

Problem A. A+-B

Input file: `stdin`
Output file: `stdout`
Time limit: 1 second
Memory limit: 64 megabytes

You are given the integer numbers A and B . Output $A + B$ or $A - B$.

Input

The first line contains two numbers A, B separated by a single space ($-2^{63} \leq A, B < 2^{63}$).

Output

Output one integer number: $A + B$ or $A - B$.

Examples

<code>stdin</code>	<code>stdout</code>
15 27	42
100 58	42

Problem B. Sum of sequences

Input file: `stdin`
Output file: `stdout`
Time limit: 1 second
Memory limit: 64 megabytes

You finished RIUSB ACBJSO university few years ago and started working hard and growing your carrier. At the some moment you tried to pass an interview at the XEDNAY company. You successfully answered all tricky questions about advanced algorithms and data structures and got the last one. Given two sequences A, B you need to find the following sum: $\sum_{i=1}^{|A|} \sum_{j=1}^{|B|} |A_i - B_j| * (i - j)$

Input

Input contains three lines. First contains two numbers lengths of sequences $|A|, |B|$. Second and third line contains $|A|$ and $|B|$ numbers separated by spaces ($1 \leq |A|, |B| \leq 10^5, 1 \leq A_i, B_i \leq 10^4$).

Output

Single line containing answer to the task.

Examples

stdin	stdout
5 4 3 4 5 4 4 1 2 3 4	42

Problem C. Substrings

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 64 megabytes

You are given a string S consisting of capital Latin letters. Consider the queries consisting of pairs of integer numbers L_i and R_i . For each of them you should find the number of substrings which are less than substring $S[L_i..R_i]$.

Input

The first line contains the string S ($1 \leq |S| \leq 10^5$). The second line contains the number N ($1 \leq N \leq 10^5$), denoting the number of queries. The next N lines contain pairs of numbers L_i, R_i ($1 \leq L_i \leq R_i \leq |S|$).

Output

For each query you should output the number of substrings which satisfy specified conditions.

Examples

<code>stdin</code>	<code>stdout</code>
BSUIROOPENPROGRAMMINGCHAMPIONSHIP	42
1	
1 15	

Problem D. BOPC

Input file: `stdin`
Output file: `stdout`
Time limit: 1 second
Memory limit: 64 megabytes

You invented a new chess figure and called it BOPC. Suppose it stands at the square grid at the point with coordinates X_1, Y_1 . The point with coordinates X_2, Y_2 is under attack if $|X_1 - X_2| < |Y_1 - Y_2|$. Let the power of the figure denote the number of fields under attack for all possible starting positions of BOPC. Your goal is to calculate the power of BOPC figure given the field size.

Input

Single line containing one integer N — size of the field ($1 \leq N \leq 10^9$).

Output

Single line containing power of BOPC figure given the field size modulo $10^9 + 7$.

Examples

<code>stdin</code>	<code>stdout</code>
3	26

Problem E. Hamilton traversal 2

Input file: **stdin**
Output file: **stdout**
Time limit: 1 second
Memory limit: 64 megabytes

You have a K -dimensional parallelepiped with dimensions $N_1, N_2 \dots N_k$. You should traverse the parallelepiped in order to visit each cell only once and returning to the original one. In one operation you can move into the neighboring cell either by coordinate $N_1, N_2 \dots$ or N_k .

Input

The first line of the input contains one integer K — number of dimensions. The second line contains K integer numbers $N_1, N_2 \dots N_k$. The third line contains K integers $X_1, X_2 \dots X_k$ representing your initial position.

Constraints: The product of $N_1, N_2 \dots N_k$ will be between 1 and 10^6 inclusive. K will be between 2 and 10.

Output

The output should contain the word **No** if there is no such traversal. In the other case output the word **Yes**. The following lines should contain a string in the format “C <+,->” where C is the coordinate, + or - denote the increase or decrease of this coordinate respectively. Please see the samples below for the details.

Examples

stdin	stdout
3	Yes
2 2 2	2 -
1 2 2	3 -
	2 +
	1 +
	2 -
	3 +
	2 +
	1 -

Problem F. Sexless marriage agency

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 64 megabytes

There are N individuals numbered from 1 to N in the database of clients of Martial sexless marriage agency. The founder of the agency is a programmer whose hobby is extrasensory perception, so to determine whether two individuals are happily married or not the agency counts the total number of hairs on their bodies. If the total amount of hairs is a power of two, the couple will live happily, otherwise the agency does not give any guarantees. The agency gave you the database of clients and asked for your help. Determine the maximum number of possible successful marriages given the fact that martians are monogamous and marry once in their lives.

Input

The first line contains the number N ($1 \leq N \leq 10^5$). The second line contains N integer numbers A_i ($0 \leq A_i < 10^{18}$), denoting the quantity of hairs on the martian with number i .

Output

At the first line output the number M , denoting the maximum number of possible successful marriages. At the next M lines output pairs F_i and S_i separated by a single space — indexes of married martians.

Examples

<code>stdin</code>	<code>stdout</code>
6	2
2 2 6 14 1 30	2 6 1 4

Problem G. String generator 2

Input file: `stdin`
Output file: `stdout`
Time limit: 1 second
Memory limit: 10 megabytes

You are given a sequence of integer numbers B . You should find the lexicographically smallest sequence A , consisting of the numbers from B , for which there is no i ($1 \leq i < |A|$) such that $A_i = A_{i+1}$. You can use each number from B only once.

Input

The first line of the input contains a single integer number N ($1 \leq N \leq 10^6$). The second line contains N integer numbers B_i separated by spaces ($1 \leq B_i \leq 10^6$).

Output

Output a sequence A which satisfies specified conditions. Output -1 if such sequence does not exist.

Examples

<code>stdin</code>	<code>stdout</code>
5 2 3 1 3 1	1 2 3 1 3

Problem H. Equalize

Input file: `stdin`
Output file: `stdout`
Time limit: 1 second
Memory limit: 64 megabytes

One of the most important responsibilities of an officer in the army is finding a task for soldiers. So, a brave platoon got an important mission: to group logs onto N equal piles. Logs have already been factored into N heaps, but, unfortunately, the numbers of logs in the piles were different. The execution of this order should not take less than an hour, that is why the officer gave the next restrictions: in one minute they can take exactly one log and shift it only to a neighboring pile. Help soldiers to determine how many minutes they need to equalize all the piles of logs.

Input

The number of piles is too big, so they are described by pairs K_i, A_i . It means that K_i piles in a row have A_i logs. The first line contains the number N . The next N lines contain the pairs of integer numbers K_i, A_i ($1 \leq N, K_i \leq 10^5, 0 \leq A_i \leq 10^5$).

Output

Output the smallest number of minutes the platoon needs to perform the task. Because the answer may be too large, you should output the answer modulo $10^9 + 7$.

Examples

<code>stdin</code>	<code>stdout</code>
3	42
4 8	
3 1	
2 5	

Problem I. Pair of paths

Input file: `stdin`
Output file: `stdout`
Time limit: 1 second
Memory limit: 64 megabytes

The kingdom has N castles numbered from 1 to N . The castles are connected with $N - 1$ roads. The road construction takes a huge amount of money, that's why thrifty inhabitants of the kingdom decided to plan the kingdom in such way that between every pair of castles there is exactly one path. Two brave knights independently start their journeys. Each knight randomly selects two castles where he starts and also finishes his trip. Then he goes through the kingdom by the shortest possible path and writes down all numbers of castles that he visits to his own list. After that they go to the chronicler and give him two lists of the castles which they have visited. And if there is no such castle which was visited by both knights, the chronicler will write to the annals that this pair of journeys was epic. Your goal is to find the number of existing different epic journeys pairs.

Input

At the first line N — number of castles in the kingdom ($1 \leq N \leq 10^5$). At the next $N - 1$ lines $A_i B_i$ — roads between castles.

Output

Single line containing the number of existing different epic journeys.

Examples

stdin	stdout
6 1 2 1 3 2 4 2 5 3 6	12

Problem J. Lucky tickets

Input file: `stdin`
Output file: `stdout`
Time limit: 1 second
Memory limit: 64 megabytes

Everyone knows that in less than a month the Ice Hockey World Championship will start in Minsk. But few of the guests of our city know that this is the reason why the new tickets and system of accounting were introduced in the public transport (which at the moment is unstable due to some structural flaws). It is obvious that the new tickets require the new formula for determining whether they are lucky or not. It is such an important task that it was given to the authors of BSUIR Open problems. After months of deliberation and disputes the following formula was proposed: a ticket is lucky if it is divisible by the number of symbol 1 in the binary representation. And then the authors thought that the number of these lucky tickets would be too big. Therefore, they asked you for help. Determine the quantity of lucky tickets with numbers from the interval $[1..N]$.

Input

The first line contains the integer number N ($1 \leq N \leq 10^{19}$).

Output

Output the number of lucky tickets.

Examples

<code>stdin</code>	<code>stdout</code>
153	42

Problem K. Cards

Input file: `stdin`
Output file: `stdout`
Time limit: 1 second
Memory limit: 64 megabytes

You have N cards with different numbers on them. Your goal is to find a card with a maximal number. At the beginning all cards are put into the hat. You start getting them one by one and look at the numbers on them. After each card you can select it and stop the process. If it is really the card with the maximal number you win otherwise you lose. Also you can skip the current card and continue process. Fortunately you have a friend who helps with a good strategy: you pull X cards and memorize their values. Then you continue the process and select as answer the first card with value greater than the maximal value you memorized. Unfortunately you don't know the value of X that maximizes you chances of winning. Your task is to find X .

Input

Single line containing one number: N ($5 \leq N \leq 100$).

Output

Single line containing one number: value of X that maximizes you chances of winning.

Examples

<code>stdin</code>	<code>stdout</code>
5	2