Problem A. Problem with Queries

Input file:	`bersql.in`
Output file:	`standard output`

In order to solve this problem, you are to implement a simplified SQL server, which is able to answer certain queries. The input file consists of queries, each query is of one of the following types: `CREATE TABLE`, `INSERT`, `SELECT`.

The queries have the following format:

- `CREATE TABLE <table name> (<list of field names>);`
- `INSERT INTO <table name> VALUES (list of constants>);`
- `SELECT <list of specialized field names> FROM <list of table names> WHERE <logical expression>;`

Note that the format will have exactly the given format (for example, the `WHERE` part of the `SELECT` query can not be omitted). The keywords are case-insensitive.

Names of tables and fields are nonempty sequences of small English letters. Constants from the `VALUES` part of `INSERT` query are just string constants written in single quotes, for example: `'constant'`, `''`. The contents of the constants consist of small English letters. Names of tables and fields, as well as the constant contents, will be no longer than 30 symbols. No name of a table or a field will not coincide with the keywords: `CREATE`, `TABLE`, `INSERT`, `INTO`, `VALUES`, `SELECT`, `FROM`, `WHERE`.

In the format descriptions given above, *lists* are concatenations of the list elements separated by commas (see the input example).

The specialized field name is the record of the format `<table name>.<field name>`.

The logical expressions consist of one or more comparisons of table fields, given by specialized field names, which are aggregated by logic operations `AND` or `OR`. The operation `AND` has bigger priority. See the input example for clarity of the format of a logic expression.

Your task is to output the result table for each `SELECT` query. The rows of the table should go in the lexicographical order. The row A is said to be lexicographically smaller than the row B if in first i columns the rows have equal values, and in $(i+1)$-th column the value from the row A is lexicographically smaller than the value from the row B.

It is guaranteed that all queries are correct, that is, they satisfy the rules above. All the names of tables and fields, as well as the components of specialized field names, will be defined by a `CREATE TABLE` query before the query they are used in. There will not be two tables with equal names. Each table contains at least one field.

Input

The input file consists a sequence of queries of a simplified SQL dialect described in the statement. Each query will be given in a single line. Each line will contain a query.

There will be no more than 20 rows inserted in each table using the `INSERT` query. Each `SELECT` query will not refer more than three tables. One table cannot occur twice in a table list of a `SELECT` query.

Output

For each `SELECT` query output the resulting table. Each line of the resulting table should be in a separate line and should consist of the field values, listed in the first part of the `SELECT` query, separated by exactly one space. After each table you should output a line with three '`-`' symbols (that is, '`---`').

Sample input and output

bersql.in

CREATE TABLE people (id, firstname, lastname, age);
INSERT INTO people VALUES ('a', 'igor', 'kulkin', 'twentyone');
INSERT INTO people VALUES ('b', 'ivan', 'romanov', 'twentythree');
INSERT INTO people VALUES ('c', 'roman', 'alekseenkov', 'twentytwo');
INSERT INTO people VALUES ('d', 'donald', 'knuth', 'seventysix');
CREATE TABLE hobbies (id, name);
INSERT INTO hobbies VALUES ('a', 'programing');
INSERT INTO hobbies VALUES ('b', 'football');
INSERT INTO hobbies VALUES ('c', 'quake');
INSERT INTO hobbies VALUES ('d', 'work');
CREATE TABLE peoplehobby (personid, relation, hobbyid);
INSERT INTO peoplehobby VALUES ('a', 'dislikes', 'a');
INSERT INTO peoplehobby VALUES ('a', 'istolerantto', 'b');
INSERT INTO peoplehobby VALUES ('a', 'likesplaying', 'c');
INSERT INTO peoplehobby VALUES ('a', 'hateslikeeveryoneelseto', 'd');
INSERT INTO peoplehobby VALUES ('b', 'dislikes', 'a');
INSERT INTO peoplehobby VALUES ('b', 'wonteverplay', 'c');
INSERT INTO peoplehobby VALUES ('b', 'hateslikeeveryoneelseto', 'd');
INSERT INTO peoplehobby VALUES ('c', 'neverdid', 'a');
INSERT INTO peoplehobby VALUES ('c', 'loves', 'b');
INSERT INTO peoplehobby VALUES ('c', 'plays', 'c');
INSERT INTO peoplehobby VALUES ('c', 'hateslikeeveryoneelseto', 'd');
INSERT INTO peoplehobby VALUES ('d', 'hates', 'a');
INSERT INTO peoplehobby VALUES ('d', 'plays', 'b');
INSERT INTO peoplehobby VALUES ('d', 'doesnotknowwhatis', 'c');
INSERT INTO peoplehobby VALUES ('d', 'hateslikeeveryoneelseto', 'd');
SELECT people.firstname, people.lastname, peoplehobby.relation, hobbies.name
FROM people, hobbies, peoplehobby WHERE people.id = peoplehobby.personid AND
hobbies.id = peoplehobby.hobbyid;

standard output
donald knuth doesnotknowwhatis quake donald knuth hates programing donald knuth hateslikeeveryoneelseto work donald knuth plays football igor kulkin dislikes programing igor kulkin hateslikeeveryoneelseto work igor kulkin istolerantto football igor kulkin likesplaying quake ivan romanov dislikes programing ivan romanov hateslikeeveryoneelseto work ivan romanov wonteverplay quake roman alekseenkov hateslikeeveryoneelseto work roman alekseenkov loves football roman alekseenkov neverdid programing roman alekseenkov plays quake ---

Problem B. Examinator

Input file:	`examinator.in`
Output file:	`standard output`

Recently Vasya had mastered counting, but his elder brother Petya didn't believe him and decided to check his skills in the following way: Petya took several cubes, each of which had an unique number written on it. Then he started building a tower out of them, placing only one cube on each tower level. While doing that he could insert cubes into arbitrary position of the tower or remove arbitrary cubes from it. As well, he would ask Vasya questions in the following format: how many cubes are there in the tower between cubes A and B inclusive, where A and B are the numbers written on some cubes, and Vasya, who observed the process, should give the replies immediately without getting stuck. Certainly Vasya did this task very easily. However, being hurt by the way Petya unbelieved him, he decided to switch with his elder brother and in his turn check Petya's counting skills with the same task on cubes. Moreover, since Petya was older, Vasya made a decision to make the task a bit more complex: while building the tower Petya would not be present in the room and would not be able to see the tower, and Vasya would make a comment on each of his actions. Having understood what was expected from him, Petya realized very soon that his counting skills did not cope with the problem and decided to ask you to help him.

Input

On the first line of the input file you will find a single integer number N — the number of actions by Vasya ($1 \le N \le 200000$). Next there will be N lines, describing the actions themselves. The first number in each of them will be the code of the operation to be executed ($0 \le X \le 2$). Consider each of the operations separately:

0 — ask how many cubes there are between cubes A and B inclusively in the tower (in this case there will be two more integers A and B in the line, denoting the numbers of cubes);

1 — insert cube A upon cube B in the tower (in this case there will be two more integers A and B in the line, denoting the numbers of cubes; it's guaranteed that cube A is absent while cube B is present in the tower);

2 — remove cube A from tower (in this case there will be one more integer A in the line, denoting the number of the cube; it's guaranteed that cube A is present in the tower).

The numbers of the cubes are integer numbers in range from 1 to 10^5. In operation 1 (cube insertion) B can be 0, that describes the case when the cube being inserted is put on the floor, and all the tower built stacks upon this cube.

All the numbers in the input file are separated by arbitrary number of spaces and line feeds.

Output

The output file must contain as many lines as many operations 0 were there in the input file. For each of those operations one integer number should be output — the answer to be given by Petya. In case when any (or both) of the cubes is absent in the tower, the output should be 0.

Sample input and output

examinator.in	standard output

11
0 3 5
1 18 0
1 13 0
1 15 13
0 13 18
0 18 13
1 25 13
1 1 0
2 15
2 13
0 1 25
	0
	3
	3
	2

Problem C. Range Minimum Query

Input file:	`rmq.in`
Output file:	`standard output`

Giggle company opens the new office in Petrozavodsk, and you are invited to the interview. Your task is to solve the following problem.

You have to create a data structure that would maintain a list of integer numbers. All elements of the list are numbered starting from 1. Initially the list is empty. The following two operations must be supported:

- query: "`? i j`" — return the minimal element between the i-th and the j-th element of the list, inclusive;
- modification: "`+ i x`" — add element x after the i-th element of the list. If $i = 0$, the element is added to the front of the list.

Of course, the performance of the data structure must be good enough.

Input

The first line of the input file contains n — the number of operations to perform ($1 \le n \le 200\,000$). The following n lines describe operations. You may consider that no boundary violations occur. The numbers stored in the data structure do not exceed 10^9 by their absolute value.

Output

For each query operation output its result on a line by itself.

Sample input and output

rmq.in	standard output
8 + 0 5 + 1 3 + 1 4 ? 1 2 + 0 2 ? 2 4 + 4 1 ? 3 5	4 3 1

The following table shows the evolution of the list from sample input.

Operation	**The list after the operation**
initially	*empty*
`+ 0 5`	5
`+ 1 3`	5, 3
`+ 1 4`	5, 4, 3
`+ 0 2`	2, 5, 4, 3
`+ 4 1`	2, 5, 4, 3, 1

Problem D. Quadtree

Input file: `quadtree.in`
Output file: `standard output`

A *quadtree* is a data structure that can describe, for example, the coloring of some square area. A quadtree is determined recursively in the following way:

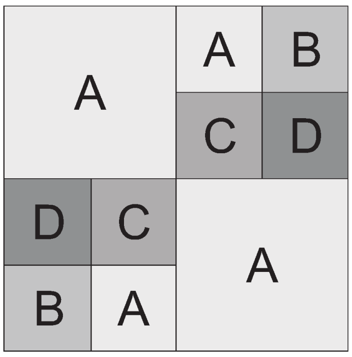
1. A quadtree defines a square area colored entirely in one color. Then the quadtree definition is just the color of the area.
2. A quadtree defines a square area which is not colored entirely in the same color. Then the quadtree definition is composed of the definitions of quadtrees which correspond to the four equal square subareas of the area.

We will represent a quadtree using a string from the following grammar:

$$
\begin{aligned}
&Q \to C\\
&Q \to \texttt{+}\ Q_{\text{upper left}}\ Q_{\text{upper right}}\ Q_{\text{lower left}}\ Q_{\text{lower right}}\\
&C \to \texttt{a}\\
&C \to \texttt{b}\\
&\dots\\
&C \to \texttt{z}
\end{aligned}
$$

where Q is a non-terminal symbol which corresponds to a quadtree, and C is a non-terminal symbol which corresponds to an area color.

For example, a string `+a+abcd+dcbaa` represents the following quadtree:



Given a quadtree definition in the form of a string s. Assume that the smallest square in this quadtree has the side length equal to 1. Find the area of the largest connected area of the same color. The area is considered to be connected, if for any two its points there exists a curve which connects these points and lies strictly inside the area.

Input

The first line of the input file contains a string s which defines a quadtree. Its length does not exceed 100 000 symbols.

Output

In the first line output the area of the largest connected area of the same color. In the second line output the symbol corresponding to the color of the largest area. If there are several such areas, output the lexicographically smallest symbol. It is guaranteed that the area does not exceed $2^{31} - 1$.

Sample input and output

quadtree.in	standard output
a	1 a
+abaa	3 a
+a+abcd+dcbaa	5 a

Problem E. Key Insertion

Input file:	`key.in`
Output file:	`standard output`

As an employee of the Macrohard Company, you have been asked to implement the new data structure that would be used to store some integer keys.

The keys must be stored in a special ordered collection that can be considered as an array A, which has an infinite number of locations, numbered starting from 1. Initially all locations are empty. The following operation must be supported by the collection: $Insert(L, K)$, where L is the location in the array and K is some positive integer value.

The operation must be processed as follows:

- If $A[L]$ is empty, set $A[L] \leftarrow K$.
- If $A[L]$ is not empty, perform $Insert(L + 1, A[L])$ and after that set $A[L] \leftarrow K$.

Given N integer numbers $L_1, L_2, \ldots, L_N$ you have to output the contents of the array after a sequence of the following operations:

$Insert(L_1, 1)$
$Insert(L_2, 2)$
$\ldots$
$Insert(L_N, N)$

Input

The first line of the input file contains N — the number of $Insert$ operations and M — the maximal position that can be used in the $Insert$ operation ($1 \le N \le 131\,072$, $1 \le M \le 131\,072$).

Next line contains N integer numbers L_i that describe $Insert$ operations to be performed ($1 \le L_i \le M$).

Output

Output the contents of the array after a given sequence of $Insert$ operations. On the first line print W — the number of the greatest location that is not empty. After that output W integer numbers — $A[1], A[2], \ldots, A[W]$. Output zeroes for empty locations.

Sample input and output

key.in	standard output
5 4 3 3 4 1 3	6 4 0 5 2 3 1

Problem F. K-th Number

Input file:	`kth.in`
Output file:	`standard output`

You are working for *Macrohard* company in data structures department. After failing your previous task about key insertion you were asked to write a new data structure that would be able to return quickly k-th order statistics in the array segment.

That is, given an array $a[1 \ldots n]$ of different integer numbers, your program must answer a series of questions $Q(i, j, k)$ in the form: "*What would be the k-th number in $a[i \ldots j]$ segment, if this segment was sorted?*"

For example, consider the array $a = (1, 5, 2, 6, 3, 7, 4)$. Let the question be $Q(2, 5, 3)$. The segment $a[2 \ldots 5]$ is $(5, 2, 6, 3)$. If we sort this segment, we get $(2, 3, 5, 6)$, the third number is 5, and therefore the answer to the question is 5.

Input

The first line of the input file contains n — the size of the array, and m — the number of questions to answer ($1 \le n \le 100\,000$, $1 \le m \le 5\,000$).

The second line contains n different integer numbers not exceeding 10^9 by their absolute values — the array for which the answers should be given.

The following m lines contain question descriptions, each description consists of three numbers: i, j, and k ($1 \le i \le j \le n$, $1 \le k \le j - i + 1$) and represents the question $Q(i, j, k)$.

Output

For each question output the answer to it — the k-th number in sorted $a[i \ldots j]$ segment.

Sample input and output

kth.in	standard output

7 3
1 5 2 6 3 7 4
2 5 3
4 4 1
1 7 3
	5
	6
	3

Problem G. Drying

Input file:	`drying.in`
Output file:	`standard output`

It is very hard to wash and especially to dry clothes in winter. But Jane is a very smart girl. She is not afraid of this boring process. Jane has decided to use a radiator to make drying faster. But the radiator is small, so it can hold only one thing at a time.

Jane wants to perform drying in the minimal possible time. She asked you to write a program that will calculate the minimal time for a given set of clothes.

There are n clothes Jane has just washed. Each of them took a_i water during washing. Every minute the amount of water contained in each thing decreases by one (of course, only if the thing is not completely dry yet). When amount of water contained becomes zero the cloth becomes dry and is ready to be packed.

Every minute Jane can select one thing to dry on the radiator. The radiator is very hot, so the amount of water in this thing decreases by k this minute (but not less than zero — if the thing contains less than k water, the resulting amount of water will be zero).

The task is to minimize the total time of drying by means of using the radiator effectively. The drying process ends when all the clothes are dry.

Input

The first line contains a single integer n ($1 \le n \le 100\,000$). The second line contains a_i separated by spaces ($1 \le a_i \le 10^9$). The third line contains k ($1 \le k \le 10^9$).

Output

Output a single integer — the minimal possible number of minutes required to dry all clothes.

Sample input and output

drying.in	standard output

3
2 3 9
5
	3

3
2 3 6
5
	2

Problem H. Inner Vertices

Input file:	`inner.in`
Output file:	`standard output`

There is an infinite square grid. Some vertices of the grid are black and other vertices are white.

A vertex V is called *inner* if it is both vertical-inner and horizontal-inner. A vertex V is called *horizontal-inner* if there are two such black vertices in the same row that V is located between them. A vertex V is called *vertical-inner* if there are two such black vertices in the same column that V is located between them.

On each step all white inner vertices became black while the other vertices preserve their colors. The process stops when all the inner vertices are black.

Write a program that calculates a number of black vertices after the process stops.

Input

The first line of the input file contains one integer number n ($0 \le n \le 100\,000$) — number of black vertices at the beginning.

The following n lines contain two integer numbers each — the coordinates of different black vertices. The coordinates do not exceed 10^9 by their absolute values.

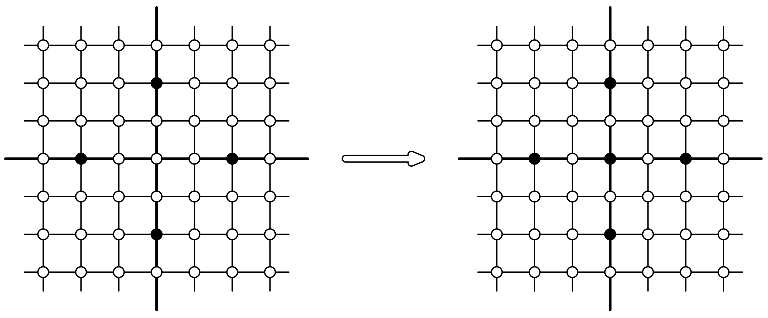
Output

Output the number of black vertices when the process stops. If the process does not stop, output `-1`.

Sample input and output

inner.in	standard output

4
0 2
2 0
-2 0
0 -2
	5



Problem I. Jenny's First Exams

Input file:	`jenny.in`
Output file:	`standard output`

First exams cause many problems to Jenny. One problem is that Jenny needs the whole day to prepare for any exam (good news is she needs only one day for any preparation). Another problem: in a day of the exam Jenny is not able to study anything. And the main problem: Jenny must prepare for i-th exam not earlier then t_i days before it, in the other case she forgets absolutely everything by the time of the exam.

Jenny wants to start preparations as later as possible but she has to pass all exams. Help Jenny to choose a day when she must start.

Input

The first line of the input file contains n ($1 \le n \le 50\,000$) — the number of exams. The following lines describes exams.

Each description consists of three lines. The first line is the name of the subject (a string containing only Latin letters, maximal length is 10). The second line is the date of the exam in format `dd.mm.yyyy`. The third line contains t_i for this exam ($1 \le t_i \le 100\,000$).

All exams take place in interval from 01.01.1900 to 31.12.2100.

Recall that if the year is divisible by 4 and is not divisible by 100, or is divisible by 400 — it is the leap one. Such year has 366 days, the additional day is on February 29.

Output

Output the latest date when Jenny may start preparation and pass all exams. Write date in format `dd.mm.yyyy`. If it is impossible to pass all the exams, output the word “`Impossible`”.

Sample input and output

jenny.in	standard output

`3`
`Philosophy`
`01.01.1900`
`1`
`Algebra`
`02.01.1900`
`3`
`Physics`
`04.01.1900`
`10`
	`30.12.1899`

`2`
`Philosophy`
`29.06.2005`
`1`
`Algebra`
`30.06.2005`
`2`
	`Impossible`

Problem J. K Best

Input file:	`k.in`
Output file:	`standard output`

Demy has n jewels. Each of her jewels has some value v_i and weight w_i.

Since her husband John got broke after recent financial crises, Demy has decided to sell some jewels. She has decided that she would keep k best jewels for herself.

She decided to keep such jewels that their specific value is as large as possible. That is, denote the specific value of some set of jewels $S = \{i_1, i_2, \ldots, i_k\}$ as

$$s(S) = \frac{\sum\limits_{j=1}^{k} v_{i_j}}{\sum\limits_{j=1}^{k} w_{i_j}}.$$

Demy would like to select such k jewels that their specific value is maximal possible. Help her to do so.

Input

The first line of the input file contains n — the number of jewels Demy got, and k — the number of jewels she would like to keep ($1 \le k \le n \le 100\,000$).

The following n lines contain two integer numbers each — v_i and w_i ($0 \le v_i \le 10^6$, $1 \le w_i \le 10^6$, both the sum of all v_i and the sum of all w_i do not exceed 10^7).

Output

Output k numbers — the numbers of jewels Demy must keep. If there are several solutions, output any one.

Sample input and output

k.in	standard output

3 2
1 1
1 2
1 3
	1 2