

## Problem A. One-time passwords

Input file:            standard input  
Output file:           standard output  
Time limit:            2 seconds  
Memory limit:         256 megabytes

Nowadays two-factor authentication, when user is required to use primary password and one or more one-time passwords, is becoming more widespread. Consider one of possible ways to generate such kind of passwords.

Let  $F(Q)$  be a number of positive integers not greater than  $Q$ , which can be represented as  $2^x - 2^y$  when  $x, y$  are non-negative integer numbers. Consider all possible numbers  $Q$  such as  $F(Q) = N$  and sort them in ascending order by number of one-bits in their binary representation. If two numbers have the same number of one-bits in binary representation, they should be compared by their values. Proposed algorithm chooses  $K$ -th number in this sorted sequence.

You are required to find one-time passwords for  $T$  authentication sessions.

### Input

First line contains an integer number  $T$  — number of authentication sessions. Next  $T$  lines contain two numbers  $N_i$  and  $K_i$  each — parameters of one-time password generation algorithms.

$$1 \leq T \leq 10^5$$
$$1 \leq N_i, K_i \leq 10^{18}$$

### Output

$T$  lines containing one integer number each — one-time password for corresponding authentication session. Each password should be computed in modulo  $10^9 + 7$ . If it is impossible to generate one time password,  $-1$  should be printed.

### Example

standard input	standard output
1	42
16 10	

## Problem B. Even answer

Input file:            standard input  
Output file:           standard output  
Time limit:            1 second  
Memory limit:         256 megabytes

Vanya has an array of  $N$  integers  $a_1, a_2, \dots, a_N$ . He wants to calculate the sum of these numbers. But he also wants the sum to be non-negative, even and minimum possible at the same time. So, he decided to multiply one of the numbers by any integer before adding the numbers.

Help Vanya to get the sum he wants.

### Input

The first line of input contains an integer  $N$  — the size of the array  $A$ .

In the next line there are  $N$  integers  $a_1, a_2, \dots, a_N$  — the elements of the array  $A$ .

$$1 \leq N \leq 10^5$$
$$-10^9 \leq a_i \leq 10^9$$

### Output

Output a single integer — the required sum. If there is no answer then output -1.

### Examples

standard input	standard output
2 7 3	4
2 7 10	10
3 42 43 86	42

## Problem C. Kingdom Partition

Input file:            standard input  
Output file:           standard output  
Time limit:           1 second  
Memory limit:         256 megabytes

Recently, King of Bandiaterra Barbato started thinking how to split his kingdom between two sons Philip and Ferdinand. Barbato's property can be represented by  $N$  castles, which can be denoted as integer coordinate points  $(X_i, Y_i)$ . Since, king equally loves his sons, he decided to perform the partition according to the Bandiaterra Fairness Code.

To perform the separation, straight line parallel to Y-axis should be drawn. This line contains point  $(X, 0)$ . Castles located on the west of the border line  $(X_i \leq X)$  will be inherited by Philip and the ones on the east  $(X < X_i)$  – to Ferdinand. Eventually, each brother will have a city, which can be represented as a convex hull of castles inherited by Philip and Ferdinand, accordingly. According to the Bandiaterra Fairness Code, partition is more fair if absolute value of difference between Philip's and Ferdinand's city areas is as close as possible to  $S$  – sacral number of Bandiaterra.

Barbato conducted Bandiaterra's council of elders to decide on exact location of the border line and record it to his testament.

### Input

First line contains two integer numbers  $N$  and  $S$  — number of castles and sacral number, accordingly.

Next  $N$  lines each contain two integer numbers  $X_i$  and  $Y_i$  — coordinates of each castle. There are no castles with same coordinates. Area of an empty city which has no castles is zero.

$$1 \leq N \leq 100\,000$$

$$0 \leq S \leq 1\,000\,000\,000$$

$$|X_i|, |Y_i| \leq 10\,000$$

### Output

Single line should contain one number — absolute value between city areas, which is as close as possible to the sacral number of Bandiaterra. If there is more than one possible border lines, output that, which makes absolute difference between city areas as low as possible. Absolute and relative error of your answer should not exceed  $10^{-4}$ .

### Example

standard input	standard output
9 31 8 5 -3 1 1 -1 -4 -3 -2 -9 3 -4 9 0 1 7 -7 2	42.0000

## Problem D. Badroadville mayoral election

Input file:            standard input  
Output file:           standard output  
Time limit:            1 second  
Memory limit:         256 megabytes

Today people discuss elections at all cross-roads. Scandals, intrigues, investigations, disclosures. In our town called Badroadville it's also time to elect a mayor. And in an attempt to avoid the usual infighting, we decided to change the concept radically: the winner will be determined by deed, not by word. And a deed will be very important — it will be road construction.

There are several districts in Badroadville, where  $N$  houses are located. Houses are connected by  $M$  bidirectional roads. There are not many roads, as always. The roads between the houses are not duplicated, and there are no roads looping and leading to the same house. There is always at least one path between any pair of houses in a district (by the way, a district can consist of the only house). But there are no roads from one district to another, so unlucky town residents have to wade knee-deep through the mud.

The candidates running to be the next mayor will build one road between two houses in turns, and the one, who first turns the city into one big and happy district, will become a new mayor of Badroadville.

We suggest you take part in the election race with our candidate and prove that you are ready to rule the town and bring it to the better tomorrow!

As a welcome guest, you can choose who will be the first to start the construction of roads.

### Input

The first line contains two integers  $N$  and  $M$  — the numbers of houses and roads in the town, respectively. The next  $M$  lines describe roads as  $U_i V_i$  — the numbers of houses connected by this road.

$$\begin{aligned} 2 &\leq N \leq 200 \\ 0 &\leq M \leq N * (N - 1) / 2 \\ 1 &\leq U_i, V_i \leq N \end{aligned}$$

### Interaction Protocol

Once you've received the current Badroadville scheme of roads, you need to print one of the numbers: "1" or "2" depending on your wish to start the election race the first or the second, respectively.

Now, if it's your turn, you need to print two integers  $U V$  — the numbers of houses that you are going to connect by the road. If it is your opponent's turn now, you get two integers  $U V$  — the numbers of houses that your opponent connected by the road.

You must correctly finish the election race when there remains just one district in the town or when the opponent send you two zeroes "0 0".

Please don't forget to output a newline character and to clear a buffer. For example, you may use `fflush(stdout)` in C++, `System.out.flush()` in Java, and `flush(output)` in Pascal.

## Examples

standard input	standard output
4 0	1
3 4	1 2 1 3
5 1	1
3 5	1 2
2 3	1 5
0 0	

## Problem E. Sweet motivation

Input file:            standard input  
Output file:          standard output  
Time limit:           1 second  
Memory limit:        256 megabytes

There is nothing the Little Programmer enjoys better than calculation. At the same time, he doesn't really enjoy reading books, which he must read in Literature classes. Especially if they are thick. Especially if they are very very fat. That's why he decided to ignore reading a book containing  $N$  pages and put up with bad grade. But the literature teacher was not ready to give up, none of his students had escaped acquaintance with this productive writer. To encourage the Little Programmer's interest, he invented a rule. After every page read, the student gets just as many sweets as the absolute value of differences of the sums of the digits in the numbers of the previous and the next pages. Such a sweet motivation radically changed the situation: the book was read from the first to the last page immediately. Now all that remains is to find out how many sweets the Little Programmer can receive for such a feat. He has already counted, but the teacher needs your help.

### Input

The first and only line contains the number of pages in the book. It's the integer  $N$ .

$$1 \leq N < 10^{100\ 000}$$

### Output

The single line contains the required number of sweets. There are a lot of them, so print it modulo  $10^9 + 7$ .

### Example

standard input	standard output
29	42

## Problem F. Sort hacking

Input file:            standard input  
Output file:           standard output  
Time limit:            1 second  
Memory limit:         256 megabytes

Vova claims that he came up with a new kind of sorting algorithm. His algorithm differs in that it can sort, according to Vova, any permutation of length  $N$  while doing the same comparisons for any permutation. In total, it makes  $M$  comparisons, where the  $i$ -th comparison occurs over the numbers at positions  $x_i$  and  $y_i$  respectively. If at the time of comparison the number at the position  $x_i$  turned out to be larger, then he swaps them.

Vanya was skeptical about the discovery of Vova. He is sure that there will be such permutations that Vova's sort can not order in ascending order. Help Vanya determine the number of such permutations.

### Input

The first line contains two integers  $N$  and  $M$  — the size of the permutation and the number of comparisons in the Vova's sort.

The following  $M$  lines contain comparisons. The  $i$ -th row contains two integers  $x_i$  and  $y_i$  — the positions of the compared elements.

$$1 \leq N \leq 15$$

$$1 \leq M \leq 200$$

$$1 \leq x_i, y_i \leq N$$

### Output

In a single line output one integer — the number of permutations that Vova's sort can not order in ascending order.

### Examples

standard input	standard output
3 3 1 2 2 3 1 3	2
2 1 1 2	0
2 1 2 1	2

### Note

In the first test case, the following permutations are suitable:  $[3, 2, 1]$ ,  $[2, 3, 1]$ .

In the second test case, the following permutations are suitable:  $[1, 2]$ ,  $[2, 1]$ .

## Problem G. A Bishop's Journey

Input file:            standard input  
Output file:           standard output  
Time limit:            1 second  
Memory limit:         256 megabytes

Once upon a time there was an unusual bishop who lived on an unusual chess board. The board was unusual because it was of an arbitrary size  $N \times M$ . The bishop was unusual because it was placed in the corner field of the board. It stood there for a long time and got bored. And so it decided to go on a journey wherever its eyes lead him. As is well known, the eyes of bishops lead them diagonally. Having reached the edge of the board, the bishop turned through 90 degrees and continued to move. And so on. The bishop had stopped only when again got into some (first available) corner.

How many unique fields did it visit during his journey around the  $N \times M$  board?

### Input

The first and only line contains integers  $N$  and  $M$  — the chess board sizes.

$$1 < N \leq 10^{18}$$

$$1 < M \leq 10^{18}$$

### Output

The single line contains the only number —required number of visited fields by prime modulo  $10^{18} + 9$ .

### Example

standard input	standard output
15 22	42

## Problem H. Tourist Agency

Input file:            standard input  
Output file:           standard output  
Time limit:            1 second  
Memory limit:         256 megabytes

For the benefit of their customers, tourist agency “42 best journeys” has chosen  $N$  cities, which are connected by  $N - 1$  direct flights and any two of them are connected by air (direct flights and flights with stop-overs). To promote their service, agency has chosen contextual advertising. To avoid annoying repeating ads, provider is only showing unique tours to their customers. Tour includes visiting  $K$  ( $K \leq N$ ) cities, which are connected by  $K - 1$  direct flights and any two of them are connected by air. Tours are considered unique if one of them has particular city and the second one doesn't.

Head of agency has developed an advertisement strategy and started thinking of their expenses. Provider of contextual ad is charging one US dollar for each advertisement and one Belarusian rouble for each city included in tour. Accountant team is now required to compute final amount of money to be paid. All possible unique tours (ads) have been shown to the customers during this campaign exactly once.

### Input

First line contains one integer number  $N$  — number of chosen cities. Next  $N - 1$  lines contain two integer numbers  $U_i$  and  $V_i$  each — indices of cities which are connected by direct flight.

$$1 \leq N \leq 10^5$$

$$1 \leq U_i, V_i \leq N$$

### Output

Single line should have two integer numbers — amount of money in US dollars and Belarusian roubles to be paid by agency. Contextual Ads provider bill agency by amount modulo  $10^9 + 7$  as a part of their discounts policy.

### Example

standard input	standard output
5	17 42
4 2	
2 5	
3 2	
5 1	

## Problem I. Permutations again

Input file:            standard input  
Output file:          standard output  
Time limit:           1 second  
Memory limit:        256 megabytes

Given a sequence  $A_i$  consisting of  $N$  integers. Find the number of pairs  $(L, R)$  for which the subsegment  $\{A_L, A_{L+1}, \dots, A_R\}$  is a permutation of  $R - L + 1$  numbers.

A permutation of  $K$  numbers is any sequence of numbers from 1 to  $K$ , where each element occurs only once.

### Input

The first line contains number  $N$  — a sequence length. The second line contains  $N$  integers — sequence  $A_i$  elements.

$$1 \leq N \leq 10^6$$

$$1 \leq A_i \leq N$$

### Output

Print the number of pairs  $(L, R)$ , fulfilling the condition.

### Example

standard input	standard output
3 3 1 2	3

## Problem J. Restore the sequence

Input file:            standard input  
Output file:           standard output  
Time limit:            1 second  
Memory limit:         256 megabytes

Vanya constructed a sequence  $f_i$  according to the following rule:

1.  $f_0 = x, f_1 = y$ ;
2.  $f_i = f_{i-1} + f_{i-2}, i > 1$ .

Unfortunately, Vanya lost this sequence. However, he memorized one number  $N$ , which belongs to this sequence. He also remembers that all elements of the sequence are non-negative integers.

Help Vanya find such  $x$  and  $y$ , by which he can restore the sequence. Vanya understands that there can be many answers, so he wants the value of  $x + y$  to be as small as possible, and in case there are several such pairs,  $x$  should be the minimum possible.

### Input

The single line contains one integer  $N$  — the number that Vanya memorized.

$$1 \leq N \leq 10^6$$

### Output

In the single line output two integers  $x$  and  $y$  — the initial parameters of the sequence.

### Examples

standard input	standard output
42	0 2
19	3 2

## Problem K. Slogan

Input file:            standard input  
Output file:           standard output  
Time limit:           1 second  
Memory limit:         256 megabytes

The company «42 shades of green» consists of  $N$  departments. To increase team spirit, the company's management asked each department to come up with a slogan, which is a string consisting of at least  $K$  lower-case English characters. To assess the work of all departments of the company it was decided to choose a lexicographically minimal string that would contain at least one substring of length  $K$  from the slogans of each of the departments as a common slogan. Since the budget for printing a slogan is limited, it should not contain unnecessary characters. So let's consider for each department slogan the first occurrence of any substring of length  $K$  in the general slogan and if there is a symbol that is not covered by these occurrences, then such symbol is unnecessary and it needs to be deleted.

Help the company «42 shades of green» find their corporate slogan from the specified slogans of departments.

### Input

The first line contains two integers  $N$  and  $K$  — number of departments and length of substrings. Each following  $N$  lines contain string  $S_i$  — slogan of the  $i$ -th department.

$$1 \leq N \leq 1000$$

$$1 \leq K \leq 100$$

$$K \leq |S_i|$$

$$1 \leq K \cdot \sum |S_i| \leq 100\,000$$

### Output

In the only line, output the corporate slogan of the company.

### Examples

standard input	standard output
5 3 abacabada abada dada cadaca adac	abaacaada
5 3 abacabada abada daada cadaca adac	aada