15-295 End of Year Party Prize Contest!

- Wednesday May 3, 2017, 6:30 to 9:00 pm
- The problems are ***not*** ordered by difficulty, so check the scoreboard to find the “easy” ones!
- All problems are 1 point.
- Good Luck!

Danny Sleator

Problem A. Circular Railway

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

There are L stations along a circular railway, numbered 1 through L. Trains travel in both directions, and take 1 minute to get from a station to the neighbouring one (i.e., between 1st and 2nd, between 2nd and 3rd, . . . , between $(L-1)$-th and L-th and between L-th and 1-st).

There are n employee's houses along the railway, and n offices, each house or office located near a railway station. You are to establish a one-to-one correspondence between houses and offices in such a way that total travel time (sum of travel times of each employee) is minimized.

Input

The first line of the input file contains two integer numbers, n and L ($1 \leq n \leq 50000$, $2 \leq L \leq 10^9$). The second line contains n locations of the employee's houses, and the third line contains n locations of the offices. Each location is an integer number between 1 and L. Some houses or offices or both can be located at the same railway station.

Output

Output the minimal total travel time followed by the description of the one-to-one correspondence. The description should be represented by n numbers (one for each employee, ordered as in the input), denoting the 1-based index of the office assigned to the corresponding employee.

Example

stdin	stdout

3 15
1 2 10
11 12 13
	9
	2 3 1

4 12
2 5 8 11
6 9 12 3
	4
	4 1 2 3

Problem B. Necessary Coins

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	256 megabytes

Vasya has been on vacation on Mars. He's a big fan of foreign coins, and thus has collected exactly one martian coin of each denomination, for a total of n coins: a_1 martian dollars, a_2 martian dollars, etc, a_n martian dollars. Unfortunately, he couldn't stand ordering the Pan Galactic Gargle Blaster at the Starport, and has to pay for it — it costs x martian dollars. Vasya is wondering which of his coins are absolutely necessary to do so (i.e., he is forced to abandon them). They don't offer change at the Starport Mars.

Input

The input file contains two integer numbers n and x ($1 \leq n \leq 200$, $1 \leq x \leq 10^4$), followed by n distinct integer numbers a_i ($1 \leq a_i \leq x$).

Output

On the first line of output, print the amount of denominations of coins that appear in any subset that sums to x martian dollars. On the second line of output, print the denominations themselves, in any order, separated with single spaces. It is guaranteed that there exists at least one way to pay x martian dollars with the given coins.

Example

stdin	stdout
5 18 1 2 3 5 10	2 5 10

Problem C. Greatest Greatest Common Divisor

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

Andrew has just made a breakthrough in sociology: he realized how to predict whether two persons will be good friends or not. It turns out that each person has an inner *friendship number* (a positive integer). And the *quality of friendship* between two persons is equal to the greatest common divisor of their friendship number.

That means there are *prime* people (with a prime friendship number) who just can't find a good friend, and. . . Wait, this is irrelevant to this problem.

You are given a list of friendship numbers for several people. Find the highest possible quality of friendship among all pairs of given people.

Input

The first line of the input file contains an integer n ($2 \leq n \leq 100\,000$) — the number of people to process. The next n lines contain one integer each, between 1 and 1 000 000 (inclusive), the friendship numbers of the given people. All given friendship numbers are distinct.

Output

Output one integer — the highest possible quality of friendship. In other words, output the greatest greatest common divisor among all pairs of given friendship numbers.

Sample input and output

`standard input`	`standard output`

`4`
`9`
`15`
`25`
`16`
	`5`

Problem D. Thiefs And Cops

Input file:	`stdin`
Output file:	`stdout`
Time limit:	0.5 seconds
Memory limit:	64 megabytes

A cop is pursuing a thief in a rectangular $H \times W$ grid. Both thief and cop occupy one cell of the grid. Initially they are placed in it somehow, and then make moves in turn. At each move one must go from a cell to any of the cells adjacent to it side-by-side. Note that it's not allowed to stay in the same cell. It's also not allowed to move outside the grid.

The cop catches the thief if they're in the same cell. The aim of the cop is to catch the thief as fast as possible, the aim of the thief is not to be caught, or at least to be free for as many moves as possible. They both see each other and the walls of the grid, so they always know the coordinates of themselves and of each other.

If both players play optimally, will the cop catch the thief, and if he will, after which move it will happen? (moves are numbered starting from 1, for example, if the cop moves first, then move 1 is the cop's move, move 2 is the thief's move, move 3 is the cop's move, etc)

Input

The first line of the input file contains two integers H and W, $1 \leq H, W \leq 5 \cdot 10^8$, denoting the number of rows and columns in the grid, respectively. The rows are numbered 1 through H, the columns are numbered 1 through W.

The second line of the input file contains two integers R_c and C_c, $1 \leq R_c \leq H$, $1 \leq C_c \leq W$, denoting the row and column of the cell where the cop resides initially.

The third line of the input file contains two integers R_t and C_t, $1 \leq R_t \leq H$, $1 \leq C_t \leq W$, denoting the row and column of the cell where the thief resides initially.

Initial positions of the cop and the thief differ.

The fourth line of the input file contains either the letter ’`C`’ (capital English letter C) — if the cop moves first, or the letter ’`T`’ (capital English letter T) — if the thief moves first (without quotes).

Output

If the cop can catch the thief with both of them playing optimally, output the number of the move after which it will happen. Otherwise, output 0 (zero).

Example

stdin	stdout

`2 2`
`1 2`
`2 1`
`C`
	`0`

`2 2`
`1 2`
`2 1`
`T`
	`2`

Problem E. Fast Typing

Input file:	`stdin`
Output file:	`stdout`
Time limit:	0.5 seconds
Memory limit:	256 megabytes

Vasya has little experience in typing, thus he has to look at the keyboard to locate the necessary keys, and still makes typos while doing so. For simplicity sake, we'll assume that the only type of typo he makes is replacing a character by another one. To correct those, he employs the following strategy: from time to time, he looks at the screen, and if there is any typo in the text, he removes all the characters from the end of the text to the first typo he made inclusive (i.e., he leaves only the correct part of the text intact) by pressing 'backspace' key several times, and continues typing from that position again.

Pressing any key (including 'backspace') takes 1 unit of time, and looking at the screen takes t units of time. Given the probabilities of making an error for each character in the text, compute the minimal possible expected time to type the entire text correctly (including verifying that by looking at the screen in the end and noticing no typos).

The text is n characters long, and the probability of mistyping i-th character is equal to a_i.

Input

The input file contains two integer numbers n and t ($1 \le n \le 3000$, $1 \le t \le 10^6$), followed by n real numbers a_i ($10^{-5} \le a_i \le \frac{1}{2}$).

Output

Output one real number — the minimal possible expected time. Your answer will be considered correct if it is within 10^{-6} relative error of the exact answer.

Example

stdin	stdout

`3 1`
`0.00001 0.5 0.00001`
	`8.000080000800008`

Problem F. Digits Permutation

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

Andrew has just made a breakthrough in the world of number 17: he realized that it's rather easy to permute the digits in almost any given number to get a number divisible by 17.

You are given a positive integer n. You must find a permutation of its digits that is divisible by 17.

Input

Input file contains single integer n, $1 \leq n \leq 10^{17}$.

Output

Output any permutation of digits of n that is divisible by 17. The output permutation may not start with a zero. If there is no such permutation, output -1.

Sample input and output

standard input	standard output
17	17
2242223	2222342
239	-1

Problem G . Treediff

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

Andrew has just made a breakthrough in complexity theory: he thinks that he can prove $P = NP$ if he can get a data structure which allows to perform the following operation quickly. Naturally, you should help him complete his brilliant research.

Consider a rooted tree with integers written in the leaves. For each internal (non-leaf) node v of the tree you must compute the minimum absolute difference between all pairs of numbers written in the leaves of the subtree rooted at v.

Input

The first line of the input file contains two integers n and m — overall number of nodes in the tree and number of leaves in the tree respectively. $2 \le n \le 50\,000, 1 \le m < n$. All nodes are numbered from 1 to n. Node number 1 is always the root of the tree. Each of the other nodes has a unique parent in the tree. Each of the next $n-1$ lines of the input file contains one integer — the number of the parent node for nodes $2, 3, \ldots, n$ respectively. Each of the last m lines of the input file contains one integer ranging from $-1\,000\,000$ to $1\,000\,000$ — the value of the corresponding leaf. Leaves of the tree have numbers from $n - m + 1$ to n.

Output

Output one line with $n - m$ integers: for each internal node of the tree output the minimum absolute difference between pairs of values written in the leaves of its subtree. If there is only one leaf in the subtree of some internal node, output number $2^{31} - 1$ for that node. Output the answers for the nodes in order from node number 1 to $n - m$.