Problem A. Blackjack

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 1024 mebibytes

Blackjack is a game played with one or more decks of playing cards. The objective of each player is to have the score of the hand close to 21 without going over 21. The score is the total points of the cards in the hand. Points are assigned to cards in next way:

- Cards from 2 to 10 cost their value, i.e. 2 costs 2 points, 3 costs 3 points etc.
- Face cards (jack, queen and king) cost 10 points.
- An ace may cost 11 **or** 1 points in such a way the score gets closer to but does not exceed 21.

Some special types of hands are defined:

- A hand of more than 21 points is called **bust**. It causes a player to lose automatically.
- A hand of 21 points with exactly two cards, that is, a pair of an ace and a ten-point card (a face card or a ten) is called a **blackjack** and causes a player to win automatically.

Consider blackjack for one player (the dealer). She first deals two cards to himself. In case the dealer gets a blackjack, the dealer wins and the game ends with blackjack. Then dealer decides to take another card (hit) or to stop taking (stand) in his turn, and she plays according to the following rules:

- Hits if the summary score of the cards on her hand is 16 or less.
- Also hits if the summary score of the cards on her hand is 17 and one of aces is counted as 11.
- Stands otherwise.

Write a program that counts the score of the dealer's hand after his play for each given sequence of cards.

Input

The first line of the input contains an integer T ($1 \le T \le 100$) — number of test cases. Then T test cases follow.

Each test case consists of two lines. The first line contains two characters, which indicate the cards in the dealer's initial hand. The second line contains eight characters, which indicate the top eight cards in the pile after all players finish their plays.

A character that represents a card is one of `A`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `T`, `J`, `Q` and `K`, where `A` is an ace, `T` a ten, `J` a jack, `Q` a queen and `K` a king.

Characters in a line are delimited by a single space.

The same cards may appear up to four times in one test case. Note that it is impossible that dealer will use more than eight cards from the stack.

Output

For each test case, print "`blackjack`" if the dealer has a blackjack; "`bust`" if the dealer busts. Otherwise print the summary score of the dealer's hand.

Example

standard input	standard output
4 4 6 7 4 A J T K 5 6 A J 2 3 9 Q Q 2 3 9 T 3 8 2 3 2 J J A A 2 2 2 4 3 J 2 3 4 K	17 blackjack 21 bust

Problem B. Longest Increasing Subsequence

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

Given an increasing integer sequence $A = a_1, a_2, \cdots, a_n$ of length n whose elements are all distinct, we generate another sequence B with the following algorithm.

Algorithm 1 Sequence Generating Algorithm

function GENERATE(A)
  $B \leftarrow A$
  while true do
    $B' \leftarrow B$
    Let S be the sequence by sorting B in increasing order
    for i in $[1, \text{length of } S)$ **do**
      if $s_i + 1 \neq s_{i+1}$ **then** ▷ s_i is the i-th element in S
        Add $\lfloor \frac{s_i + s_{i+1}}{2} \rfloor$ to the end of B' ▷ $\lfloor x \rfloor$ is the largest integer not larger than x
    if $B = B'$ **then**
      break
    $B \leftarrow B'$
  return B

It is easy to prove that this algorithm will terminate and that elements of B are all distinct. Calculate the length of the longest increasing subsequence of B.

Input

There is only one test case in each test file.

The first line contains an integer n ($1 \le n \le 10^5$) indicating the length of sequence A.

The second line contains n integers $a_1, a_2, \cdots, a_n$ ($1 \le a_1 < a_2 < \cdots < a_n \le 10^{18}$) indicating the given sequence.

Output

Output one line containing one integer indicating the length of the longest increasing subsequence of B.

Example

standard input	standard output
3 1 5 20	11

Note

For the sample test case, $B = \{1, 5, 20, 3, 12, 2, 4, 8, 16, 6, 10, 14, 18, 7, 9, 11, 13, 15, 17, 19\}$. Its longest increasing subsequence is $\{1, 3, 4, 6, 7, 9, 11, 13, 15, 17, 19\}$ of length 11.

Problem C. Inverse LCS

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 mebibytes

Consider next algorithm to find *least common subsequence* of integer sequences a and b.

Lets $a = \{a_1, a_2, \ldots, a_n\}$, $b = \{b_1, b_2, \ldots, b_m\}$.

$$\begin{aligned} f(i,j) &= \begin{cases} max(f(i-1,j), f(i,j-1), f(i-1,j-1)+1), \text{if } a_i = b_j \\ max(f(i-1,j), f(i,j-1)), \text{else.} \end{cases} \\ f(i,0) &= 0 \\ f(0,j) &= 0 \end{aligned}$$

You are given the matrix f, produced by this algoritm. Your goal is to find possible initial sequences a and b for this matrix.

Note that subsequence $x = \langle x_1, x_2, \ldots, x_n \rangle$ is called *subsequence* $y = \langle y_1, y_2, \ldots, y_m \rangle$, if exists increasing sequence of indices $1 \le i_1 < i_2 < \ldots i_n \le m$ such as for any $j = 1, 2, \ldots, n$ $y_{i_j} = x_j$.

Input

First line of the input contains two integers n and m — lengths of the sequences a and b, respectively ($1 \le n, m \le 500$). Next n lines contain description of the matrix f, $f(i,j), i \in [1..n], j \in [1..m]$.

It is guaranteed that input data are correct and at least one pair of sequences a and b give matrix f as the result of work of the given algorithm.

Output

First line of the output must contain n positive integers — sequence a, second line must contain m positive integers — sequence b. If more than one answers are possible, print any of them.

Example

standard input	standard output
5 8 0 1 1 1 1 1 1 1 0 1 1 2 2 2 2 2 1 1 1 2 2 3 3 3 1 1 1 2 2 3 3 4 1 1 1 2 3 3 4 4	1 1 3 2 4 3 1 7 1 4 3 4 2

Problem D. Interesting String Problem

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	1024 mebibytes

JB loves string problems. Here is another interesting string problem created by JB.

Suppose there is a string S, JB will list all the substrings of S. For a substring x, if it occurs multiple times in S, then JB will list it multiple times. After that, JB will sort the strings in the list by lexicographic order, and if two strings are the same, they will be sorted by the positions in S where they occur.After sorting the strings, JB will concatenate the strings by order into one string T.

Now JB will ask you Q queries, each question denoted by one integer k. JB wants you to tell him, for the k^{th} character in string T, where is its position in S?

Input

The first line contains one string $S(1 \le |S| \le 5 \times 10^5)$, contains only lowercase letters.

The second line contains one integer $Q(1 \le Q \le 5 \times 10^5)$.

The following Q lines each contains one integer $k(1 \le k \le |T|)$.

Output

For each query, output one integer denotes the answer.

Examples

standard input	standard output
pbpbppb 3 1 2 3	2 4 7
potatop 3 6 30 60	6 3 4

Problem E. Card Shark

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 mebibytes

There are n decks of cards on the table, the i-th deck consists of a_i cards and the j-th card from top to bottom is $s_{i,j}$. To simplify this problem we only consider cards to be high and low. If the card is a high card then $s_{i,j} = 1$. If it is a low one then $s_{i,j} = 0$.

There are m gamblers sitting around the table waiting for a card game. You are the dealer of the game and your task is to stack all decks of cards into one big pile. You can only control the order between decks. Neither can you insert or remove a card nor can you change the order of cards within a deck.

After the pile is made the cards will be dealt to the gamblers. The i-th gambler will be given the i-th, $(m + i)$-th, $(2m + i)$-th, ..., $(km + i)$-th, ... card from top to bottom. What the other gamblers do not know is that the b-th gambler is actually your boss. To ensure the victory, all cards given to your boss must be high cards and all cards given to the other gamblers must be low cards.

Find an order to stack the decks to fulfill this requirement or report that this is impossible. If a valid answer exists, report the answer with the smallest lexicographical order.

The answer, if it exists, is obviously a permutation of n. We say permutation $P = p_1, p_2, \cdots, p_n$ is lexicographically smaller than permutation $Q = q_1, q_2, \cdots, q_n$ if there exists an integer t such that $p_i = q_i$ for all $1 \le i < t$ and $p_t < q_t$.

Input

There is only one test case in each test file.

The first line of the input contains three integers n, m and b ($1 \le n \le 2 \times 10^5$, $2 \le m \le 2 \times 10^5$, $1 \le b \le m$) indicating the number of decks, the number of gamblers and the boss.

For the following n lines, the i-th line contains a string $s_{i,1}s_{i,2}\cdots s_{i,a_i}$ ($s_{i,j} \in \{0, 1\}$, $1 \le a_i \le 10^6$) indicating the i-th deck.

It's guaranteed that

- There is at least one high card in each deck.
- The total number of cards is divisible by m.
- The total number of cards does not exceed 10^6.

Output

If a valid answer exists, output n integers $d_1, d_2, \cdots, d_n$ separated by a space indicating the answer with the smallest lexicographical order, where d_i is the index of the i-th deck in the pile from top to bottom.

If no vaild answer exists, simply output "-1" (without quotes).

Examples

standard input	standard output
5 4 3 0100010 00100 001000100 0010 0100010	2 1 3 5 4
4 2 1 010 10101 010 10101	2 1 4 3
1 5 3 001000010000100	1
2 5 3 01000 00010	-1
1 5 3 11111	-1

Problem F. Coprime Matrices

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

Given n, m, x, y, w, construct a matrix M satisfying following constraints:

1. the number of rows and columns of M are n, m respectively
2. for each integer $i\,(1 \le i \le nm)$, i appears exactly once in M
3. $M_{x,y} = w$
4. for each entry $M_{i,j}\,(1 < i < n, 1 \le j \le m)$, either $\gcd(M_{i,j}, M_{i-1,j}) = 1$ or $\gcd(M_{i,j}, M_{i+1,j}) = 1$ or both holds
5. for each entry $M_{i,j}\,(1 \le i \le n, 1 < j < m)$, either $\gcd(M_{i,j}, M_{i,j-1}) = 1$ or $\gcd(M_{i,j}, M_{i,j+1}) = 1$ or both holds

If multiple solution exist, print any one of them. If no solution, report it.

Input

Input one line containing five integers $n, m, x, y, w\,(1 \le x \le n \le 300, 1 \le y \le m \le 300, 1 \le w \le nm)$.

Output

If no solution, print “No”(without quotes) in one line.

If solution exists, print “Yes”(without quotes) in the first line. Then print n lines each containing m integers $M_{i,1}, M_{i,2}, \cdots, M_{i,m}$, denoting the answer matrix.

Example

standard input	standard output
3 3 2 1 3	Yes 4 9 2 3 5 7 8 1 6

Problem G. Quardilateral

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

You have given two points A and B such as $0 < x_a, y_a, x_b, y_b \leq 10^4$ and $(x_a - x_b) \cdot (y_a - y_b) < 0$. Build two points C and D such as:

- quadrilateral $ABCD$ is convex;
- exactly one of points C and D lies on the x-axis, and exactly one lies on the y-axis;
- perimeter of the quadrilateral must be minimum possible.
- all points A, B, C and D are pairwise distinct.

Then you should print the resulting perimeter.

Input

Input contains four integers x_a, y_a, x_b and y_b — coordinates of points A and B respectively ($0 < x_a, y_a, x_b, y_b \leq 5000$, $(x_a - x_b) \cdot (y_a - y_b) < 0$.

Output

Print the perimeter with absolute error 10^{-3} or better.

Example

standard input	standard output
3 2 1 4	10.04

Problem H. Fill the Square

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

You need fill the square $n \times n$ with integers from 1 to n^2 in next way: fill first line of square with integers from 1 to n from left to right, then fill second line of square with integers from $n + 1$ to $2n$, listed **from right to left**, then fill third line of square with integers from $2n + 1$ to $3n$ listed in normal order etc (i.e. if numeration of lines is 1-indexed, numbers in odd-indexed lines are listed from left to right, numbers in even-indexed lines are listed from right to left).

Input

Input constists of one integer n ($1 \leq n \leq 100$).

Output

Print n lines, each containing n integers — answer to the problem.

Example

standard input	standard output
5	1 2 3 4 5 10 9 8 7 6 11 12 13 14 15 20 19 18 17 16 21 22 23 24 25

Problem I. Optimal Assortment

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Potato is a toy retailer. He has n types of toys in his warehouse. Selling the i-th toy can make a profit v_i. He has conducted market research in advance. He knows that a customer buys at most one toy and customer's preference w_i for toy i will be in range $[l_i, r_i]$.

When potato offers a toy set S to the customer, the probability of a customer buying toy $i \in S$ is $\frac{w_i}{w_0+\sum_{i\in S} w_i}$ and the probability of a customer not making a purchase is $\frac{w_0}{w_0+\sum_{i\in S} w_i}$. Specifically, $w_i = 0$ indicates that the customer prefers not to buy such toy. When $w_i = 0$ for $i = 0$ and all $i \in S$, potato gains nothing. Potato wants to choose an optimal set of toys to maximize the mathematic expectation of his profit in the worst case, where w_i can be arbitrarily chosen within the ranges.

Potato is a smart guy and he can easily solve the above problems by himself. He raises a harder question. If there are two kinds of modification operations, the first modification operation will change the range of customer's preference and the second modification operation will change the profit of the toy i. Here one operation will effect all follow-up calculations. Can you quickly answer the the maximum profit after each modification?

Input

The first line contains two integers n and m ($1 \le n, m \le 2 \times 10^5$) - the number of types of toys and the number of modification operations.

The second line contains n integers $v_1, v_2, \cdots, v_n$ ($1 \le v_i \le 10^6$) - the profit of each type of toys.

The third line contains $n+1$ integers $l_0, l_1, \cdots, l_n$ - the lower bounds of customer's preference to buy toys.

The fourth line contains $n+1$ integers $r_0, r_1, \cdots, r_n$ ($0 \le l_i \le r_i \le 10^6$) - the upper bounds of customer's preference to buy toys.

The next m line contains m modification operations, which is in one of the following two types:

- 1 x y z ($0 \le x \le n, 0 \le y \le z \le 10^6$) - change range of customer's preference to buy toy x to $[y, z]$
- 2 x y ($1 \le x \le n, 1 \le y \le 10^6$) - change the profit of toy x to y

Output

Print $m + 1$ irreducible fraction (in the form of a/b, where the greatest common divisor of a and b is 1) — the initial profit and the profit after each modification.

Example

standard input	standard output
2 5 4 2 4 3 2 4 3 2 2 1 2 1 1 2 3 1 0 0 0 1 1 0 0 1 2 0 0	16/9 10/9 1/1 2/1 2/1 0/1

Problem J. Cell Tower

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Cell Tower is an interesting daily puzzle game, `https://www.andrewt.net/puzzles/cell-tower/`.

Here we consider a simplified version. You are given 8×8 square with one character in each cell and a dictionary. Please divide the square into several parts, so that each part is a **connected block** and the characters in this connected block (from top to bottom and from left to right) make up a valid word (i.e., appear in the dictionary).

Two cells A, B are called **connected pair**, if A and B directly share the same side, or there exists another cell C so that both A, C and B, C are connected pairs.

A group of cells is called **connected block** if any pair of cells in this group are **connected pairs** and the size of this group is either 3 or 4.

Input

The first 8 lines of input each contains 8 integers $Sq_{i,j} (0 \le Sq_{i,j} \le 9)$ indicating the given 8×8 square.

In the next line, there is one integer $n (1 \le n \le 11000)$ indicating the size of the dictionary.

In the next n lines, there is a string $S_i (3 \le |S_i| \le 4)$ in each line, describing the word in the dictionary. The character set of S_i is $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

Output

Please output the number of valid divisions.

Example

standard input	standard output
1 1 1 1 2 3 3 3 0 4 4 4 2 2 2 3 0 0 5 5 6 6 7 7 0 9 5 5 6 8 7 7 9 9 9 1 6 8 8 8 3 1 1 1 2 2 2 2 4 5 6 0 0 4 4 3 7 8 9 0 0 4 3 3 16 1111 2222 3333 444 0000 5555 6666 7777 8888 9999 111 333 3456 789 3478 569	2

Problem K. Triangles

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Consider integer K. We call integer A *K-interesting*, if for all integer $B \geq A$ exists exactly K distinct ingegers C such as triangle with sides A, B and C exists.

Given K, find number of K-interesting integers.

Input

Input consists of one integer K ($0 \leq K \leq 10^{18}$).

Output

Print the answer to the problem — number of K-interesting integers for a given K.

Example

standard input	standard output
1	1
2	0