We can all benefit by doing occasional "toy" programs, when artificial restrictions are set up, so that we are forced to push our abilities to the limit. … The art of tackling miniproblems with all our energy will sharpen our talents for the real problems.

Donald E. Knuth

Quarterfinal

Central region of Russia

Rybinsk, October 17-18-2012

Input file name	INPUT.TXT
Output file name:	OUTPUT.TXT

A. Hanoi tower (64 Mb, 1 sec / test)

Each programmer knows the puzzle “The Hanoi Tower”. We will briefly remind you the conditions of this task.

There are 3 pivots: A, B, C. Initially, ***n*** disks of different diameter are placed on the pivot A: the smallest disk is placed on the top and every next one is placed in an increasing order of their diameters. The second and the third pivots are still empty.

You have to move all the disks from pivot A to pivot B, using pivot C as an auxiliary.

By one step you can take off 1 upper disk and put it either on an empty pivot or on another pivot over a disk with a bigger diameter.

Almost all books on programming contain a recursive solution of this task. In the following example you can see the procedure, written in Pascal.

```
Procedure Hanoi (A, B, C: integer; N:integer);
Begin
 If N>0 then
    Begin
       Hanoi (A, C, B, N-1);
       Writeln(‘диск ’, N, ‘ from ‘, A, ‘ to ‘, B);
       Hanoi (C, B, A, N-1)
    End
End;
```

The number of step	Disk	From	To	Combination
0.				AAA
1.	1	A	B	BAA
2.	2	A	C	BCA
3.	1	B	C	CCA
4.	3	A	B	CCB
5.	1	C	A	ACB
6.	2	C	B	ABB
7.	1	A	B	BBB

It is well known that the solution given above requires (2^n–1) steps. Taking into account the initial disposition we totally have 2^n combinations of n disks disposition between three pivots. Thus, some combinations don’t occure during the algorithm execution. For example, the combination «CAB» will not be reached during the game with ***n*** = 3 (herein the smallest disk is on pivot C, the medium one is on pivot A, the biggest one is on pivot B).

Write a program that establishes if the given combination is occurred during the game.

Input

The first line of an input file contains a single integer ***n*** – the number of disks, and the second line contains ***n*** capital letters (“A”, “B” or “C”) – the disposition of the ***n*** disks between the pivots. The first (leftmost) letter designates position (a pivot name) of the smallest disk, the second letter – position of the second lagest, and so on…

Output

The output file must contain "YES" or "NO" depending on the reachability of the disks disposition during the game.

Limitations

$1 \leq n \leq 250$

Example

Input.txt	Output.txt
3 ACB	YES

Input.txt	Output.txt
3 CAB	NO

B. Island (64 Mb, 1 sec/test)

On February 30th this year astronauts from the International Space Station flew over the Pacific Ocean and took a picture, on which was discovered a previously unknown island. On the digitized picture the island appears as a connected set of square cells. This means that someone can reach some cell of land from some other cell land, going from cell to cell through their common side. There is no other water area within the island. The island is surrounded by water.

–	–	–	–	–	–	–	–
–	–	–	+	+	+	–	–
–	–	–	+	+	+	–	–
–	+	+	+	+	+	–	–
–	+	+	+	+	+	–	–
–	–	+	+	–	+	–	–
–	–	–	–	–	–	–	–

The coastline of the island is a closed polygonal line. The water cell are marked by minus sign ("–"), and the land cell – by plus sign ("+").

The coastline cell is a cell, which have a common border with water cell. In the figure below the length of the coastline is 14 cells. The other five cells of land are internal cells of the island.

Write a program that, given dimensions of the rectangle ***n*** and ***m*** and digitized picture, calculates ***l*** – the number of cells that form the coastline of the island.

Limitations

3 ≤ ***n, m*** ≤ 1 000, ***l*** > 0.

Input

The first line of input file contains two integers ***n*** and ***m***. The following ***n*** lines contain ***m*** characters (the char "–" – cell with water, and the "+" – cell with land)

Output (Output.txt)

The output file should consist of one integer ***l*** – the number of cells that form the coastline of the island.

Examples

Input.txt	Output.txt
7 8 -------- ---+++-- ---+++-- -+++++-- -+++++-- --++-+-- --------	14

Input.txt	Output.txt
3 3 --- -+- ---	1

C. Sequence (64 Mb, 1 sec / test)

Integer sequences are very interesting mathematical objects. Let us examine a sequence generated with the use of two operations: doubling and “digit sorting”. The latter operation consists in ascending-order sort of the individual digits in the decimal representation of the argument. For example, “digit sorting” of number 5726 gives 2567.

The first member of the considered sequence is 1. To generate a member of the sequence from the previous member, double the previous one and apply “digit sorting” to the result. The first 15 members of the sequence are as follows:

1, **2**, **4**, **8**, **16**, **23**, **46**, **29**, **58**, **116**, **223**, **446**, **289**, **578**, **1156**, …

Write a program to determine the value of the ***n***-th member of this sequence.

Limitations

1 ≤ ***n*** ≤ 2 147 483 647.

Input

The first line contains an integer ***n***, the number of sequence member to be calculated.

Output

The output file should contain a single integer ***k***, the value of the ***n***-th member of the sequence.

Example

Input.txt	Output.txt
1	1

Input.txt	Output.txt
6	23

D. Selection (64 Mb, 1 sec / test)

When selecting files in an application dialog, Vasya noted that he can get the same selection in different ways.

A simple mouse click selects a single file (the existing selection is discarded). A shift-click is used to select a range of files from the file clicked last time to the current file (the existing selection is discarded). Finally, a control-click is used to invert the selection state of a single file.

Consider a sequence of actions. First we select file #5 simply by clicking it. Then, shift-clicking file #10 we get the following selection: #5, #6, #7, #8, #9, #10. If after that we control-click files #7 and #3 then we will have files #3, #5, #6, #8, #9, and #10 selected. Shift-clicking file #1 we select files #1, #2, and #3 (last time we clicked file #3, and the previous selection is gone).

Vasya is wondering, what the minimum number of clicks will be, to make a certain selection from the list of files.

Write a program to determine the optimal way of making the required selection. If there are several minimal solutions, any of them is considered correct.

Example. Suppose we are to select files #2, #5, #6, #8, #9 from a list of 10 files. A possible optimal solution will include the following clicks: 5, Shift+9, Ctrl+2, Ctrl+7.

Limitations

$1 \le \boldsymbol{n} \le 10^5$.

Input

The first line contains an integer ***n***, the number of files in the list. The following line contains ***n*** characters defining the required selection. If ***i***-th file is to be selected then there is an asterisk (“*”) in position ***i***, and a dot (“.”) otherwise.

Output

The first line of the output file must contain a single integer ***k*** – the minimum number of clicks necessary to make the given selection. The following ***k*** lines must define the way to make such a selection. Each line should contain the number of file to be clicked on the corresponding step, and a prefix “Ctrl+” or “Shift+” (without quotation marks) where necessary.

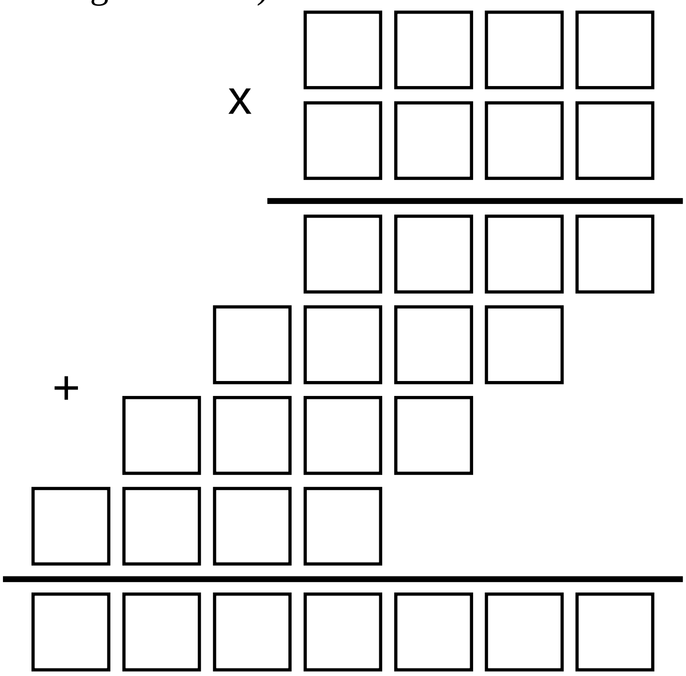
Example

Input.txt	Output.txt
10 .*..**.**.	4 5 Shift+9 Ctrl+2 Ctrl+7

Input.txt	Output.txt
10 .*..**..**	5 2 Ctrl+5 Ctrl+6 Ctrl+9 Ctrl+10

E. Multiplication Puzzle (64 Mb, 1 sec / test)

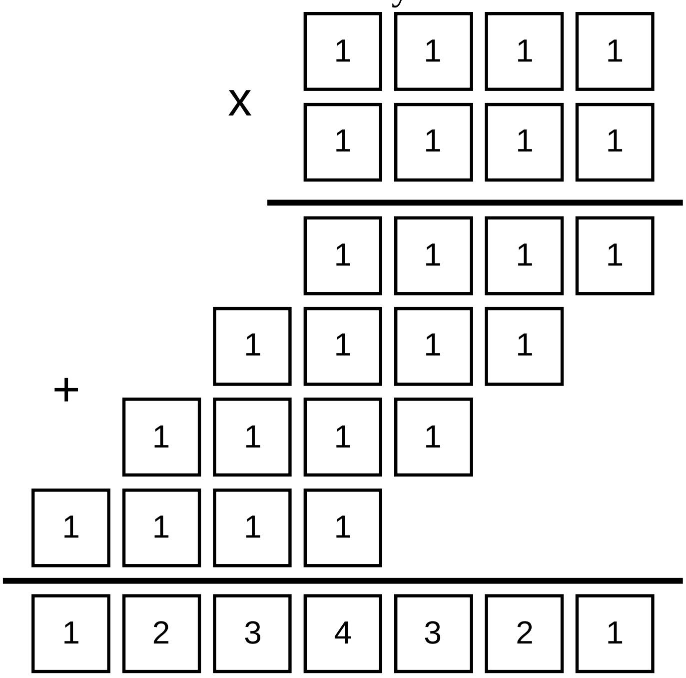
Let us consider an example of long multiplication of four-digit decimal numbers (see a figure below).



Each cell in the figure corresponds to a certain decimal digit. It is unknown what digit goes to each cell; however the total number of occurrences for each digit is given.

Write a program that, given the number of occurrences of each digit, will fill the cells in the example, thus producing a correct long multiplication whenever possible.

Example. Suppose we have 26 occurrences of digit 1, two 2's and 3's, and a single 4. In this case the cells may be filled in the following manner:



Input

The first line of the input file contains ten space-delimited integers – the number of occurrences of each decimal digit. The first integer ($\boldsymbol{n}_0$) is the number of 0's, the second one ($\boldsymbol{n}_1$) is the number of 1's, etc.

Limitations

$0 \le \boldsymbol{n}_\mathbf{i} \le 31$, $\boldsymbol{i}=0..9$; $\sum_{i=0}^{i=9} n_i \le 31$.

Output

The output file should contain two lines with an integer number in each, describing the first and the second multiplier, respectively. Leading zeros can be omitted.

If no solution exists then both lines must contain "`-1`" (without quotation marks).

Example

Input.txt	Output.txt
`20 10 1 0 0 0 0 0 0 0`	`11` `11`

Input.txt	Output.txt
`0 26 2 2 1 0 0 0 0 0`	`1111` `1111`

Input.txt	Output.txt
`0 26 2 2 0 0 0 0 0 0`	`-1` `-1`

F. GCDs (64 Mb, 1 sec / test)

Schoolboy Vasya is toying with numbers. He chooses $\boldsymbol{n}$ distinct positive integers $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_{\boldsymbol{n}}$ in ascending order and calculates the greatest common denominators (GCD) for all pairs $<\boldsymbol{a}_{\boldsymbol{i}}, \boldsymbol{a}_{\boldsymbol{j}}>$ with $\boldsymbol{i} < \boldsymbol{j}$, writing down the results in a table. For example, numbers 1, 2, 3, 4, 5, 6, 7, 8 will produce the following GCD table:

1: 1, 1, 1, 1, 1, 1, 1
2: 1, 2, 1, 2, 1, 2
3: 1, 1, 3, 1, 1
4: 1, 2, 1, 4
5, 1, 1, 1
6: 1, 2
7: 1
8:

Obviously, several sequences can produce the same GCD table, thus in general it is impossible to restore the exact original sequence using only the GCD table. Write a program that, given the GCD table, will restore a possible initial sequence $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_{\boldsymbol{n}}$. If several such sequences exist then any of them will be considered as a correct answer.

Limitations

$1 \le \boldsymbol{n} \le 100$; $1 \le \boldsymbol{a}_{\boldsymbol{i}} \le 10^9$; $1 \le \mathrm{GCD}(\boldsymbol{a}_{\boldsymbol{i}}, \boldsymbol{a}_{\boldsymbol{j}}) \le 10^4$.

Input

The first line contains an integer $\boldsymbol{n}$, the number of integers in the sequence. The following $\boldsymbol{n}-1$ lines describe the rows of the GCD table for $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_{\boldsymbol{n}-1}$, respectively. The line for $\boldsymbol{a}_{\boldsymbol{i}}$ contains $\boldsymbol{n}-\boldsymbol{i}$ integers separated by spaces: $\mathrm{GCD}(\boldsymbol{a}_{\boldsymbol{i}}, \boldsymbol{a}_{\boldsymbol{i}+1})$, $\mathrm{GCD}(\boldsymbol{a}_{\boldsymbol{i}}, \boldsymbol{a}_{\boldsymbol{i}+2})$, …, $\mathrm{GCD}(\boldsymbol{a}_{\boldsymbol{i}}, \boldsymbol{a}_{\boldsymbol{n}})$. You can safely assume that the input data is based on some real sequence $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{\boldsymbol{n}}$, which means that a solution always exists.

Output

The output file should contain $\boldsymbol{n}$ space-delimited integers: possible initial numbers $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_{\boldsymbol{n}}$, respectively.

Example

Input.txt	Output.txt
4 4 4 2 4 2 6	4 8 12 18

Input.txt	Output.txt
4 2 2 2 2 2 2	2 6 8 10

G. Function (64 Mb, 1 sec / test)

One can determine the Gray code by its number using a very simple function: G(x) = x **xor** (x **div** 2), where **xor** stands for bitwise exclusive OR (bitwise modulo 2 addition), and **div** means integer division. It is interesting to note that function G(x) is invertible, which means it is always possible to uniquely restore x given the value of G(x).

Write a program to restore number x from the given value of G(x).

Limitations

$0 \le x, y \le 10^9$.

Input

The input file contains an integer number y, the value of G(x).

Output

The output file should contain a single integer x such that G(x) = y.

Example

Input.txt	Output.txt
15	10

Input.txt	Output.txt
1723	1234

H. Milestones (64 Mb, 1 sec/test)

The longest road of the Fairy Kingdom has ***n*** milestones. A long-established tradition defines a specific color for milestones in each region, with a total of ***m*** colors in the kingdom. There is a map describing all milestones and their colors.

A number of painter teams are responsible for milestone maintenance and painting. Typically, each team is assigned a road section spanning from milestone #***l*** to milestone #***r***. When optimizing the assignments, the supervisor often has to determine how many different colors it will take to paint all milestones in the section ***l…r***.

Example. Suppose there are five milestones #1, #2, #3, #4, #5 to be painted with colors 1, 2, 3, 2, 1, respectively. In this case, only two different paints are necessary for milestones 2…4: color 2 for milestones #2 and #4, and color 3 for milestone #3.

Write a program that, given a map, will be able to handle multiple requests of the kind described above.

Limitations

$1 \le n \le 10\,000$; $1 \le m \le 255$; $1 \le l_i \le r_i \le n$; $1 \le k \le 100\,000$.

Input

The first line contains two integers, ***n*** and ***k*** – the number of milestones and the number of requests, respectively. The second line consists of ***n*** integers separated by spaces and defines the sequence of colors for milestones from #1 to #***n***. The following ***k*** lines contain pairs of integers, one pair per line. Each pair consists of two numbers – l_i and r_i – and defines a range of milestones for request ***i***.

Output

The output file should contain ***k*** integers separated by line breaks. Result number ***i*** should present the result of the ***i***-th request, i. e. the number of colors required to paint all milestones in the road section $l_i \ldots r_i$.

Example

Input.txt	Output.txt
5 3 1 2 3 2 1 1 5 1 3 2 4	3 3 2

I. Dunno (64 Mb, 1 sec/test)

All who read N. N. Nosov's books about the adventures of the boy called Dunno and his friends remember Dunno painting portraits and writing poems. But few know that Dunno also wrote several novels that attained widespread popularity. The novels featured his friends. One of them, Knowall, has closely studied Dunno's literary work and noted an interesting thing.

For each of Dunno's friends it is possible to select some novels where he takes part but no other friend takes part in all of these novels.

It is known that Dunno had n friends. Knowall wondered what was the smallest number of novels that Dunno could have produced to keep the property noted above.

For example, if Dunno had three friends Bolt, Nut, and Flower, he could have done with just three novels: first about Bolt and Nut, the second about Bolt and Flower, and the third about Nut and Flower. The only common character for the first and second novels will be Bolt, for the first and third Nut, for the second and the third Flower.

To make it easier to reason about the problem, Knowall has numbered Dunno's friends from 1 to n. Characters in each novel form a subset of the natural numbers from 1 to n. To find the common characters in the novels you take the intersection of the corresponding subsets. Initially Knowall had thought that at least n novels were necessary. But it turned out to be false. For example, for $n = 7$ one can do with 6 novels, with characters {1, 2, 3, 4, 5}, {1, 7}, {2, 3, 4, 6, 7}, {2, 5, 6}, {3, 5, 6, 7}, {4, 6}.

Then each friend can be found as the only common character of some novels:

$$\{1\} = \{1, 2, 3, 4, 5\} \cap \{1, 7\}$$
$$\{2\} = \{1, 2, 3, 4, 5\} \cap \{2, 3, 4, 6, 7\} \cap \{2, 5, 6\}$$
$$\{3\} = \{1, 2, 3, 4, 5\} \cap \{2, 3, 4, 6, 7\} \cap \{3, 5, 6, 7\}$$
$$\{4\} = \{1, 2, 3, 4, 5\} \cap \{4, 6\}$$
$$\{5\} = \{1, 2, 3, 4, 5\} \cap \{2, 5, 6\} \cap \{3, 5, 6, 7\}$$
$$\{6\} = \{2, 5, 6\} \cap \{4, 6\}$$
$$\{7\} = \{1, 7\} \cap \{3, 5, 6, 7\}$$

Write a program that for given n finds a collection of novels that possesses the property noted by Knowall. The collection must consist of the smallest possible number of novels.

Limitations

$1 \le n \le 10\,000$.

Input

The first line contains single integer n – the number of Dunno’s friends.

Output

The first line of the output file should contain single integer m – the smallest number of novels that Dunno must write. On each of the following m lines should contain the space delimited numbers of characters of the corresponding novel.

Example

Input.txt	Output.txt
5	4 1 2 3 1 4 2 4 5 3 5

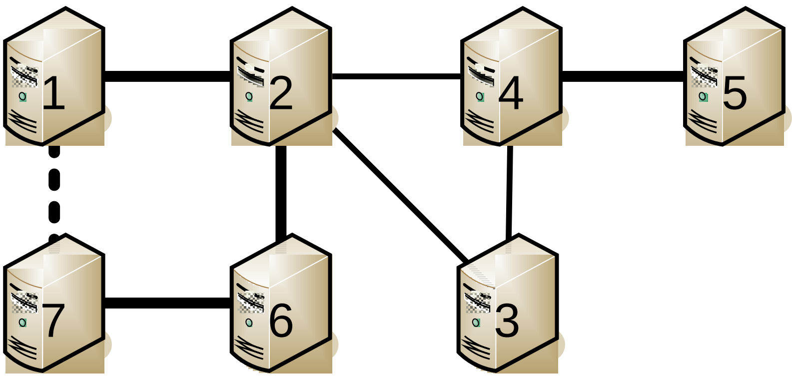
J. Computer Network (64 Mb, 1 sec / test)

The computer network of “Plunder & Flee Inc.” consists of n servers and m two-way communication links. Two servers can communicate either through a direct link, or through a chain of links, by relaying information from server to server. Current network setup enables communication for any pair of servers.

The network administrator strives to maximize network reliability. Some communication links in the network were identified as *critical*. A failure on any critical link will split the network into disconnected segments. Company management responded to the administrator’s concerns and agreed to fund another communication link, provided that when the new link goes online the number of critical links will be minimized.

Write a program that, given a network configuration, will pick a pair of servers to be connected by the new communication link. If several such pairs allow minimizing the number of critical links then any of them will be considered as a correct answer.

Example. The following figure presents a network consisting of 7 servers and 7 communication links. Essential links are shown as bold lines. A new link connecting servers #1 and #7 (dotted line) can reduce the number of the critical links to only one.



Limitations

$1 \le n \le 10\,000$; $1 \le m \le 100\,000$; $1 \le x_i, y_i \le n$; $x_i \ne y_i$.

Input

The first line contains two space-delimited integer numbers n (the number of servers) and m (the number of communication links). The following m lines describe the communication links. Each line contains two space-delimited integers x_i and y_i, which define the IDs of servers connected by link number i. Servers are identified with natural numbers ranging from 1 to n.

Output

The output file should contain a single line with space-delimited integers x and y, the IDs of servers to be connected by the new link.

Example

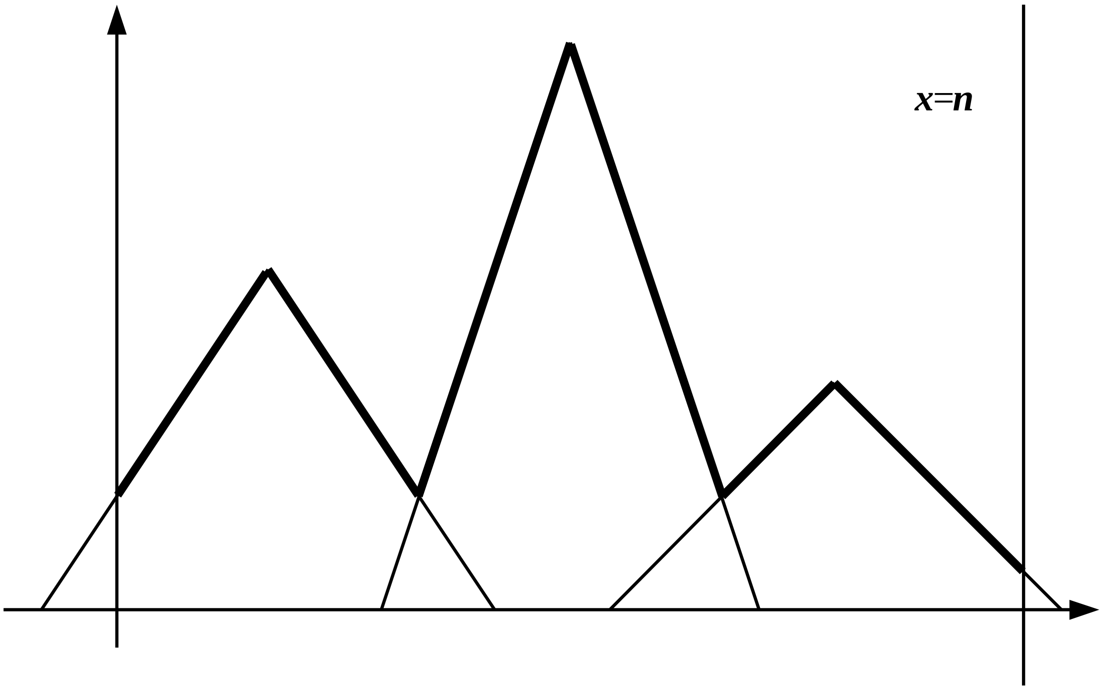
Input.txt	Output.txt
7 7 1 2 2 3 2 4 2 6 3 4 4 5 6 7	1 7

Input.txt	Output.txt
5 6 1 2 2 3 3 1 4 3 4 5 5 4	3 4

K. Triangles (64 Mb, 5 sec / test)

Schoolboy Vasya enjoys challenging geometrical problems. While reading an advanced-level math book he came across the following problem.

Located along the horizontal axis of the Cartesian plane, are k isosceles triangles. Bases of the triangles lie on the X-axis, and the remaining vertexes are in the first quadrant. The sides of adjoining triangles intersect (the right side of triangle i intersects the left side of triangle i+1). The left side of the first triangle intersects the Y-axis, and the right side of the last triangle intersects the line $x = n$. So, segments of triangle sides form a zigzag connecting the Y-axis on the left and the line $x = n$ on the right.



Let us define *peaks* as zigzag vertices formed by upper triangle corners, and *pits* as remaining zigzag vertices, formed by intersections. It is known that the coordinates of all zigzag vertices—peaks and pits—are nonnegative integers.

A question arises—how many different zigzag lines exist for given peak (not pit) coordinates? Help Vasya by writing program that will answer this question.

Limitations

$1 \le n \le 3\ 000$; $2 \le k \le 500$;
$1 \le x_i \le 3\ 000$; $1 \le y_i \le 1\ 000$; $x_i + 1 < x_{i+1}$, $i = 1\ldots k$.

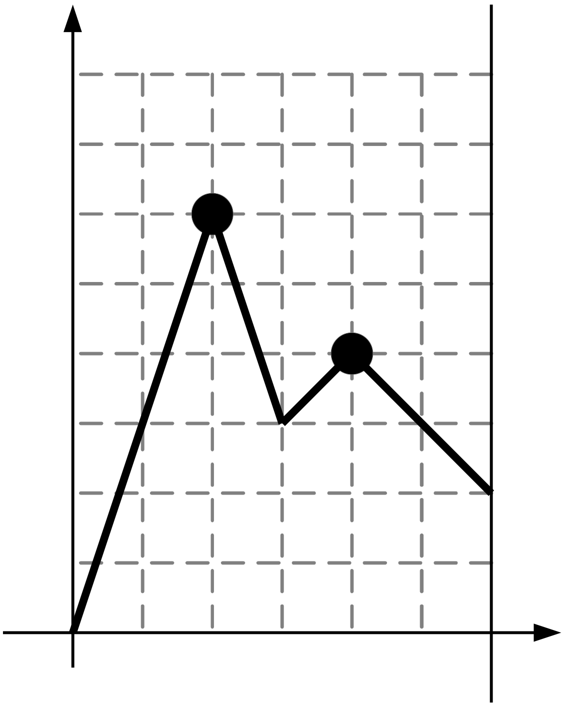
Input

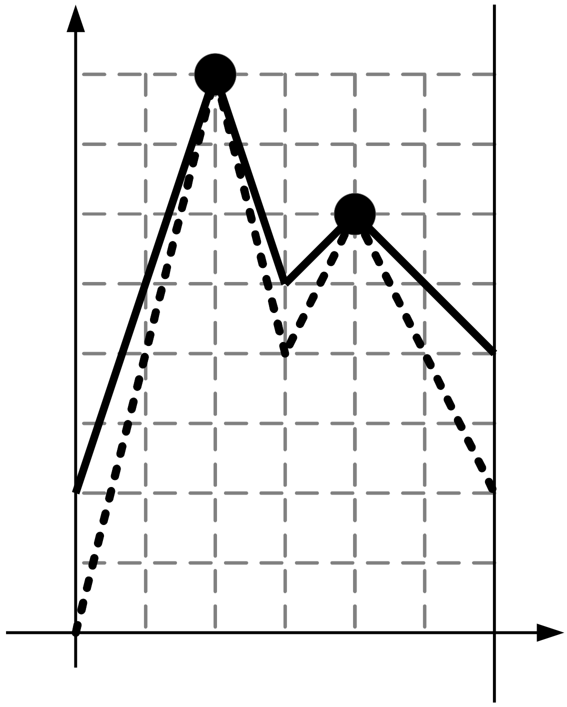
The first line contains two space-delimited integers, n (the X-coordinate of the line ($x = n$), and k (the number of triangles). The following k lines carry space-delimited pairs $\langle x_i, y_i \rangle$, the coordinates of peaks, one pair per line.

Output

The output file should contain a single integer ***k***, the number of different zigzag lines modulo 2^{30}.

Example





Input.txt	Output.txt
6 2 2 6 4 4	1

Input.txt	Output.txt
6 2 2 8 4 6	2

Rybinsk State Aviation Technical University

Task authors:
© Sergey G. Volchenkov, 2012 ***(volchenkov@yandex.ru)***
© Vladimir N. Pinaev, 2012 ***(vpinaev@mail.ru)***
© Michael Y. Kopachev, 2012 ***(rybkmu@mail.ru)***