# Problem A. Maximal Difference

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 1 second |
| Memory limit: | 64 mebibytes |

Vasya chooses two positive integers. The standard representation of each of them in base-$b$ numeral system consists of $N$ digits, and sums of digits in these representations are equal. Vasya wants to make a choice so that the difference between the numbers is maximal. Help Vasya to find this difference.

$2 \le b \le 36$, $1 \le N \le 10^5$.

## Input

The only line of the input file contains two integers $b$ and $N$ ($2 \le b \le 36$, $1 \le N \le 10^5$).

## Output

Output a single integer — the maximal difference between $N$-digit numbers in base-$b$ numeral system with equal sums of digits. The result should be presented in the standard notation in base-$b$ numeral system. For digits greater than 9, use uppercase latin letters: '`A`','`B`', '`C`', ..., '`Z`'.

## Examples

| standard input | standard output |
|---|---|
| 10 2 | 72 |
| 16 3 | E01 |

# Problem B. Polygon Construction

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 5 seconds |
| Memory limit: | 64 mebibytes |

You are given a sequence of $N$ integers $a_1, a_2, \ldots, a_N$. You have to construct a convex polygon with the given sides' lengths, no three vertices lie on the same line.

## Input

The first line of the input file contains the integer $N$ ($3 \le N \le 10^5$). The second line contain $N$ integers $a_1, a_2, \ldots, a_N$, all those numbers does not exceed $10^5$ by their absolute value.

## Output

Output $N$ lines. Each line should contain Cartesian coordinates of the corresponding polygon's vertex. The distance between the vertices with numbers $i$ and $i+1$ must be equal to $a_i$, and the distance between the vertices with numbers $N$ and 1 must be equal to $a_N$. All the equalities must be satisfied with an error not exceeding $10^{-5}$. All the coordinates must not exceed $2 \cdot 10^9$ by their absolute values. If it is impossible to construct a polygon with the required properties, output the single line "`Impossible`".

## Examples

| standard input | standard output |
|---|---|
| 3<br>3 4 5 | 0.0000000 3.0000000<br>0.0000000 0.0000000<br>4.0000000 0.0000000 |
| 3<br>1 2 3 | Impossible |
| 5<br>2 5 13 10 10 | 0.0000000 0.0000000<br>2.0000000 0.0000000<br>5.0000000 4.0000000<br>0.0000000 16.0000000<br>-6.0000000 8.0000000 |

# Problem C. Beautiful Patterns

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 1 second |
| Memory limit: | 64 mebibytes |

Company BrokenTiles plans to lay pattern in a yard with black and white tiles. Each of tiles has size $1 \times 1$ meter. A yard has the form of the rectangle $N \times M$ meters. However, there were $K$ tiles found, which have been already located in some squares of the yard. You have to determine, how many pattern variants can be laid in the yard, supposing that a pattern should be beautiful.

A pattern is considered to be beautiful, if in every $2 \times 2$ square, there are three black tiles and one white tile, or vice versa, one black and three white tiles.

## Input

The first line of the input file contains three integers $N$, $M$ and $K$. ($1 \le N, M \le 10^5$, $0 \le K \le 10^5$).

Each of the next $K$ lines contains three integers $x$, $y$, $c$ ($1 \le x \le N$, $1 \le y \le M$), indicating that there is a tile located in the square with coordinates $(x, y)$. If it is black, then $c = 0$, and if it is white, then $c = 1$. All the squares are different.

## Output

Output a single integer — the number of different beautiful patterns that can be laid in the yard, modulo $10^9 + 7$. Patterns are considered as different, if there is at least one square containing a white tile in one pattern and a black tile in another one.

## Examples

| standard input | standard output |
|---|---|
| 5 3 5<br>2 1 0<br>5 1 1<br>1 2 1<br>4 2 0<br>3 3 0 | 4 |

# Problem D. String without repetitions

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 1 second |
| Memory limit: | 64 mebibytes |

Lets look at the string of characters. We can say that the line $s_1 s_2 \dots s_n$ is a repetition, if there are two identical substrings that follows one after another. That is, if for some $i$ and $k$ ($i, k > 0$, $i + 2k - 1 \le n$)) following expression $s_i = s_{i+k}$, $s_{i+1} = s_{i+k+1}$, $\dots$, $s_{i+k-1} = s_{i+2k-1}$ is true.

Find a string of length $n$ without repetitions with minimum of used letters.

## Input

The single line contains an integer $n$ — the length of matched string ($1 \le n \le 4 \cdot 10^6$).

## Output

In the first line output the minimal number of different characters to be used to build a string without repetitions, and in the second line — a desired string. It is allowed to use only small Latin letters. It is guaranteed that for all input, 26 data symbols will be enough to build a string without repetitions (perhaps not optimal).

## Examples

| standard input | standard output |
|---|---|
| 5 | 3<br>abaca |

# Problem E. Beans gathering

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 1 second |
| Memory limit: | 64 mebibytes |

There is an endless one-way line, consisting of cells numbered $0, 1, 2, \ldots$. There are some beans in some cells. On each move you are allowed to perform the next step. If there are not less than $i$ beans in a cell with index $i$ ($i > 0$), the $i$ of them are taken out of the cell and are put one by one into cells with numbers $i-1, i-2, \ldots, 0$. If there are several such cells, you can make your move from any of them. Your task is to assemble all available beans in the cell with the number 0 on the strip.

## Input

The first line contains an integer $N$ ($0 \le N \le 2 \cdot 10^5$). Each of the next $N$ rows contains two integers $i_k$ and $a_k$ ($0 \le i_k \le 10^9$, $1 \le a_k \le 10^{18}$), where $i_k$ is number of the cells, and $a_k$ is number of beans in $i_k$ cell. It is guaranteed that all $i_k$ are different.

## Output

Output "`Yes`", if sequence, where all the beans can be assembled in 0 cell, exists. Otherwise,output "`No`".

## Examples

| standard input | standard output |
|---|---|
| 2<br>1 1<br>2 2 | Yes |
| 3<br>1 1<br>2 2<br>3 3 | No |
| 4<br>0 3<br>1 3<br>2 3<br>3 3 | Yes |

# Problem F. Reverse beans gathering

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 1 second |
| Memory limit: | 64 mebibytes |

There is an endless one-way strip, consisting of cells numbered $0, 1, 2, \ldots$. You have $K$ beans. You have to put them in this strip cells. You should do this in such a way that a sequence of actions exists that will lead to the fact that all $K$ beans will be in the 0-th cell. Let's recall all permitted steps. If a cell with index $i$ ($i > 0$) have at least $i$ beans , the $i$ of them are collected from that cell and placed one by one into cells with numbers $i-1, i-2, \ldots, 0$ . If there are several such cells, you can make your move from any of them.

## Input

The single line contains an integer $K$ ($0 \le K \le 10^{11}$).

## Output

Output $N$ rows where $N$ is the number of cells, which will have nonzero number of beans. Each line should contain two integers $i_k$ and $a_k$, meaning that the cell with the number $i_k$ is in the initial state for $a_k$ beans. Strings must be ordered by $i_k$ ascending.

If there are multiple cases, select anti-lexicographically minimal. Recall that the set $(x_0, x_1, \ldots, x_i, \ldots)$ anti-lexicographicaly precedes the set $(y_0, y_1, \ldots, y_i, \ldots)$, if such $i$ exists that following is true: $x_i > y_i$ and $x_j = y_j$ for all $j > i$.

## Examples

| standard input | standard output |
|---|---|
| 3 | 1 1<br>2 2 |
| 7 | 1 1<br>3 2<br>4 4 |

# Problem G. Equation

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 2 seconds |
| Memory limit: | 64 mebibytes |

There is an equation

$$a_N x^N + a_{N-1} x^{N-1} + \ldots + a_1 x + a_0 = y p^2,$$

where $a_i$, $N$ and $p$ are known integers and $x$ and $y$ are unknown integer variables, $x \geq 0$. It is required to check whether there is a solution of the equation in integers.

## Input

The first line contains two integers $p$ and $N$ ($0 \leq N \leq 20$, $2 \leq p \leq 10^6$, $p$ is prime). The second line contains $N + 1$ numbers $a_N, \ldots, a_0$ ($|a_i| \leq 10^9$).

## Output

If the equation has no solution in integers, output $-1$. Otherwise, output the value $x$ of a pair $(x, y)$, that satisfies the equation. If there are several solutions, select one in which $x$ takes the smallest non-negative value.

## Examples

| standard input | standard output |
|---|---|
| 3 2<br>1 2 3 | 3 |
| 2 2<br>1 1 1 | -1 |

# Problem H. Competition

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 1 second |
| Memory limit: | 64 mebibytes |

There are two teams on karate competition — $A$ and $B$. And there are $N$ members in each team with known level of skills. The competition consists of $N$ rounds, one member from each team is involved on each of them. According to the rules, each member can participate in only one round. Each round ends with a victory of one participant (with the higher skill level) or a draw (if the participants' skill levels are the same). If there is a draw, both teams receive 1 point, otherwise the winner brings his team 2 points, the loser gets 0. At the beginning of competition, coaches of both teams give organizers lists, where they indicate the order their fighters will go to the ring. However, coach of the team $B$ gets to know in what order does fighters of the team $A$ will participate. With this information, he want to create own list so that he can get as many points as he can, and asks you to help him.

## Input

The first line contains an integer $N$ ($1 \le N \le 2 \cdot 10^5$). The second line contains $N$ numbers $a_i$, each defines appropriate team $A$ member's skill level. Values are specified in the order, in which members will go to the ring. The third line contains $N$ numbers $b_i$ — team $B$ members' skill levels ($1 \le a_i, b_i \le 10^9$).

## Output

Output the maximum number of points that team $B$ can get, if coach of this team chooses the most optimal order.

## Examples

| standard input | standard output |
|---|---|
| 3<br>3 4 5<br>1 2 6 | 2 |
| 4<br>4 5 6 2<br>1 7 3 8 | 6 |

## Note

In the first round of the second example coach $B$ sends member with skill level 1 (which will get 0 score points in a fight with member with skill level 4), and the subsequent - karate levels 7, 8 and 3 respectively, who has won their matches against 5, 6, and 2 and has got two points each.

# Problem I. The incircle

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 1 second |
| Memory limit: | 64 mebibytes |

You are given a convex polygon. We say that some circle is inscribed in a given polygon if all its points are contained within polygon or are on its boundary. We need to find an inscribed circle with the largest radius.

## Input

The first line contains an integer $N$ ($3 \le N \le 50000$). Each of the following $N$ lines contains two real numbers $x_i$, $y_i$ ($-10^7 \le x_i, y_i \le 10^7$), with no more than 6 digits after the decimal point and determining the coordinates of the corresponding vertices of the polygon. Vertices of the polygon are given in the order of traversal. No one of these three points lie on the same line.

## Output

Print a single number — the maximum radius of the inscribed circle with accuracy of at least $10^{-5}$.

## Examples

| standard input | standard output |
|---|---|
| 3<br>2.0 0.0<br>0.0 0.0<br>1.0 1.0 | 0.414214 |
| 4<br>-1 0<br>0 1<br>2 -1<br>1 -2 | 0.707107 |

# Problem J. The dividing line

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 1 second |
| Memory limit: | 64 mebibytes |

You are given a set consisting of $N$ points on the plane. The line is considered dividing for this set, if there are two points of the set, lying in different half-planes about this line (but not on this exact line). Your task is to determine for each of the given lines, whether it is dividing or not.

## Input

The first line contains an integer $N$ ($0 \le N \le 10^5$). Each of following $N$ rows contains two integers $x_i$, $y_i$, determining the coordinates of the corresponding point in the set. The integer $M$ ($0 \le M \le 10^5$) is given in the $(N+2)$-th line. The following $M$ lines contain 4 integers $X_1$, $Y_1$, $X_2$, $Y_2$, where $(X_1, Y_1)$ and $(X_2, Y_2)$ — are two different points on the corresponding line.

All coordinates are not exceeding $10^9$ in absolute value.

## Output

Print $M$ rows, each of them defines the result for corresponding line. If the line is dividing, string should contain two numbers — the numbers of points lying in different half-planes relative to the line (points are numbered starting from one). If the line is not dividing, print number 0.

## Examples

| standard input | standard output |
|---|---|
| 4<br>0 2<br>-1 -2<br>3 1<br>2 -1<br>2<br>-1 -3 1 1<br>2 3 5 4 | 1 3<br>0 |

# Problem K. Stringangulation

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 1.5 seconds |
| Memory limit: | 64 mebibytes |

You are given a string of symbols $s_1 s_2 \ldots s_l$. It is assumed that it is cyclical, i.e. $s_l$ is followed by $s_1$. The stringangulation of this string is it's partition into three consecutive substrings (no longer cyclic, we denote them $a$, $b$ and $c$ respectively), each satisfies the triangle inequality, i.e.

$$a + b > c,\ b + c > a,\ c + a > b,$$

where the "+" symbol is usually understood as concatenation for the strings. Strings are compared lexicographically.

Your task is to find the number of its various stringangulations for a given string.

## Input

The single input line specifies the string $s$, consisting of lowercase English letters. String $s$ length is in the range from 3 to 2014.

## Output

Print a single number – the number of different stringangulations of the line $s$.

## Examples

| standard input | standard output |
|---|---|
| aaa | 1 |
| cbccbcacb | 2 |
| stringangulation | 0 |

# Problem L. Building a cube

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 1 second |
| Memory limit: | 64 mebibytes |

A 3D plane is defined by an equation $Ax + By + Cz + D = 0$. You have to build a cube with vertices located on given distances $d_1, d_2, \ldots, d_8$ from this plane.

## Input

The first line contains coefficients of the plane equation of the plane $A$, $B$, $C$ and $D$, and the second line contains distances — $d_1$, $d_2$, ..., $d_8$.

All values are real, and do not exceed $10^4$ in absolute value and have no more than two digits after the comma. At least one number of $A$, $B$ and $C$ are nonzero, all the $d_i$ values — nonnegative.

## Output

Output 8 lines, each of them contains three numbers — coordinates $x$, $y$ and $z$ of the corresponding cube vertices. The first printed point should be at the distance $d_1$ from the plane (with accuracy up to $10^{-5}$), the second one — at the distance $d_2$ and so on.

## Examples

| standard input | standard output |
|---|---|
| 1.0 1.0 1.0 1.0<br>1.0 3.0 2.0 4.0 2.0 3.0 3.0 2.0 | 0.244017 0.244017 0.244017<br>1.976068 1.976068 0.244017<br>1.976068 0.244017 0.244017<br>1.976068 1.976068 1.976068<br>0.244017 1.976068 0.244017<br>1.976068 0.244017 1.976068<br>0.244017 1.976068 1.976068<br>0.244017 0.244017 1.976068 |
| -1 2 -3 6<br>0 0 0 0 0 0 0 0 | Impossible |