

Problem A. Palindromes

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 64 megabytes

Vasya does not like palindromes, because he confuses them with polynomials. Recently Vasya has got a dog and he wants to give it a nickname. For mentioned obvious reasons, he wants to have as few palindromic substrings as possible in the nickname. The nickname should be n characters long and consist only of the first k characters of English halphabet. Please help Vasya to pick a nickname.

Palindrome is a string that can be read in the same way in both left-to-right and right-to-left directions. For example: "mononom". Any single letter is considered to be a palindrome by itself. The word "bee" contains four palindromic substrings: "b", "e" (the second letter of the word), "e" (the third letter of the word), "ee".

Input

A single line of an input contains two integers, separated by a space: n — the required nickname length, and k — the number of allowed first characters of Latin alphabet. $1 \leq n \leq 100\,000$, $1 \leq k \leq 26$.

Output

Output a nickname with the least possible number of palindromic substrings as a single line, written in lowercase letters. The nickname doesn't have to be meaningful.

Example

standard input	standard output
3 15	dog
2 1	aa

Problem B. Sequence

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 64 megabytes

Alex likes number sequences. Once he took one of his favorite sequences, and decided to create another one out of it. To do this, he calculated for each element of the given sequence a length of the longest increasing subsequence that ends in this element. These lengths formed a new sequence. After that he told his friend Sasha a new sequence. Of course, he did not describe Sasha his favorite sequence.

Help Sasha to regenerate Alex's favorite sequence.

Input

The first line of the input file contains n ($1 \leq n \leq 100\,000$) — the length of new sequence. The new sequence is given in the second line as numbers a_i , $1 \leq i \leq n$, $1 \leq a_i \leq 10^9$, separated by spaces.

Output

Output Alex's favorite sequence as a single line of numbers, separated by spaces. If several solutions exist, output any of them. It is guaranteed that solution always exists. We also know that all Alex's favorite sequences consist only of positive integers not greater than 10^9 .

Example

standard input	standard output
1 1	1
2 1 1	1 1

Problem C. Cryptanalysis

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 64 megabytes

Once upon a time Vasya and Peter wanted to implement their encrypted data exchange system. They decided to use the Diffie-Hellman algorithm for the encryption.

Diffie-Hellman algorithm consists of several steps:

1. Vasya and Peter choose the numbers g and p together.
2. Vasya chooses a secret number a and calculates $A = g^a \bmod p$, where $x \bmod y$ is the remainder on division of x by y .
3. Peter chooses a secret number b and calculates $B = g^b \bmod p$.
4. Vasya and Peter let each other know of the values of A and B .
5. Vasya calculates the number $K_1 = B^a \bmod p$.
6. Peter calculates the number $K_2 = A^b \bmod p$.

It's easy to notice that after these steps $K_1 = K_2 = K$. So Vasya and Peter can use K as their encryption key.

Later, Vasya and Peter realized that this algorithm can be accelerated by choosing $p = 2^t$, and they implemented the accelerated version of the algorithm in their system. Your task is to conduct a cryptanalysis of the system. Namely, for given numbers t, g, A, B you should find a secret key K .

Note that in this problem we deal with non-negative integers, hence $0^0 = 1$.

Input

Four lines of an input contain the following numbers: t, g, A and B . $0 \leq t \leq 1024$, $0 \leq g \leq 2^t - 1$, $0 \leq A \leq 2^t - 1$, $0 \leq B \leq 2^t - 1$. All numbers are in hexadecimal notation, using symbols 0123456789ABCDEF as numbers. Correctness of the input data and the uniqueness of an answer are guaranteed.

Output

Output a single line containing a number K in hexadecimal notation.

Example

standard input	standard output
3 5 5 1	1
6 2 4 4	10
A B 11B 133	3E3

Problem D. Suggester

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 megabytes

Eugen works at a very large company. Once he was given a task to implement a convenient autocompletion for a product search field. The task was as follows: by entering a word prefix an algorithm is required to prompt a user some appropriate phrases (from a certain set) which satisfy the given prefix. A phrase satisfies the prefix, if any of its words starts with this prefix (the words in a phrase are separated by spaces). Besides, each phrase has its own rating (user value). Therefore the phrases should be prompted in an order of decreasing rating. Each request also specifies the maximum number of phrases that need to be prompted.

Help Eugen to implement a given functionality.

Input

The first line of the input file contains two integers: n ($1 \leq n \leq 100\,000$) — the number of phrases and k ($1 \leq k \leq 100\,000$) — the number of user requests.

The next n lines contain word phrases. Each line contains a rating (positive integer up to 10^9 , different for all word phrases), a number of words in the phrase (positive number), separated by a space, and a phrase itself which consists of English lowercase characters. The total length of all word phrases does not exceed 10^5 symbols.

The next k lines describe requests. Each request is a pair (t_i, s_i) — t_i is the maximum number of phrases required to output (not necessarily all that satisfy a prefix), and a non-zero length prefix. The sum of t_i is less than or equal to $5 \cdot 10^5$. The total length of all prefixes does not exceed 10^5 symbols.

Output

Output k lines in the output file. In each line output the number of phrases that satisfy the given request, and after a space output numbers of first t_i word phrases indexed as in the input file starting from 0, sorted out by rating. If neither of the word phrases satisfy the request, you must output 0 without a space after it.

Example

standard input	standard output
2 1 100 2 white wine 20 2 with lemon 1 wi	2 0
3 3 1 1 brisket 2 2 tooth brush 3 1 bread 10 br 1 t 2 zakaz	3 2 1 0 1 1 0

Problem E. Player

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 64 megabytes

Bob has recently bought himself a player. His player is minimalistic, it plays all the tracks in a loop, and it has only two buttons: "switch to the next track" and "switch to the random track". Bob wants to know what is the minimum expectation of keystrokes needed to switch to his favorite track.

Input

The only line of input contains three integers, separated by spaces: n — the number of tracks, m_1 — the number of the current track, m_2 — the number of Bob's favorite track. $1 \leq n \leq 10^{18}$, $1 \leq m_1 \leq n$, $1 \leq m_2 \leq n$.

Output

Output a single number — the required value. Answer will be accepted if the relative or absolute error will not exceed 10^{-6} .

Example

standard input	standard output
4 2 1	2.333333

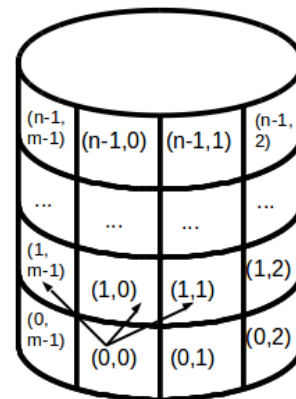
Problem F. Fort Boyard

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 megabytes

Vasya likes to watch a TV show "Fort Boyard". The participants of this show have to perform various tasks. One of the tasks enforces a participant to climb a tower under certain conditions. Let us divide a tower into nm sections, dividing it into n equal disks in height and m equal sectors. A participant may climb bottom to top, optionally moving to any adjacent sector. Also the rules specify a set of sectors that have to be visited along the way.

Formally, let the discs be numbered from 0 to $n - 1$ bottom to top. Let the sectors also be numbered from 0 to $m - 1$ counter-clockwise if viewed from above, starting from an arbitrary sector. Then each section is defined by two coordinates (x, y) where x is a disk number, and y is a sector number. A participant may move from the section with (x, y) into section $(x + 1, (y - 1) \bmod m)$ or $(x + 1, y \bmod m)$ or $(x + 1, (y + 1) \bmod m)$, where $x \bmod y$ is a remainder on division of x by y . The participant may start his way from any section with a coordinate $x = 0$ and finish it in any section with a coordinate $x = n - 1$.

Each section (x, y) contains some number of keys $a_{x,y}$. Visiting a section, a participant collects all keys that are present in it. A desire is to collect as many keys as possible and do not violate any rules of the game, visiting all the mandatory sectors. Write a program that calculates the greatest number of keys that can be collected, or determine the impossibility of satisfying game conditions.



Input

The first line contains two integers separated by a space: n, m , $1 \leq n \leq 250$, $1 \leq m \leq 250$. The next line contains m numbers separated by spaces. j -th number is equal to one if you have to visit a sector $j - 1$, zero otherwise. The following n lines contain m numbers each, separated by spaces, where the j -th number in the i -th row is $a_{i-1,j-1}$, $0 \leq a_{i-1,j-1} \leq 1\,000\,000$.

Output

Output a single line containing the greatest number of keys that can be collected, or -1 if it is impossible to satisfy the game conditions.

Example

standard input	standard output
2 3 0 0 0 1 1 2 2 1 1	4
2 3 1 1 1 1 1 2 2 1 1	-1
3 4 1 0 1 0 2 5 4 6 4 1 7 3 2 3 0 5	9