Problem A
Adventure Begins

The game Pokenom Go has just been released. Pokenom trainers can now travel the world, capture Pokenom in the wild and battle each other! Bash — the Pokenom trainer — has decided to drop out of his university to pursue his childhood dream of becoming the best Pokenom trainer!

However, Linux — Bash's university headmaster — does not allow his students to drop out so easily ...

Linux puts N black boxes on a straight line. The black boxes are numbered from 1 to N from left to right. Initially, all black boxes are empty. Then Linux gives Bash Q queries. Each query can be one of the following 2 types:

- Linux puts exactly one stone inside exactly one box between u-th box and v-th box, inclusive, with equal probability. $(1 \le u \le v \le N)$.
- Let a_i be the number of stones in black box numbered i. Let $A = \sum_{i=1}^{N} a_i^2$. Bash has to calculate the expected value $E(A)$.

Bash can only drop out of his university if he is able to answer all queries correctly. But now all Bash can think of is Pokenom. Please help him!

Input

The first line of input contains exactly 2 positive integers N and Q. $(1 \le N, Q \le 10^5)$.

Q lines follow, each line contains exactly one query. As explained, a query can be one of the following 2 types:

- $1\ u\ v$: Linux puts a stone inside one of the boxes between u and v.
- 2: Linux asks Bash to compute $E(A)$.

Output

It can be proved that the expected value can be represented as an irreducible fraction $\frac{P}{Q}$. For each query of type 2, print one line containing the value $P \times Q^{-1}$ modulo $10^9 + 7$. The given input guarantees that Q is not a multiple of $10^9 + 7$.

Explanation for examples

- In the first example: With a probability of 0.5, two stones are in different squares. Hence, the answer to the forth query is $0.5 \times (1^2 + 1^2) + 0.5 \times 2^2 = 3$.
- In the second example: With a probability of $\frac{2}{3}$, two stones are in different squares. Hence, the answer to the forth query is $\frac{2}{3} \times 2 + \frac{1}{3} \times 4 = \frac{8}{3}$.

Sample Input 1

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

Sample Output 1

```
1
3
```

Sample Input 2

```
3 4
1 1 3
2
1 1 3
2
```

Sample Output 2

```
1
666666674
```

Problem B
Battle of Pokenom

In order to become the very best Pokenom trainer, Bash - the Pokenom trainer - must first understand the Pokenom battle rules.

A Pokenom battle between 2 Pokenom trainers A and B goes as follow: First, each Pokenom trainer must choose one Pokenom. For simplicity, let's assume trainer A chooses Pokenom a and trainer B chooses Pokenom b. The battle consists of exactly K rounds. In each round, there are exactly 2 steps:

- Step 1: Pokenom a attacks Pokenom b. The attack can be effective or not effective.
- Step 2: Pokenom b attacks Pokenom a. The attack can be effective or not effective.

If an attack is effective, the corresponding trainer will receive exactly 1 point. Otherwise, the trainer does not receive any point. After all K rounds, the trainer with more points wins. If 2 trainers have same number of points, the battle is a draw.

Bash watched several Pokenom battles between 2 trainers A and B. Bash wrote down the result of all steps (there are exactly $2 \cdot K$ steps in total).

After carefully reviewing the result, Bash noticed something extraordinary: It is possible to know the outcome of the battle before the last step! More formally, for each battle, there exist a smallest step $S \leq 2 \cdot K$, such that no matter what happens from step $S + 1$ to step $2 \cdot K$, we know for sure which trainer wins the battle, or if the battle is a draw.

For example, consider the following battle with $K = 5$: (E for 'effective' and N for 'not effective'):

- Round 1: 'E E'
- Round 2: 'E N'
- Round 3: 'E E'
- Round 4: 'E E'
- Round 5: 'E N'

After $S = 9$ steps (i.e. after the first step of the 5-th round), A has $P_A = 5$ points and B has $P_B = 3$ points. No matter what happens in the 10-th step, we know for sure that B wins.

Excited, Bash immediately tells his friend Cee 3 numbers: S - the **earliest** step after which Bash know for sure which Pokenom trainer wins the battle, P_A - how many points Pokenom trainer A has after S steps, and P_B - how many points Pokenom trainer B has after S steps.

After that, Bash tells Cee the result of each step from 1 to S. However, after only C steps, Cee stops Bash and tell him: 'now I know the result of all steps from $C + 1$ to S'.

For example, in the above battle, Bash tells Cee $S = 9$, $P_A = 5$ and $P_B = 3$. Bash then tells Cee the result of each step. After $C = 4$ steps, here are the result given to Cee ('?' indicates that the result is not given):

- Round 1: 'E E'
- Round 2: 'E N'
- Round 3: '? ?'
- Round 4: '? ?'
- Round 5: '? ?'

Because Cee also knows that after $S = 9$ steps, A has $P_A = 5$ points and $P_B = 3$ points, he can deduce that A gains 3 more points in step 5, 7 and 9, and B gains 2 more points in step 6, 8.

As Cee is very smart, he always tells Bash as soon as he can infer the result of all step from 1 to S. In other words, C should be as small as possible. Also, remember that Cee knows the meaning of the number S: the winner (if any) is surely determined after S steps, but not after $S - 1$ steps.

Given the result of all $2 \times K$ steps, find the number S and C.

Input

The first line of input contains exactly one positive integer T - the number of test cases $(1 \leq T \leq 65)$.

T test cases follow, each test case consists of:

- First line contains exactly one positive integer K $(1 \leq K \leq 6)$.
- In the next K lines, the i-th line contains 2 space-separated characters, representing the result of the 2 steps in the i-th round.

Output

For each test case, print exactly one line containing 2 integers S and C.

Sample Input 1

```
2
5
E E
E N
E E
E E
E N
3
N E
N E
E N
```

Sample Output 1

```
9 4
4 0
```

Problem C
Communicating The Strategy

This is an interactive problem.

During a Pokenom battle, a Pokenom trainer must come up with a strategy and communicate the strategy to his Pokenom in a very efficient way.

To become the best Pokenom trainer, Bash is practicing how to communicate the strategy to his Pokenom, Chikapu.

More formally, Bash's strategy is a sequence of integers of length n $(1 \le n \le 100)$. Bash needs to communicate this strategy to Chikapu. Let denote this sequence as $A_1, A_2, \ldots, A_n$ $(1 \le A_i \le 10^9)$.

First, Bash gives Chikapu the integer n. Then Chikapu can ask Bash multiple questions about the sequence. Bash answers each question immediately, before Chikapu asks another question. Each time Chikapu gives Bash 2 indices l and r $(1 \le l \le r \le n)$. Bash then selects an arbitrary integer k $(1 \le k \le r - l + 1)$, considers all subsets of size k of the sequence $a_l, a_{l+1}, \ldots, a_r$ (There are $\binom{r-l+1}{k}$ such subsets). For each subset, Bash calculates the product of all elements. Finally, Bash gives Chikapu the number $S \mod (10^9 + 7)$, where S is the sum of all products; as well as the number k he has chosen.

Note that Bash can select any values of k as he wants, Chikapu only knows which numbers he chooses, Chikapu does not know how and why he chooses them.

Because the communication has to be as efficient as possible, Bash considers the weight of a (l, r) query equal $\frac{1}{(r-l+1)^2}$. Chikapu must find the sequence after asking queries with total weight of at most $\frac{5}{3}$.

Chikapu can run fast and shoot thunderbolt, but Chikapu is not good with numbers. Please help Chikapu!

Interaction

First, your program reads the integer n $(1 \le n \le 100)$ from the standard input. Then the following process repeats:

- Your program writes to the standard output in either format:
  - $1\ l\ r$, meaning that you want to ask a query with two indices l and r $(1 \le l \le r \le n)$.
  - $2\ A_1 A_2 \ldots A_n$, meaning that you want to guess the sequence $(1 \le A_i \le 10^9)$.
- If your program asks a query, two integers k S will be available in the standard input. Your program then reads them.
- If your program guesses the sequence, your answer will be checked. **Your program should terminate immediately after this.**

The input guarantees that S is not a multiple of $10^9 + 7$ for all possible k, l, r.

Communication example

Your output (standard output)	Kattis' answer (standard input)	Interpretation
	3	The array has 3 elements
1 2 3		Your program asks for information about sequence $[A_2, A_3]$.
	2 10	Kattis gives you information of subsets of size 2. There is one subset of size 2: $\{A_2, A_3\}$. The product of its elements and 10 are congruent modulo $10^9 + 7$.
1 1 2		Your program asks for information about sequence $[A_1, A_2]$.
	2 15	Kattis gives you information of subsets of size 2. There is one subset of size 2: $\{A_1, A_2\}$. The product of its elements and 15 are congruent modulo $10^9 + 7$.
2 3 5 2		You guess the sequence is [3, 5, 2].

Note

When you write the solution for the interactive problem it is important to keep in mind that if you output some data it is possible that this data is first placed to some internal buffer and may be not directly transferred to the interactor. In order to avoid such situation **you have to use special 'flush' operation each time you output some data**. There are these 'flush' operations in standard libraries of almost all languages. For example, in C++ you may use 'fflush(stdout)' or 'cout « flush' (it depends on what do you use for output data — 'scanf/printf' or 'cout'). In Java you can use method 'flush' for output stream, for example, 'System.out.flush()'. In Python you can use 'stdout.flush()'.

Problem D
Dreamer

Bash just woke up from his sweetest dream ever. In his dream, he became the best Pokenom trainer — like no one ever was. It was on the date ...

Unfortunately, Bash forgot the exact date. He only remembered that the date was written in format 'DD MM YYYY' with exactly 8 digits. He also remember these 8 digits (but he does not remember their order). Of course, the date must be a valid date.

Bash really wants to know the date when he become the best Pokenom trainer. How many possible valid dates could there be? What is the earliest valid date that Bash could become the best Pokenom trainer?

Notes

On leap year, February has 29 days. Following is the rule for leap year:

- A year divisible by 400 is a leap year,
- A year divisible by 100 but not by 400 is NOT a leap year,
- A year divisible by 4 but not by 100 is a leap year,
- A year not divisible by 4 is NOT a leap year.

Input

- The first line contains one integer t $(1 \leq t \leq 50)$ — the number of test cases.
- Each of the next t lines describes one test case, contains eight digits in the format 'DD MM YYYY' (eight digits, separated by two blank spaces).

Output

For each test case, output a single line, the number of possible dates, and the earliest date which Bash could become the best Pokenom trainer, in the format 'DD MM YYYY'. If there is no valid dates, print a single line '0' instead.

Sample Input 1	Sample Output 1
3 04 11 2018 23 45 6789 01 01 0001	1084 28 11 0014 0 16 01 01 0001

Problem E
Evolutions

In order to become the very best Pokenom trainer, Bash is studying Pokenom's evolutions.

Each Pokenom has a combat power (CP), indicating how strong the Pokenom is. After certain amount of training, a Pokenom can evolve, and the evolved Pokenom will have higher CP. A Pokenom can evolve multiple times. There is no known limit on how many times a Pokenom can evolve.

CP0 = 4 CP1 = 6 CP2 = 9

When a Pokenom evolves, his CP always increases by a constant ratio.

More formally, let's CP_i denotes the CP of the Pokenom after it evolved i times. If the Pokenom evolves k times, then the following conditions must be true:

- $CP_0 < CP_1$ or $k < 1$,
- $\frac{CP_1}{CP_0} = \frac{CP_2}{CP_1} = \cdots = \frac{CP_k}{CP_{k-1}}$,
- CP_i is a positive integer.

A sequence is called CP-sequence if it satisfies the above conditions. For example:

- $1, 2, 4, 8, 16$ is a CP-sequence.
- $4, 6, 9$ is a CP-sequence.
- $4, 2, 1$ is NOT a CP-sequence, because $4 > 2$.
- $4, 6, 9, 13.5$ is NOT a CP-sequence, because 13.5 is not an integer.
- $4, 6, 9, 13$ is NOT a CP-sequence, because $\frac{13}{9} \neq \frac{9}{6}$.

Bash is very excited to learn about CP-sequence. Given an integer S, he wants to know how many CP-sequences whose sums equal S.

For example, when $S = 7$, there are 5 CP-sequences: $(7), (1, 6), (2, 5), (3, 4), (1, 2, 4)$. When $S = 19$, there are 11 sequences: $(19), (1, 18), (2, 17), \ldots, (9, 10), (4, 6, 9)$.

Input

- The first line contains one integer t ($1 \le t \le 1\,000$).
- The second line contains t distinct integers $S_1, S_2, \ldots, S_t$ ($1 \le S_i \le 10^6\ \forall 1 \le i \le t$).

Output

Print t integers in one line, the i-th number should be the answer to the problem when $S = S_i$.

Sample Input 1	Sample Output 1
7 3 5 7 11 13 17 19	2 3 5 6 8 9 11

Problem F
Forest of Celery

Celery — the legendary Pokenom has been spotted in Alexa Forest.

To become the best Pokenom trainer, Bash has arrived at Alexa Forest to capture Celery. After lots of information gathering, Bash was able to draw a map of Alexa Forest, and noted down K sightings of Celery.

Alexa Forest's map is a convex polygon A with N vertices on the Cartesian plane. K sightings of Celery can be considered as K points — all are strictly inside Alexa Forest.

Bash is ready to search Alexa Forest to find Celery. However, Bash realized that Alexa Forest is simply too big. It would take decades to search the entire forest. But Bash is smart. Based on his research, Bash knows that Celery can only be found inside a polygon Z, where vertices of Z is a subset of A, and all K sightings of Celery must be strictly inside polygon Z.

Of course, there can be multiple polygons Z satisfying the above conditions. Your task is to help Bash find the polygon Z with smallest number of vertices.

Note

A point P is strictly inside Polygon A, iff P is inside A and P does not lie on the border of A.

Input

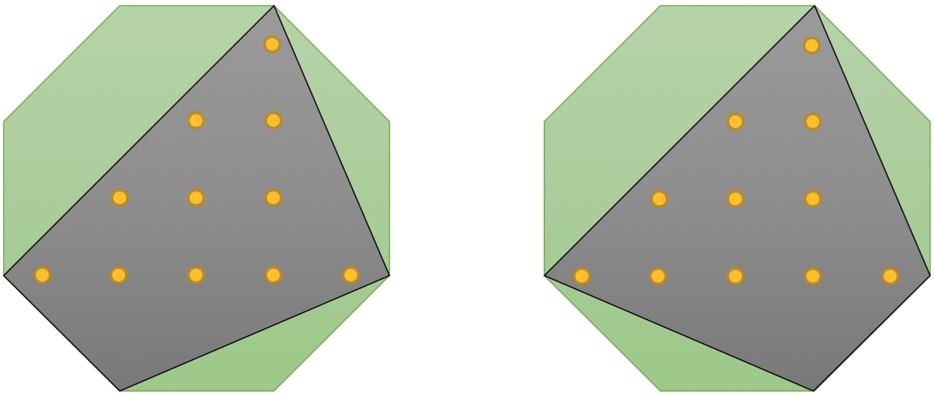
- The first line of input contains a single positive integer N ($3 \le N \le 2 \times 10^5$).
- The next N lines, each line contains 2 integers x_i, y_i — the coordinates of the i-th vertex of Alexa Forest ($-10^9 \le x_i, y_i \le 10^9$). The vertices are listed in either clockwise or counterclockwise order. It is guaranteed that Alexa Forest is convex.
- The next line contains a single positive integer K ($1 \le K \le 10^5$).
- The next K lines, each line contains 2 integers x_i, y_i — the coordinates of a sighting of Celery ($-10^9 \le x_i, y_i \le 10^9$). All points are guaranteed to be inside Alexa Forest and no points are on the border of Alexa Forest.

Output

Output a single integer — the smallest number of vertices of polygon Z.

Sample Clarification

- In the first example, the only valid polygon satisfied is the whole Alexa Forest.
- In the second example, there are two possible solutions with 4 vertices:



Sample Input 1

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

Sample Output 1

```
4
```

Sample Input 2

```
8
3 0
7 0
10 3
10 7
7 10
3 10
0 7
0 3
11
1 3
3 3
5 3
7 3
9 3
3 5
5 5
7 5
5 7
7 7
7 9
```

Sample Output 2

```
4
```

Problem G
Gotta Catch Em All!

The Kanto region has N junctions and $N-1$ bidirectional roads. Junctions are numbered from 1 to N, inclusive. All roads have the same length and each of them connects two different junctions.

At any moment, a Pokenom can appear in any of these junctions.

To become the best Pokenom trainer, like no one ever was, Bash is studying the behavior of the Pokenom in Kanto region. Bash has found that, when trying to capture a Pokenom at junction u, the Pokenom can run away to junction v, iff the optimal path between u and v has length exactly 2.

More formally, a path of length K from junction s to junction t is an ordered sequence of junctions $v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_K$, where $v_0 = s, v_K = t$ and for each valid index i, v_i and v_{i+1} are connected directly by some road. A path is called optimal iff there is no shorter path with the same starting and ending junctions. Two paths $v_0 \rightarrow v_1 \rightarrow \ldots \rightarrow v_k$ and $w_0 \rightarrow w_1 \rightarrow \ldots w_l$ are different iff either $k \neq l$ or there exist some valid i such that $v_i \neq w_i$.

A Pokenom can use an optimal path of length exactly 2. Help Bash count the number of such paths.

Input

- The first line contains one integer N $(1 \leq N \leq 3 \times 10^5)$ — the number of junctions.
- Each of the rest $N-1$ lines contains two integers u and v $(1 \leq u, v \leq N, u \neq v)$ — two endpoints of a single road.

Output

- The only line contains exactly one integer — the number of optimal paths of length 2.

Explanation for the first example

- There are two optimal paths of length 2: $(2 \rightarrow 1 \rightarrow 3)$ and $(3 \rightarrow 1 \rightarrow 2)$.
- The path $(1 \rightarrow 2 \rightarrow 1)$ is a valid path of length 2 from 1 to 1 but it is not optimal since there is a path of length 0 from 1 to 1.

Sample Input 1	Sample Output 1
3 1 2 1 3	2

Sample Input 2	Sample Output 2
5 2 1 1 5 3 1 4 3	8

Sample Input 3	Sample Output 3
10 1 2 2 3 2 4 1 5 3 6 2 7 7 8 5 9 5 10	24

Problem H
How to Paint?

Being a Pokenom trainer is very stressful. You always need to take good care of all your Pokenoms and always be prepared to battle other Pokenom trainers.

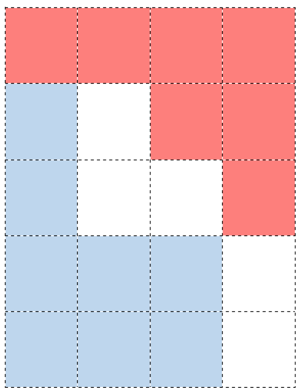
To cope with the stressful life, Bash learns to paint in his free time. Today Bash is working on his masterpiece: 'The 2 staircases'.

The painting is divided into a grid with M rows and N columns. Rows and columns are numbered starting from 1, from bottom to top and from left to right, respectively. Let (i, j) denote the cell at the i-th row and j-th column.

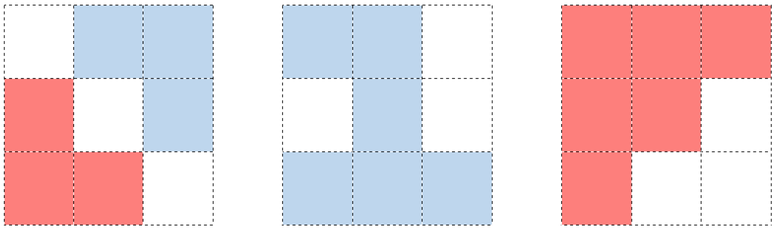
Bash colors each cell red or blue, such that the bottom part forms a blue 'staircase' and the top part forms a red 'inverted staircase'. More formally, Bash's picture has the following properties:

- For every column i $(1 \le i \le N)$, there exists two integers b_i, r_i satisfying:
  - $0 \le b_i, 0 \le r_i, b_i + r_i \le M$.
  - b_i bottommost cells (i.e, cells $(1, i), (2, i), \ldots, (b_i, i)$) are blue.
  - r_i topmost cells (i.e, cells $(M, i), (M-1, i), \ldots, (M - r_i + 1, i)$) are red.
  - All other cells are not painted.
- $M \ge b_1 \ge b_2 \ge \ldots \ge b_N \ge 0$.
- $0 \le r_1 \le r_2 \le \ldots \le r_N \le M$.

Hence, Bash's picture can be uniquely determined by two sequences $b = (b_1, b_2, \ldots, b_N)$ and $r = (r_1, r_2, \ldots r_N)$. This is an example of a valid picture with $M = 5$, $N = 4$, $b = (4, 2, 2, 0)$ and $r = (1, 1, 2, 3)$:



Below are three examples of invalid pictures:



After few hours of hard work, Bash has finished his painting, and shows his best friend Cee. The picture satisfies all the above properties, with parameter $b = (c_1, c_2, \ldots, c_N)$ and $r = (M - c_1, M - c_2, \ldots M - c_N)$. No cells are left unpainted in this picture.

Cee wants to know step-by-step of how Bash created such beautiful painting. Bash can not remember the order which he painted the cells, but Bash remembers that he always followed these rules:

- Bash starts with an empty picture.
- First, Bash paints the bottom-left cell $(1, 1)$ blue and the top-right (M, N) cell red.
- In each step, Bash chooses some unpainted cell, paints it either red or blue, such that the picture after this step satisfies all the above properties.
- The process stops when the picture is completed.

Cee tries to follow Bash's rules to replicate the painting. But first Cee wants to know how many ways Cee can create the painting. Two ways are considered different if there exists a step where the painted cells are different.

Represent the result as $100\,003^X \times Y$, where Y is not divisible by $100\,003$, and output X Y_m where Y_m is the result of $Y \bmod 100\,003$.

Input

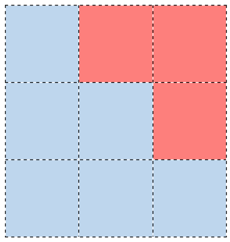
- Line 1: Two integers N and M $(1 \le M, N \le 1\,000)$.
- Line 2: N integers $c_1, c_2, \cdots, c_N$ $(M \ge c_1 \ge c_2 \ge \cdots \ge c_n \ge 0)$ describes the blue parameters in Bash's final picture (see description above).

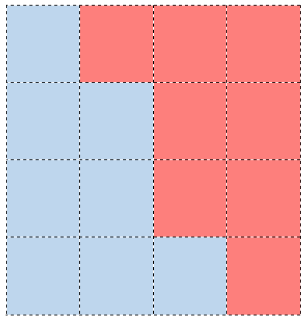
Output

Two integers X and Y_m as described above.

Sample clarification

Bash's pictures in 2 sample cases:





Sample Input 1

```
3 3
3 2 1
```

Sample Output 1

```
0 672
```

Sample Input 2

```
4 4
4 3 1 0
```

Sample Output 2

```
0 16296
```

Problem I
I Wanna Be The Very Best

In order to become the very best Pokenom trainer, Bash needs to prepare a team of Pokenom to participate in the Pokenom world championship.

Bash has N Pokenoms. Each Pokenom has 3 stats: Attack, Defense and Health. Bash wants to select K Pokenoms with highest attack, K Pokenoms with highest defense and K Pokenoms with highest Health.

After selection, Bash found something strange: his team have less than $3 \times K$ Pokenoms!

Bash looks carefully at $N = 4$ Pokenoms he has:

- 'Chikapu': Attack = 100, Defense = 100, Health = 100
- 'Batterfly': Attack = 10, Defense = 10, Health = 10
- 'Mewthree': Attack = 200, Defense = 200, Health = 80
- 'Dragonon': Attack = 150, Defense = 150, Health = 90

When Bash selects Pokenom with $K = 1$, only 'Mewthree' and 'Chikapu' are selected! This is because 'Mewthree' has highest attack and highest defense!

Your task is simple, you are given the stats of all N Pokenom and the number K. Calculate how many different Pokenom are there in Bash's team.

Input

- The first line of input contains 2 integers N and K ($1 \le K \le N \le 1\,000$).
- In the next N lines, the i-th line contains 3 integers: A_i, D_i and H_i, representing the 3 stats of the i-th Pokenom. A_i, D_i and H_i are unsigned 32-bit integers.

It is guaranteed that no 2 Pokenom have same Attack, no 2 Pokenom have same Defense, and no 2 Pokenom have same Health.

Output

Output one line containing exactly one integer: the number of Pokenom in Bash's team.

Sample Input 1	Sample Output 1
4 1 100 100 100 10 10 10 200 200 80 150 150 90	2

Problem J
Joyless Game

Playing game is the best way to improve flexibility, critical thinking and strategy.
To become the best Pokenom player, Bash is playing some games with his Pokenom Chikapu.

- Bash writes down a string S containing only lowercase letters. No 2 consecutive characters in S are equal.
- Bash and Chikapu alternatively take turn to play.
- In each turn, a player must delete one character in S. There are 2 conditions:
  - The first and last character can not be deleted.
  - After the character is deleted, in the new string, no 2 consecutive characters are equal.
- The player who can not delete a character loses.
- Chikapu plays first.

After playing $10^9 + 7$ games, Chikapu won 0 games and lost all $10^9 + 7$ times. Chikapu thinks that Bash is cheating, by selecting a string S such that Bash always wins.

Given some string S, can you help determine who would win the game, if they both play optimally?

Input

The first line of input contains the only integer T — the number of test cases $(1 \leq T \leq 20)$.
The next T lines, each line contains exactly one string S $(3 \leq |S| \leq 10^5)$.

Output

For each test case, print on one line the name of the winner, if they both play optimally. Please note that this problem uses case-sensitive checker.

Sample Input 1	Sample Output 1
2 vietnam icpc	Chikapu Bash

Problem K
Keep Them Separated

Team Socket was defeated by Bash — the Pokenom trainer — and Bash's best Pokenom Chikapu for the $10^9 + 7$-th time. Team Socket realized that, Bash and Chikapu were simply too strong together. Now team Socket is devising an evil plan to keep Bash and Chikapu separated! Team Socket has built an evil machine, which can instantly build a rectangular wall or instantly remove a rectangular wall.

Given the locations where Team Socket is going to build and remove walls, can you help Team Socket check whether Bash and Chikapu are separated?

You are given Q queries, numbered from 1 to Q. The i-th query can be one of the following 3 types:

- $1\ x_1\ y_1\ x_2\ y_2$: Team Socket build a rectangular wall, with sides parallel to the axes, and 2 opposite corners at (x_1, y_1) and (x_2, y_2) $(x_1 \neq x_2, y_1 \neq y_2)$.
- $2\ j$: Team Socket remove the rectangular wall built in j-th query. It is guaranteed that j-th query is of 1st type, the wall was built before this query (i.e. $j < i$), and the wall was not removed previously.
- $3\ x_1\ y_1\ x_2\ y_2$: Bash is standing at (x_1, y_1), and Chikapu is standing at (x_2, y_2). Please let Team Socket know if there is a path from Bash to Chikapu. Of course, both Bash and Chikapu cannot walk through any walls.

Input

The first line of input contains exactly one integer Q — the number of queries $(1 \leq Q \leq 10^5)$.

Then Q lines follow, the i-th line is one of 3 types:

- $1\ x_1\ y_1\ x_2\ y_2$
- $2\ j$
- $3\ x_1\ y_1\ x_2\ y_2$

All coordinates in the input file are integers from 1 to $5\,000$, inclusive. It is guaranteed that:

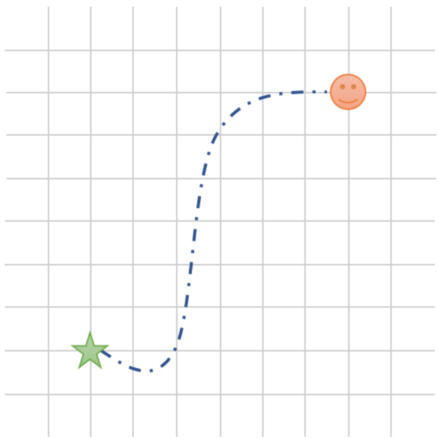
- After each query, no 2 walls have a common point.
- In all queries of 1st type, x_1, y_1, x_2, y_2 are odd numbers.
- In all queries of 3rd type, x_1, y_1, x_2, y_2 are even numbers.

Output

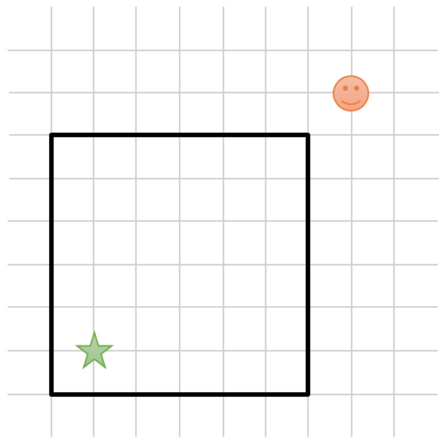
For each query of third type, print a character 'Y' if there is a path from Bash to Chikapu. Otherwise, print a character 'N'. Please note that this problem uses case-sensitive checker.

Sample clarification

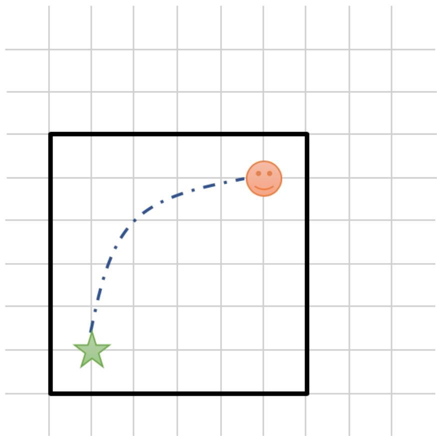
Query 1



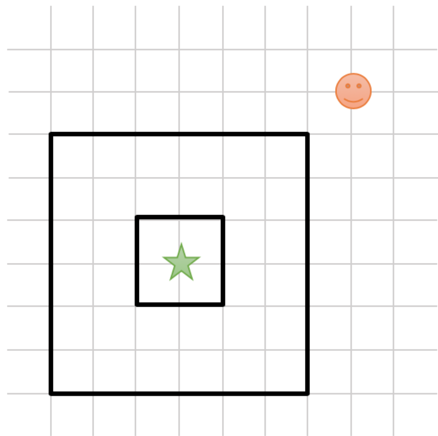
Query 3



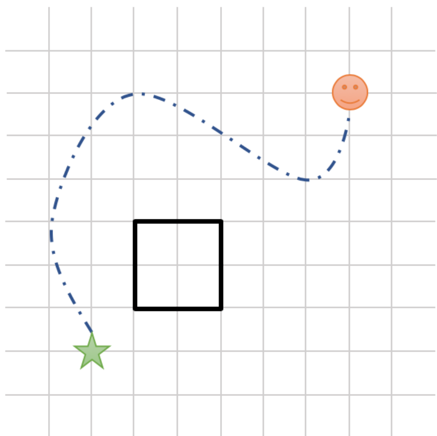
Query 4



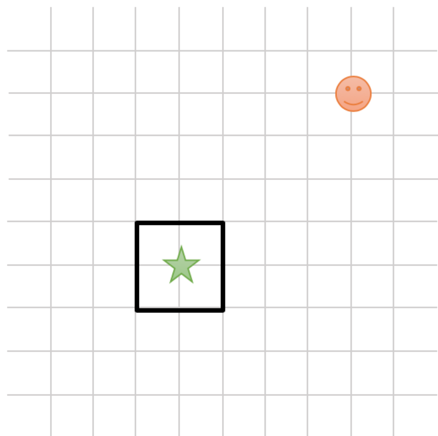
Query 6



Query 8



Query 9



Sample Input 1	Sample Output 1
9 3 2 2 8 8 1 1 1 7 7 3 2 2 8 8 3 2 2 6 6 1 3 3 5 5 3 4 4 8 8 2 2 3 2 2 8 8 3 4 4 8 8	YNYNYN

Problem L
Like No One Ever Was

Years have passed since Bash dropped out of university to become a Pokenom trainer. The adventure was full of difficulties and hardship, but Bash overcame all obstacles and became the best Pokenom trainer, like no one ever was!

Today Bash is celebrating his 13th anniversary dropping out of university and becoming a Pokenom trainer.

For the celebration party, Bash decided to prepare cakes for his N Pokenoms. Bash's Pokenoms are numbered from 1 to N, inclusive. There are N different cakes. The i-th Pokenom wants to eat the i-th cake.

The cakes are made using some ingredients. Each ingredient is uniquely marked by a prime between 1 and N, inclusive.

The recipe of the X-th cake contains k grams of ingredient p iff p^k divides X, and p^{k+1} does not. In other words, let the prime factorization of X be $X = p_1^{k_1} \times p_2^{k_2} \times \cdots \times p_m^{k_m}$, the recipe of the X-th cake contains k_1 grams of ingredient p_1, k_2 grams of ingredient p_2, $\ldots$, k_m grams of ingredient p_m.

Bash goes to a supermarket to buy ingredients. There, Bash realizes that the ingredients are very expensive. If Bash buys k grams of ingredient q, Bash's happiness decreases by $k^2 \times C_q$.

If the i-th Pokenom sees that Bash buys enough ingredient for the i-th cake, the Pokenom's happiness increases by V_i.

Please help Bash buy ingredients, so that the total happiness of Bash and his N Pokenoms is maximized!

Note that the i-th Pokenom just need to see that Bash has enough ingredient for the i-th cake. So even if the amount of ingredients Bash buys is enough to make either the x-th cake or the y-th cake, but not both, the total happiness still increases by $V_x + V_y$.

For example, considering $N = 100$ and Bash buys 2 grams of ingredient 2, 1 gram of ingredient 3 and 1 gram of ingredient 5: Bash's happiness decreases by $4 \times C_2 + 1 \times C_3 + 1 \times C_5$. Bash has enough ingredient for cakes $1..6, 10, 12, 15, 20, 30, 60$. So the happiness of the Pokenoms increases by $V_1 + V_2 + \cdots + V_6 + V_{10} + V_{12} + V_{15} + V_{20} + V_{30} + V_{60}$.

Input

- The first line contains one integer N $(1 \leq N \leq 10^4)$.
- The second line contains N integers $V_1, V_2, \ldots V_N$ $(0 \leq V_i \leq 10^4)$.
- The third line contains N integers $C_1, C_2, \ldots, C_N$ $(0 \leq C_i \leq 10^4)$. It is guaranteed that $C_i = 0$ if i is not a prime.

Output

Print a single integer B — the maximum total happiness of Bash and his Pokenoms.

Sample Input 1

```
10
1 2 3 40 5 6 7 8 9 10
0 2 3 0 5 0 7 0 0 0
```

Sample Output 1

```
51
```

Sample Input 2

```
1
2207
0
```

Sample Output 2

```
2207
```

Sample Input 3

```
2
0 3
0 5
```

Sample Output 3

```
0
```