Problem A. The one when Hendy graduated

Input file:	`shops.in`
Output file:	`standard output`
Balloon Color:	`Red`

Hendy is graduating, so he wants to have a special day with his friends. One of his friends suggested a great idea of buying a present for each one of Hendy's friends. To be fair, Hendy will buy identical presents for all of them. Hendy has K friends, so he needs K presents.

His friend suggested N shops where each shop provides an infinite supply of the present that he wants to buy. Each shop has its own price P_i. At each shop, there are three options and you have to choose **exactly** one of them:

- Take one item for free.
- Don't buy anything.
- Buy some items, for price P_i per item, where P_i is the price of the i^{th} shop.

Hendy wants to pay the minimal cost. But it's not always so simple, because it's not possible to take too many free items, so there can't be a contiguous subarray (of shops) of a length longer than M where the items were taken for free from all of them. In other words, for any subarray of shops of length $M + 1$, there must be at least one shop where you used the second or the third options.

Input

The first line of the input consists of an integer T, the number of test cases. Each test case consists of two lines, the first line contains three space-separated integers N, M, K, and the second line contains N space-separated integers $P_1, \cdots, P_N$ where $2 \leq M \leq N \leq 10^5$, $1 \leq K \leq 10^5$ and for all i, $1 \leq P_i \leq 10^6$.

Output

For each test case, output a single line containing an integer, the minimum cost.

Example

shops.in	standard output
5 5 2 6 2 3 3 3 3 5 3 10 2 4 4 4 4 5 2 12 12 5235 2711 54 12 6 2 3 2 4 2 2 1 1 5 2 100 1 1 5 7 9	6 14 108 0 97

Problem B. The one with Ayman's Cuboid

Input file:	`gcd.in`
Output file:	`standard output`
Balloon Color:	`Yellow`

Ayman has a cuboid M of size X x Y x Z, each cell will contain a number. He likes inventing some puzzles that involve the cuboid and challenge his friends with them.

This time he invented the following puzzle in which he asks to remove at most one sub-cuboid from inside such that the cells in the outer layers of the cuboid remain not touched and the GCD of all the remaining values will be a prime number and as maximum as possible. The dimensions of the removed cuboid are not necessarily equal.

When Ayman told Badry about his new puzzle, Badry found it very interesting and decided to include it in the problem set of the ECPC contest for the contestants to have some fun during the contest.

Input

The first line of the input contains a single integer T, the number of test cases.

The first line of each test case consists of three space-separated integers, X, Y and Z. $(1 \leq X, Y, Z \leq 100)$.

Then **Z** matrices follow, each matrix represents a layer in the cuboid, and is given as **X** lines, each containing **Y** integers v_i, representing integers in the cuboid. $(1 \leq v_i \leq 10^9)$. **Please be careful with the order of reading the cuboid.**

Output

For each test case output a single line containing the maximum prime GCD or 1 otherwise.

Example

gcd.in	standard output
2 1 1 1 3 3 3 3 2 4 2 2 12 4 2 6 8 2 2 4 6 5 4 2 4 8 2 4 2 2 8 4 2 6 8	3 2

Note

A **cuboid** is a three-dimensional array, like a cube but not necessarily with equal sizes.

The GCD of a set of numbers is the biggest integer value that divides all these numbers.

Problem C. The one when Badry takes the U Bahn

Input file:	`arrival.in`
Output file:	`standard output`
Balloon Color:	`Purple`

The U Bahn (The Metro) is one of the best transportation methods in Vienna. Badry loves taking the U Bahn every day to work. Sometimes he arrives at the station very early and waits for the U Bahn to arrive. In order to pass the boring waiting time, Badry sometimes rides the escalators (the moving staircases) up and down and sometimes switching from one escalator to another until the U Bahn arrives.

After some analysis, Badry found out that the escalators can be represented as a weighted directed graph of some nodes and the weighted edge between two nodes represents the time needed to go from the first node to the other. After even more analysis, He found out that the connections between the Escalators inside each U Bahn station in Vienna must be generated using the following recursive procedure:

- Graph is built level by level.
- At first level, there is a directed chain of some non zero number of nodes connected by some direct edges and all the edges have the same direction.
- We choose **zero or more** non-intersecting intervals of the chain. Each interval contains at least one node.
- For each interval, we build another chain at the next level (i.e. Second level) of one or more nodes.
- Then, we connect the new chain with the corresponding interval by two directed edges: **one from end node of the interval to the start node of the new chain and one from the end node of the new chain to the start node of the interval**.
- After connecting all intervals to new chains in the next level, we move to the next level and for each new chain of the new level, you can decide to use it as an initial chain or to stop.

in addition to the connections between the nodes representing the escalators, there is one extra source node representing the entrance of the station and connected with a direct weighted edge to the starting node of the chain in the first level and a connection between the end node of that chain to a destination node representing the platform where the U Bahn arrives.

Badry then had a question. Imagine that the time he takes to traverse any full cycle in the graph can be taken in **negative** or in **positive**. If that happened and he arrived the source node at time zero, what will be the minimum **waiting** time between **reaching the destination node** and the **arrival of the U Bahn**?

Input

The first line contains a single integer T, the number of test cases. Each test case starts with three space-separated integers N, M, S, the number of nodes including the source and the destination, the number of edges and the arrival time of the U Bahn, where $3 \leq N \leq 2,000$, $2 \leq M \leq 4,000$, $1 \leq S \leq 10^9$. Followed by M lines containing three space-separated integers x, y, c, where $1 \leq x, y \leq N$, $x \neq y$ and $1 \leq c \leq 10^9$, denoting an edge $x \to y$ with c time units cost. It is guaranteed that the given graph can be generated using the mentioned procedure.

Output

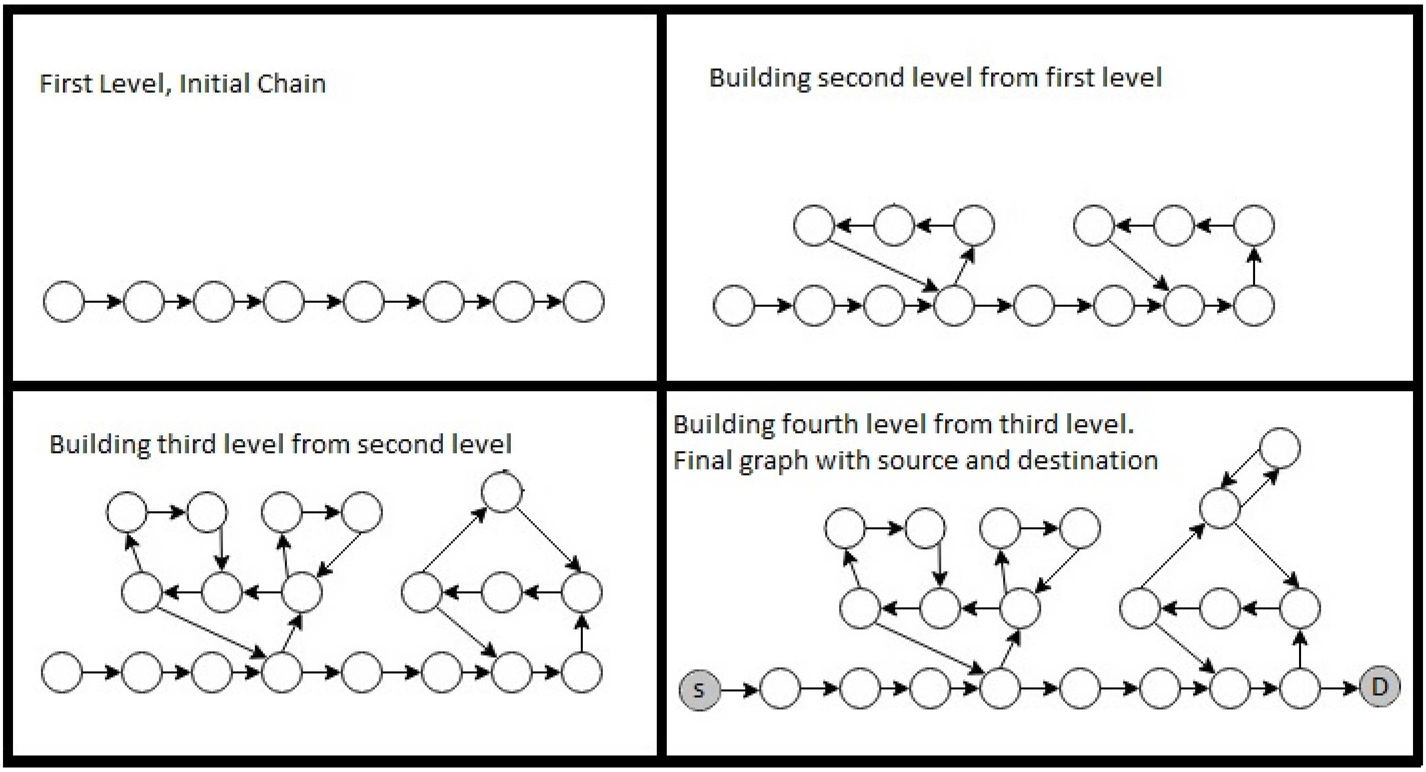
For each test case, output the minimal waiting time. In case it is impossible to reach the platform on time (or earlier), then please output 'impossible' instead.

Example

arrival.in	standard output
3 8 8 20 2 3 10 3 1 10 1 6 10 6 7 10 6 4 5 4 5 3 5 8 1 8 3 1 5 4 31 2 3 2 3 1 5 1 4 10 4 5 3 5 4 3 2 3 2 3 1 5 1 4 10 4 5 3	10 11 impossible

Note

Please note that after adding the source and the destination nodes, the source will be the only node with in-degree = 0 and the destination will be the only node with out-degree = 0. **The following figure shows the building stages of one possible graph according to the mentioned procedure:**



Problem D. The one when Anany became 'Red Pillow'

Input file:	`grid.in`
Output file:	`standard output`
Balloon Color:	`Cyan`

Anany became a red coder, so Badry wanted to celebrate his great achievement by inviting him along with two other judges to the cinema.

The seats of the cinema can be represented as a grid of size NxM where only K of these seats have one small English letter **between 'a' and 'j' inclusively** written on them.

After watching the movie and leaving the cinema, the four judges told each other that their four seats were the most comfortable seats in the world and that they should always choose them whenever they go to the cinema. Unfortunately, none of them remembers exactly where the seats were, but they remember the following :

1. The four seats had English letters on them.
2. For each seat (i, j) of the four seats, if you wrote down all the letters on the seats inside the sub-grid bounded by the upper left corner $(1, 1)$ and the lower right corner (i, j) on a piece of paper and then rearranged all letters of the four papers together, you will be able to form a palindrome of **even** length.

How many different sets of four seats could be the seats of our judges ?

Input

The first line of the input contains a single integer T, the number of test cases.
The first line of each test case consists of three space seperated integers, N, M, K, $1 \le N, M, K \le 2 \cdot 10^5$.
Each of the following K lines will contain *three* space seperated tokens c, i and j, meaning that there is a letter c on seat (i, j), where i, j are integers and $1 \le i \le N$ and $1 \le j \le M$

Output

For each test case, output a single line containing the answer to the problem mod $10^9 + 7$.

Example

grid.in	standard output
1 6 2 6 a 1 2 a 2 2 a 3 2 a 4 2 a 5 2 a 6 2	9

Note

A palindrome is word that can be read the same way from both directions. E.g, 'noon'.

Problem E. The one when they are judges

Input file:	`nodes.in`
Output file:	`standard output`
Balloon Color:	`Blue`

There's a competition held every year among N students of a university. It consists of multiple problems where each student solves these problems and validates his solutions from any of the onsite judges.

Students are divided into the rooms of the university such that each student will be in a single room, these rooms are represented as nodes of an undirected connected graph with no self-loops or multiple edges. Each edge has some weight w which represents the distance between the two nodes connected by this edge.

Eagle, Ouka, and wewark are the three onsite judges for the competition this year, each one of them must stay in one of the students' room. They are wondering what is the best room for each one of them to stay in, such that the total distance that will be covered by all students to validate their solutions is as minimum as possible. (validation by one judge is enough).

Note that distance between room u and room v is the summation of the weights of edges between them.

Help them to calculate the minimal possible distance that could be covered by students to validate their solutions.

Input

The first line of the input contains a single integer T, the number of test cases.

The first line of each test case consists of two space seperated Integers N and M where $3 \leq N \leq 50$, $1 \leq M \leq N \cdot (N-1)/2$.

Each of the following M lines will contain three space seperated Integers, u, v, w, representing an edge between u and v where $1 \leq u, v \leq N$, $1 \leq w \leq 10^9$.

Output

For each test case, output a single line containing the optimal total distance.

Example

nodes.in	standard output
1 7 7 1 2 3 3 1 2 3 2 1 3 4 5 4 5 2 5 6 1 5 7 4	6

Problem F. The one where Thrax is a coach

Input file:	`max.in`
Output file:	`standard output`
Balloon Color:	`Orange`

Medo asked Thrax to coach him during his competitive programming training. Thrax agreed and prepared an array of N different problems for Medo to solve, each having an excitement level. If a problem is very boring then it can have a negative excitement level.

Thrax then used these problems for creating contests where each contest will contain problems consisting of a sub-array of the original array he created. Each contest will contain at least two problems. The sub-arrays used in two different contests can intersect but cannot be exactly the same.

Thrax used all possible sub-arrays of length at least two for creating the contests. Each day he will invite Medo to participate in one of these contests. However, Medo is lazy so in each contest he will solve **exactly two different problems** only. Medo has an excitement level that is initially zero. When he solves a problem, its excitement level will be added to his. What is the maximum excitement level that Medo can have after participating in all the contests?

Please note that if Medo solved the same problem multiple times in different contests, its excitement level will be added each time to his total excitement level

Input

The first line of the input contains a single integer T, the number of test cases.

The first line of each test case consists of a single integer N, $1 \le N \le 10^5$, the number of problems.

The following line will contain N space-separated integers representing the excitement level of each problem, where $-10^5 \le A_i \le 10^5$.

Output

For each test case, output a single line contains the answer to the problem.

Examples

max.in	standard output
4 2 5 10 5 5 5 10 5 10 3 10 5 10 10 5 6 10 5 5 10 5 6 5 7	15 160 50 740
1 1 -10	0

Problem G. The one when Kharouba is the groom

Input file:	`matrix.in`
Output file:	`standard output`
Balloon Color:	`Gold`

Kharouba just finished his Computer Vision course. In this course, he learnt a lot about filters and how they can be applied on images to enhance them.

Kharouba is getting married very soon, so he decided to invent his own filter and use it in enhancing his wedding photos. The filter is applied on matrices that represent images and work as follows:

Given N x N matrix, you need to apply an operation K times sequentially. In each time, you will replace each cell by **the number represented by the least 4 significant digits** of the sum of its eight neighbours (if the sum is zero or consists of less than 4 digits, then use the whole number). If a neighbour doesn't exist assume that its value is 0. For a specific operation, all replacements are done at the same time depending on the values of the previous operation. Print the matrix after the last operation.

Kharouba was lately busy preparing for his wedding or more accurately 'Katb Ketab' and he didn't have time for implementing his filter, that is why Badry decided to implement the filter for him as a surprise wedding gift, but he has no idea how to do that. Could you please help Badry to surprise his friend ?

Input

The first line of the input contains a single integer T, the number of test cases.

The first line of each test case consists of two space-separated integers N and K, where $1 \leq N \leq 10$ and $1 \leq K \leq 10^{15}$.

Each of the following N lines will contain N space-separated integers representing a row in the matrix where $0 \leq M_{i,j} \leq 10^9$

Output

For each test case, output N lines representing the final matrix.

Example

matrix.in	standard output
1 3 2 99999 99999 99999 88888 88888 88888 77777 77777 77777	2206 5534 2206 3312 4416 3312 7762 1090 7762

Note

The least 4 significant digits of a number are the last 4 digits from the right, e.g The least 4 significant digits in 123456 are 3456.

Problem H. The one with Moustafa's strange plane

Input file:	`ninja.in`
Output file:	`standard output`
Balloon Color:	`Pink`

Moustafa will return to Egypt soon and he wants to bring at least two gifts to his family. Moustafa can only bring one gift from each country he visits.

Imagine that the world map is a two-dimensional plane with N countries. Each country is represented as some point (x_i, y_i). Moustafa's flight can fly starting at any point on the Y-axis and his family can wait for him at any point on the vertical line $x = S$.

Moustafa's plane is a little strange. It cannot fly continuously more than a distance of L. Besides, if the plane visits three consecutive places A, B, C, the angle between the three places on the map must be in the range $[\alpha, \beta]$.
i.e. the absolute value of the minimum angle between the lines BA and BC should be inside the range $[\alpha, \beta]$.

Moustafa wants to know the minimum distance to return to his family with at least two gifts (i.e. visit two different countries) but he was busy preparing the contest.
Can you help Moustafa finding the minimal distance?

Input

First line of input contains a single integer T, the number of test cases.

Following t test cases. The first line of each test case starts with five integers separated by spaces N, L, S, α and β. $(1 \leq N \leq 100)$, $(1 \leq L, S \leq 10^9)$, $(0 \leq \alpha \leq \beta \leq 180)$.

Each of the following N lines contains two integers separated by space x_i and y_i. $(1 \leq x_i \leq S)$, $(1 \leq y_i \leq 10^9)$

Note that angles are given in degrees and the given points are distinct

Output

Print T lines each have a single decimal of the minimal distance or '-1' (without quotes) if Moustafa cannot reach his family with at least two gifts.

The output is checked with a relative error of 10^{-4}.

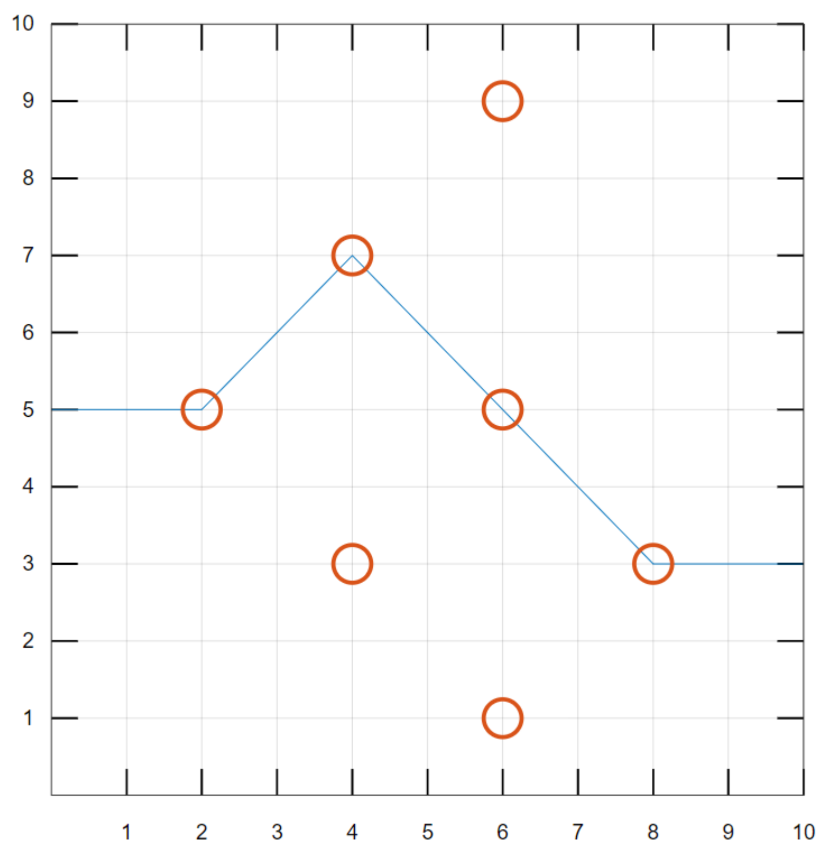
Example

ninja.in	standard output
2 7 3 10 0 180 2 5 4 3 4 7 6 1 6 5 6 9 8 3 4 2 10 0 10 1 1 3 1 5 1 7 1	12.485281 -1.000000

Note

Please note that the plane **doesn't necessarily** move always in the direction of increasing the X or the Y coordinates.

For the first sample input, the plot hereafter shows a visualization of a possible solution :



Problem I. The one when Fouad starts a startup

Input file:	`startup.in`
Output file:	`standard output`
Balloon Color:	`Black`

Fouad has a great idea for a startup. He is excited about it, but he needs a fund, so he went to **Endure Capital**, which is an early-stage investment fund company, and filed an application for funding.

Endure received N applications from different startup companies including Fouad's application and they will select the best K applications to fund.

Each application has a **profit** factor and a **risk** factor, and one application is considered to be better than the other if the difference between its **profit** factor and its **risk** factor is bigger than the difference of the other one and if there is a tie, the one that came first is considered better.

Could you please tell Fouad if his application will be accepted or not?

Input

The first line of the input consists of an integer T, the number of test cases. The first line of each case contains two space-separated integers N, K where $1 \leq N, K \leq 100$, representing the number of applications and the maximum number of applications that Endure will accept.

Each of the following N lines will represent an application and will contain two space-separated P_i, R_i where $1 \leq R_i \leq P_i \leq 100$, representing the profit factor and the risk factor of the i_{th} application.

The applications are given in the same order Endure received them and Fouad's application is the first one.

Output

For each test case, output a single line contains 'yes' if Fouad will be accepted or 'no' otherwise.

Example

startup.in	standard output
2 3 2 3 1 5 2 6 3 3 1 4 1 5 2 6 3	no yes

Problem J. The one when Safrout narrates Mafia

Input file:	`lex.in`
Output file:	`standard output`
Balloon Color:	`White`

The judges were working very hard preparing the problem set of the ECPC contest, so they decided to have a little break and have some fun. After some votes, they decided to play the most interesting game ever, 'Mafia'.

In Mafia, there is a narrator that divides the players into two groups, citizens and Mafia where the citizens try to find out who the Mafia are.

Safrout was chosen to be the narrator of the game and now he is supposed to select some players to be the Mafia. Safrout wasn't sure who he should select, so he decided to follow the following method:

He asked the judges to stand in a line then wrote their names on a piece of paper in the same order. He will then select people in such a way that when he concatenates their names in the same order as written in the paper, he gets the maximum lexicographical string possible.

Safrout already chose the Mafia and the judges had fun, but given the players' names, can you tell what was the concatenation of the Mafia's names?

Input

The first line of the input consists of an integer T, the number of test cases. The first line of each test case will contain a single integer $1 \leq N \leq 3,000$, the number of players. Each of the following N lines will contain one of the names and each name consists of lower case English letters only. The sum of lengths of all names will not exceed $3,000$.

Output

For each test case, output a single line containing the answer to the problem.

Examples

lex.in	standard output
2 3 anany ahmed badry 4 contest programming collegiate egyptian	badry programmingegyptian
2 2 a ab 2 z za	ab zza
4 3 zzx zz zz 3 zz zz zzx 4 zz zzx z z 4 zz zx z z	zzzz zzzzzzx zzzzxzz zzzz

Problem K. The one when Badry brought chocolate

Input file:	`ropes.in`
Output file:	`standard output`
Balloon Color:	`Light green`

Badry wanted to surprise the judges, so one day he brought two long chocolate bars, one dark of length L_1 and the other is white of length L_2.

For more fun the judges decided to eat the chocolate bars in rounds. At each round, two of the judges will approach the bars and take two pieces one of each bar.

In order to make things even more interesting, Badry added a rule that the lengths of the two pieces at each round should be **integer** and have the **same ratio**. In other words, $s_1/c_1 = s_2/c_2 = s_3/c_3$ .. and so on, where s_i is the size of the i_{th} piece of the first bar and c_i is the size of the i_{th} piece of the second bar.

In how many ways can the judges eat the two bars?

Two ways are different if the number of rounds is different, or the size of the i_{th} piece of the first bar is different, or the size of the i_{th} piece of the second bar is different.

Input

The first line of the input contains a single integer T, the number of test cases.
The first line of each test case consists of two space-separated numbers, L_1, L_2 where $1 \le L_1, L_2 \le 10^9$.

Output

For each test case, output a single line contains the answer to the problem modulo $10^9 + 7$.

Example

ropes.in	standard output
2 2 4 2938227 10773499	2 29992

Problem L. The one where the story is real

Input file:	`boarding.in`
Output file:	`standard output`
Balloon Color:	`Silver`

Badry was travelling via a plane and while he was standing in a long boarding line to enter the plane, he was thinking about some problems to add to the problem set of the next ECPC contest. Suddenly, he got the following problem idea:

A group of passengers are entering the plane. They should enter the plane in a line. Each one of them has a unique seat number between 1 and 10^9. If one of them reached his seat, it will take him 1 minute to put his luggage in their place and this might block some people who didn't reach their seat yet. Please note that people who reach their seats at the same time, put their luggage in parallel and also the time someone takes while walking in the plane if they are not blocked is **negligible**.

More formally passenger A is called to be blocked at some minute t if these two conditions were satisfied:

1. A didn't reach his place at some minute less than t.
2. There is some person who is putting his luggage at minute t at some seat with index less than the seat of passenger A **OR** there is some other blocked passenger in front of A that prevents A of walking further and reaching his seat.

At minute 1 the first passenger to enter is standing exactly before the first seat. **What is the total blocking times for all the passengers ?**

To add some challenges, the initial line will change dynamically. We start with an empty line and have M queries of 3 types to handle.

1. Add a passenger with seat S to the end of the line.
2. Given some query index idx, make the line the same way as it was **before** applying that query. If idx equals to the index of the current query then the line shouldn't change.
3. Output the answer to the problem for the current line.

Can you answer Badry's queries?

Input

The first line of the input contains a single integer T, the number of test cases.
The first line of each test case consists of a single integer M where $0 \leq M \leq 2 \cdot 10^5$. Each of the following M lines will contain a query i in one of the following formats :

- 1 S
- 2 idx, where $(1 \leq idx \leq i)$ and i is the the index of the current query.
- 3

Output

For each test case, output K lines where K is the number of type 3 queries and each line contains the answer for the corresponding query **OR** -1 if the current line is invalid and has 2 people with the same seat number.

Example

boarding.in	standard output
2 10 1 3 1 2 1 5 1 1 1 4 3 1 10 1 12 2 6 3 11 1 4 1 6 1 4 3 2 3 1 3 3 2 3 1 2 1 5 3	4 4 -1 2 2

Note

No two passengers can stand in front of the same seat at the same time and no person can pass the person in front of him in the line unless the person in front of him already sat.

For the first query of type three in the sample, the following figure shows the different states of the line at each minute until all people reach their seats, where circles represent people and squares represent seats. Also blocked people are marked with **filled circles** and people already sat on their seats are represented by filled squares.