workers choose a single road and flip its type (so it becomes a stone road if it was a sand road and vice versa). Also everyone here loves ramen, that's why every morning a ramen pavilion is set in the middle of every **stone** road, and at the end of each day all the pavilions are removed.

For each of the following m days, after another road is flipped, Naruto and Jiraiya choose a simple path — that is, a route which starts in a village and ends in a (possibly, the same) village, and doesn't contain any road twice. Since Naruto and Jiraiya also love ramen very much, they buy a single cup of ramen on each stone road and one of them eats it. Since they don't want to offend each other, they only choose routes where they can eat equal number of ramen cups. Since they both like traveling, they choose any longest possible path. After every renovation find the maximal possible length of a path (that is, the number of roads in it) they can follow.

Input

The first line contains the only positive integer n ($2 \leq n \leq 500\,000$) standing for the number of villages in the Land of Fire.

Each of the following $(n-1)$ lines contains a description of another road, represented as three positive integers u, v and t ($1 \leq u, v \leq n$, $t \in \{0, 1\}$). The first two numbers denote the villages connected by the road, and the third denotes the initial type of the road: 0 for the sand one and 1 for the stone one. Roads are numbered from 1 to $(n-1)$ in the order from the input.

The following line contains a positive integer m ($1 \leq m \leq 500\,000$) standing for the number of days Naruto and Jiraiya travel for.

Each of the following m lines contains the single integer id ($1 \leq id \leq n-1$) standing for the index of the road whose type is flipped on the morning of corresponding day.

It is guaranteed that there is a road path between any pair of villages.

Output

Output m lines. In the i-th of them print the only integer denoting the maximal possible length of any valid path on the i-th day.

Example

input

```
5
1 2 0
1 3 0
3 5 0
3 4 0
5
3
4
1
3
4
```

output

```
3
2
3
```

```
3
2
```

Note

After the renovation of the 3-rd road the longest path consists of the roads 1, 2 and 4.

After the renovation of the 4-th road one of the longest paths consists of the roads 1 and 2.

After the renovation of the 1-st road one of the longest paths consists of the roads 1, 2 and 3.

After the renovation of the 3-rd road the longest path consists of the roads 1, 2 and 4.

After the renovation of the 4-rd road one of the longest paths consists of the roads 2 and 4.

E. A Convex Game

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Shikamaru and Asuma like to play different games, and sometimes they play the following: given an increasing list of numbers, they take turns to move. Each move consists of picking a number from the list.

Assume the picked numbers are v_{i_1}, v_{i_2}, $\ldots$, v_{i_k}. The following conditions must hold:

- $i_j < i_{j+1}$ for all $1 \leq j \leq k-1$;
- $v_{i_{j+1}} - v_{i_j} < v_{i_{j+2}} - v_{i_{j+1}}$ for all $1 \leq j \leq k-2$.

However, it's easy to play only one instance of game, so today Shikamaru and Asuma decided to play n simultaneous games. They agreed on taking turns as for just one game, **Shikamaru goes first**. At each turn, the player performs a valid move in any single game. The player who cannot move loses. Find out who wins, provided that both play optimally.

Input

The first line contains the only integer n ($1 \leq n \leq 1000$) standing for the number of games Shikamaru and Asuma play at once. Next lines describe the games.

Each description starts from a line with the only number m ($m \geq 1$) denoting the length of the number list. The second line contains the increasing space-separated sequence v_1, v_2, ..., v_m from the game ($1 \leq v_1 < v_2 < \ldots < v_m \leq 10^5$).

The total length of all sequences doesn't exceed 10^5.

Output

Print "YES" if Shikamaru can secure the victory, and "NO" otherwise.

Examples

input

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

output

```
YES
```

input

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

output

```
NO
```

input

```
4
7
14404 32906 41661 47694 51605 75933 80826
5
25374 42550 60164 62649 86273
2
7002 36731
8
23305 45601 46404 47346 47675 58125 74092 87225
```

output

```
NO
```

Note

In the first example Shikamaru can pick the last number, and Asuma cannot do anything because of the first constraint.

In the second sample test Asuma can follow the symmetric strategy, repeating Shikamaru's moves in the other instance each time, and therefore win.