Problem A. Astronomy

Input file:	`astronomy.in`
Output file:	`astronomy.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

There are n planets in the planetary system of star X. They orbit star X in circular orbits located in the same plane. Their tangent velocities are constant. Directions of orbiting of all planets are the same.

Sometimes the event happens in this planetary system which is called *planet parade*. It is the moment when all planets and star X are located on the same straight line.

Your task is to find the length of the time interval between two consecutive planet parades.

Input

The first line of the input file contains n — the number of planets ($2 \le n \le 1\,000$).

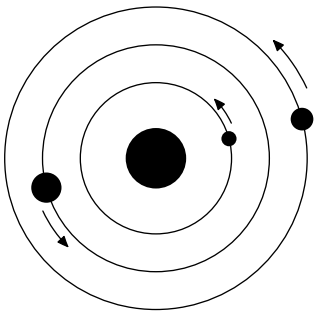
Second line contains n integer numbers t_i — the orbiting periods of planets ($1 \le t_i \le 10\,000$). Not all of t_i are the same.

Output

Output the answer as a common irreducible fraction, separate numerator and denominator by a space.

Example

astronomy.in	astronomy.out
3 6 2 3	3 1



Problem B. Bee Garden

Input file:	bee.in
Output file:	bee.out
Time limit:	2 seconds
Memory limit:	64 megabytes

George's father Ben is fond of bee breeding. He has got a bee garden near his country house and he likes to walk around the garden and gather honey from the hives.

Ben has n hives, located at various points of his garden. Some of them are connected to each other or Ben's house by straight roads. Ben always walks along the roads, and he never moves from one road to another anywhere except at hives. Neither Ben's house nor any hive is located on a road.

The roads are organized in such a way, that Ben has exactly one way to get from his house to any hive.

Each week Ben makes a roundtrip. He starts from his house and walks by the roads in such a way that he visits each hive at least once. George once came to see his father and noticed that his father takes the shortest path that allows him to visit each hive and return to his house.

He thought that his father is getting older, so Ben must be getting really tired with all that walking. He decided to build another straight road that would help his father to visit all the hives faster. That road must connect two hives or a hive and Ben's house. Help George to build the road in such a way that his father needs to walk as short a path as possible to visit all the hives and return home.

Note that neither Ben's house nor any hive must be located on the new road.

Input

The first line of the input file contains n — the number of the hives ($1 \le n \le 200$).

Let us consider Ben's garden located on a Cartesian plane in such a way that his house occupied the point $(0, 0)$. The following n lines contain two integer numbers each — the coordinates of the hives. The coordinates do not exceed 10 000 by their absolute values, no two hives coincide, no hive is located at $(0, 0)$. Let the hives be numbered from 1 to n.

The following n lines describe roads — each road is described by the numbers of hives it connects, Ben's house is denoted by 0.

Output

Output two numbers — the numbers of the hives to be connected by the road. Output 0 instead of hive's number for Ben's house.

If there is no straight road that shortens Ben's roundtrip output "-1".

Example

bee.in	bee.out
3 1 0 1 1 0 1 0 2 1 2 1 3	0 3

Problem C. Cutting a Block

Input file:	`cutting.in`
Output file:	`cutting.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

Carpenter Bill has a huge wooden block. The block has a shape of rectangular parallelepiped.

The block is so huge that it cannot pass through the door of Bill's house. So he decided to cut it into n smaller blocks. Bill is not a very smart guy so he wants to make all small blocks rectangular, and all of them should be equal.

Write a program that would help Bill to cut his block.

Input

Let us introduce a coordinate system such that the edges of the block are parallel to the coordinate axes and one of the block's corners is placed at the origin. The opposite corner of the block has coordinates (x, y, z).

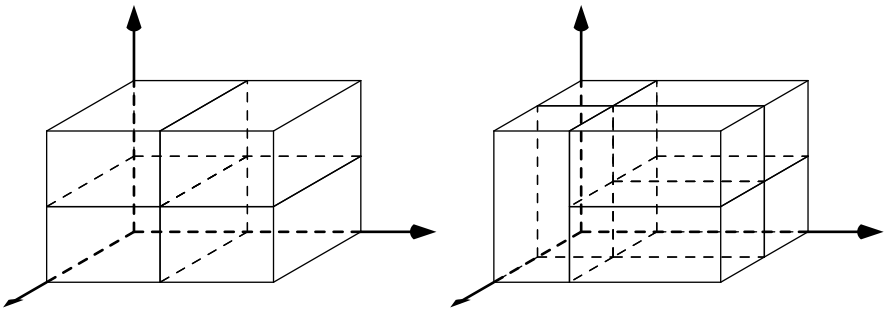
The first line of the input file contains four integer numbers — x, y, z and n ($1 \le x, y, z, n \le 1\,000$).

Output

Output file should contain n lines each describing one small block. Each small block is described by six numbers: x_1, y_1, z_1, x_2, y_2, z_2, where (x_1, y_1, z_1) are the coordinates of one corner of the block and (x_2, y_2, z_2) are the coordinates of its opposite corner.

Coordinates must be precise up to eight digits after the decimal point.

Example



cutting.in	cutting.out
3 2 2 4	0 0 0 1.5 1 2 3 0 0 1.5 1 2 0 1 0 1.5 2 2 3 1 0 1.5 2 2
3 2 2 6	0 0 0 1 2 1 0 0 1 1 2 2 1 0 0 3 1 1 1 0 1 3 1 2 1 1 0 3 2 1 1 1 1 3 2 2

Problem D. Drying

Input file:	drying.in
Output file:	drying.out
Time limit:	2 seconds
Memory limit:	64 megabytes

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.

Example

drying.in	drying.out
3 2 3 9 5	3
3 2 3 6 5	2

Problem E. Expectation

Input file:	`expectation.in`
Output file:	`expectation.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

Eric has constructed an easy scheme for generating random integer numbers. His scheme inputs an integer n and generates evenly distributed random integer value between 0 and $(n-1)$ inclusive. For example, if $n = 3$, the scheme generates 0, 1 or 2, each with probability 1/3.

Now Eric is planning to construct more complicated schemes. The first one will consist of two independent random generators with their outputs forwarded to his favorite `XOR` gate, which does bitwise *exclusive or* with its two inputs.

Eric's friend Nick studies math. Nick said that the scheme is very interesting, and the most interesting thing is the expectation of the result. Now they both think how to calculate the expectation. Could you help them?

Remember that the expectation of the random variable is its average value. For a variable ξ with non-negative integer values it can be calculated as

$$\mathbf{E}\xi = \sum_{i=0}^{+\infty} i \cdot p_i,$$

where p_i is the probability of ξ being equal to i.

Input

The first line of the input contains the number k of cases to solve ($1 \le k \le 1\,000$). Each case consists of a single integer $1 \le n \le 10^9$ on a separate line.

Output

For each case output the expected value of the new Eric's scheme with at least two digits after the decimal point. Output each result on a separate line.

Example

expectation.in	expectation.out
2 3 4	1.33 1.50

Problem F. Flip and Turn

Input file:	`flip.in`
Output file:	`flip.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

Let us define a set of operations on a rectangular matrix of printable characters.

A matrix A with m rows (1-st index) and n columns (2-nd index) is given. The resulting matrix B is defined as follows.

- Transposition by the main diagonal (operation identifier is '`1`'): $B_{j,i} = A_{i,j}$
- Transposition by the second diagonal ('`2`'): $B_{n-j+1,m-i+1} = A_{i,j}$
- Horizontal flip ('`H`'): $B_{m-i+1,j} = A_{i,j}$
- Vertical flip ('`V`'): $B_{i,n-j+1} = A_{i,j}$
- Rotation by 90 ('`A`'), 180 ('`B`'), or 270 ('`C`') degrees clockwise; 90 degrees case: $B_{j,m-i+1} = A_{i,j}$
- Rotation by 90 ('`X`'), 180 ('`Y`'), or 270 ('`Z`') degrees counterclockwise; 90 degrees case: $B_{n-j+1,i} = A_{i,j}$

You are given a sequence of no more than 100 000 operations from the set. Apply the operations to the given matrix and output the resulting matrix.

Input

At the first line of the input file there are two integer numbers — m and n ($0 < m, n \le 300$). Then there are m lines with n printable characters per line (we define a printable character as a symbol with ASCII code from 33 to 126 inclusive). There will be no additional symbols at these lines.

The next line contains the sequence operations to be performed, specified by their one-character identifiers. The operations should be performed from left to right.

Output

Two integer numbers, the number of rows and columns in the output matrix. Then the output matrix must follow, in the same format as the input one.

Example

flip.in	flip.out
3 4 0000 a0b0 cdef A1	3 4 cdef a0b0 0000

Problem G. Godfather

Input file:	`godfather.in`
Output file:	`godfather.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

Last years Chicago was full of gangster fights and strange murders. The chief of the police got really tired of all these crimes, and decided to arrest the mafia leaders.

Unfortunately, the structure of Chicago mafia is rather complicated. There are n persons known to be related to mafia. The police have traced their activity for some time, and know that some of them are communicating with each other. Based on the data collected, the chief of the police suggests that the mafia hierarchy can be represented as a tree. The head of the mafia, Godfather, is the root of the tree, and if some person is represented by a node in the tree, its direct subordinates are represented by the children of that node. For the purpose of conspiracy the gangsters only communicate with their direct subordinates and their direct master.

Unfortunately, though the police know gangsters' communications, they do not know who is a master in any pair of communicating persons. Thus they only have an undirected tree of communications, and do not know who Godfather is.

Based on the idea that Godfather wants to have the most possible control over mafia, the chief of the police has made a suggestion that Godfather is such a person that after deleting it from the communications tree the size of the largest remaining connected component is as small as possible. Help the police to find all potential Godfathers and they will arrest them.

Input

The first line of the input file contains n — the number of persons suspected to belong to mafia ($2 \le n \le 50\,000$). Let them be numbered from 1 to n.

The following $n - 1$ lines contain two integer numbers each. The pair a_i, b_i means that the gangster a_i has communicated with the gangster b_i. It is guaranteed that the gangsters' communications form a tree.

Output

Print the numbers of all persons that are suspected to be Godfather. The numbers must be printed in the increasing order, separated by spaces.

Example

godfather.in	godfather.out
6 1 2 2 3 2 5 3 4 3 6	2 3

Problem H. Horn Clauses

Input file:	`horn.in`
Output file:	`horn.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

Consider some set of boolean variables and a boolean formula. A set of values for the variables is called *satisfying* if the formula evaluates to true after replacing the variables by the corresponding values. A formula is called *unsatisfiable* if there exist no such set.

In general, there is no known algorithm finding the satisfying set of values in polynomial time, although it is not yet proved that such algorithm does not exist. The same holds for determining whether the given formula is unsatisfiable. Despite this there are some particular classes of boolean formulae for which the problem of satisfiability and unsatisfiability can be solved in polynomial time. Now we will define one of such classes, and your task will be to create polynomial time algorithm for it.

A *Horn clause* is a boolean formula, constructed according to the rules below. Let x be a variable from the set. Then a *literal* is a boolean expression which consists only of the variable by itself or of the variable negation: either x (*positive* literal) or $\neg x$ (*negative* literal). A *clause* is a disjunction of one or more literals. A *Horn clause* is a clause with at most one positive literal.

Consider some Horn clause $\neg n_1 \vee \neg n_2 \vee \ldots \vee \neg n_k \vee p$. It would be more convenient to replace disjunctions with implication: $(n_1 \wedge n_2 \wedge \ldots \wedge n_k) \Rightarrow p$.

For consistency, when *succedent* (the right part of the implication) is empty we will imagine that there is a constant `false` specified instead; similarly empty *antecedent* (the left part of the implication) is supposed to be `true`.

Consider a formula which is a conjunction of one or more Horn clauses. In this particular case, satisfiability and unsatisfiability problems can be solved by polynomial time algorithms. Write a program that would do it.

Input

The file consists of formulae, written according to the following BNF. Here [⟨expression⟩] means that ⟨expression⟩ may be omitted, {⟨expression⟩} means that ⟨expression⟩ may occur zero or more times. Characters in quotes denote themselves.

⟨char⟩ ⟶ 'A' | 'B' | ... | 'Z'

⟨variable⟩ ⟶ ⟨char⟩ {⟨char⟩}

⟨horn-clause⟩ ⟶ '(' [⟨variable⟩ {'&' ⟨variable⟩}] '=>' ⟨variable⟩ ')'

| '(' ⟨variable⟩ {'&' ⟨variable⟩} '=>' [⟨variable⟩] ')'

⟨formula⟩ ⟶ ⟨horn-clause⟩ {'&' ⟨horn-clause⟩}

Each formula is specified in a separate line. The total length of the input file does not exceed 20 000 characters.

Output

Your output must contain either the set of variables with their values that satisfy the corresponding formulae, or word "`unsatisfiable`". The variables may be specified in arbitrary order; the value for each variable must be specified exactly once. If there is more than one satisfying set, output any one.

Example

horn.in	horn.out
`(A&B&C=>QQQQ)&(=>A)` `(A=>)&(=>A)`	`A=true,C=false,B=false,QQQQ=false` `unsatisfiable`

Problem I. Inner Vertices

Input file:	`inner.in`
Output file:	`inner.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

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.

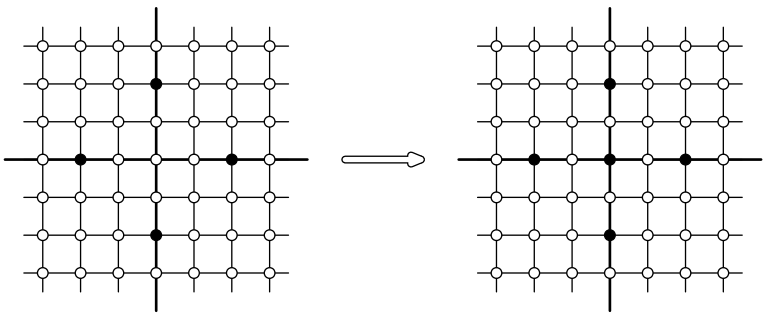
The following n lines contain two integer numbers each — the coordinates of different black vertices. The coordinates do not exceed 10^9 by their absolute values.

Output

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

Example

inner.in	inner.out
4 0 2 2 0 -2 0 0 -2	5



Problem J. Jenny's First Exams

Input file:	`jenny.in`
Output file:	`jenny.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

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`".

Example

jenny.in	jenny.out
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 K. K Best

Input file:	k.in
Output file:	k.out
Time limit:	2 seconds
Memory limit:	64 megabytes

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

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

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.

Example

k.in	k.out
3 2 1 1 1 2 1 3	1 2