Problem A. Knapsack

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

You are given n items with masses $m_1, \ldots, m_n$ and costs $c_1, \ldots, c_n$ respectively.

The knapsack can hold items with total mass up to m. You are to find the set of items having the maximal possible total cost which can be held by the knapsack.

Input

First line of the input file contains a positive integer number n not exceeding 1000 and positive integer number m not exceeding 10000.

Second line contains n positive integer numbers m_i, each of them does not exceed 100.

Third line contains n positive integer numbers c_i, each of them does not exceed 100.

Output

Output the number of items in the set in the first line.

Second line must contain numbers of these items (ranging from 1 to n).

Example

system input	system output
4 6 2 4 1 2 7 2 5 1	3 1 3 4

Problem B. Longest common subsequence

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

You are given two sequences. Find their longest common subsequence.

Input

First line of the input file contains an integer number N — length of the first sequence ($1 \le N \le 2000$). Second line contains N numbers — the first sequence. All sequence elements do not exceed 10^9 by absolute value.

Third line of the input file contains an integer number M — length of the first sequence ($1 \le M \le 2000$). Fourth line contains M numbers — the second sequence. All sequence elements do not exceed 10^9 by absolute value.

Output

Output the length of the longest common subsequence in the first line. Output the subsequence in the second line. If there are several longest common subsequnces — output any of them.

Example

system input	system output
3 1 2 3 4 2 3 1 5	2 2 3

Problem C. Levenshtein distance

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

Consider a string and a set of operations:

- Substitute one character of the string.
- Delete one character from any position of the string.
- Insert one character in any position of the string.

E.g., using first operation one can transform "ABC" to "ADC", using second — to "AC", using third — to "ACBC".

Minimal number of such operations needed to transform one string to the other is called the *Levenshtein distance*.

You are given two strings. Find the Levenshtein distance between them.

Input

Input file contains two lines each of the containid one of the given strings. Lengths of these strings do not exceed 5000 and strings consist only from capital Latin letters.

Output

Output one number — the Levenshtein distance.

Example

system input	system output
ABCDEFGH ACDEXGIH	3

Problem D. Matrix multiplication

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

Matrix product is said to be fully parenthized if one of the following holds:

- It consists of one matrix.
- It is a put in parentheses product of two fully parenthized products.

A full parenthezation is called optimal if the number of operations needed to calculate the product is minimal possible.

You are to find the optimal full parenthezation for matrix product.

Input

First line of the input file contains an integer number n — number of matrices ($1 \le n \le 400$).

Each of the following n lines contains two integer numbers a_i and b_i — number of rows and columns in the i-th matrix, respectively ($1 \le a_i, b_i \le 100$).

It is guaranteed that $b_i = a_{i+1}$ for each $1 \le i \le n-1$

Output

Output the optimal parenthezation. If there are several of them, output any.

Example

system input	system output
3 10 50 50 90 90 20	((AA)A)

Note

In this case there are two possible parenthezations: $((AA)A)$ and $(A(AA))$. The number of operations is $10 \cdot 50 \cdot 90 + 10 \cdot 90 \cdot 20 = 63000$ for the first one, and $10 \cdot 50 \cdot 20 + 50 \cdot 90 \cdot 20 = 100000$ for the second one.

Problem E. Longest subpalindrome

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

Palindrome is a string which reads the same from both directions.

Subpalindrome is a sequence of characters from the string (not necessarily consecutive) which is a palindrome. E.g., “HELOLEH” is a subpalindrome of “HTEOLFEOLEH”.

You are to find the longest subpalindrome of the given string.

Input

Input file contains a string of capital Latin letters. Its length do not exceed 2000.

Output

Output the length of the longest subpalindrome on the first line. Second line must contain the maximal subpalindrome itself. If there are several longest subpalindromes, you can output any of them.

Example

system input	system output
HTEOLFEOLEH	7 HEOLOEH

Problem F. Number of paths in acyclic graph

Input file: `countpaths.in`
Output file: `countpaths.out`
Time limit: 2 seconds
Memory limit: 256 megabytes

You are given an oriented acyclic graph. Find the number of distinct paths from vertex 1 to vertex n.

Input

First line of the input file contains two integer numbers n and m — number of vertices and edges in the graph ($2 \le n \le 10^5$, $2 \le m \le 2 \cdot 10^5$).

Each of the following m lines contains two integer numbers: numbers of vertices connected by the corresponding edge.

Output

Output the number of distinct paths from vertex 1 to vertex n taken modulo $10^9 + 7$.

Example

countpaths.in	countpaths.out
4 4 1 2 1 3 3 2 2 4	2

Problem G. Travelling salesman problem

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

You are given an non-oriented weighted graph without loops and parallel edges. You are to find the path visiting all its vertices and having the minimal weight.

Input

First line contains two integer numbers n and m — number of vertices and edges in the graph ($1 \le n \le 18$, $0 \le m \le \frac{n \cdot (n-1)}{2}$). Each of the following m lines describes one edge and contains three numbers: numbers of vertices connected by the edge and the weight of the edge ($1 \le a_i, b_i \le n$, $1 \le w_i \le 10^8$).

Output

Output one number — weight of the path. If there no such path, output -1.

Examples

system input	system output
4 6 1 2 20 1 3 42 1 4 35 2 3 30 2 4 34 3 4 12	62
4 3 1 2 1 1 3 1 1 4 1	-1

Problem H. Teams creation

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

Dr. Emmett Brown has decided to change his job and is now working as a Computer Science teacher in a high school. The Dr. Brown's class has n students. Dr. Brown wants to run a programming contest for his students. But his classroom only has k computers, so he needs to run a team contest.

Dr. thinks that the teamwork would be good for the students if the students in the team all have close levels of their skills. For each student Emmett Brown knows its skill level a_i. He has decided to create teams in such way that for any two teams there is a number x such that students in one team have skill level at most x, and in the other team all students have skill level at least x. There must be exactly k teams, each team must have at least one student, but there is no upper limit for the number of students in one team.

Help Doctor to find out how many ways are there for him to create the teams. Two ways are different if there are two students that are in the same team in one of them, and in different teams in the other. You should output the number of ways modulo $10^9 + 7$.

Input

The first line contains two integers n and k ($1 \le n, k \le 2000$) — the number of students in the class and the number of teams that must be created.

The second line contains n integers a_i ($1 \le a_i \le n$) — the skill levels of all students.

Output

Output one integer — the number of ways to create the teams modulo $10^9 + 7$.

Example

system input	system output
3 2 1 2 3	2
7 3 2 4 3 4 3 3 2	53

Problem I. Array

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

You are given an array a_i. You have to find an array b_i, such that:

- No more than k pairs (b_i, b_{i+1}) where $b_i \neq b_{i+1}$.
- The sum $\sum_{i=0}^{n} |a_i - b_i|$ is minimal.

Input

The first line of the input contains two integers n and k ($1 \le n \le 1000$, $0 \le k \le 100$) — the length of the arrays and the maximal number of pairs of consequent different elements.

The second line contains n integers a_i ($1 \le a_i \le 10^9$) — the elements of the array a.

Output

Output the minimal possible sum $\sum_{i=0}^{n} |a_i - b_i|$.

Example

system input	system output
5 2 1 2 3 4 5	2
5 0 1 2 3 4 5	6

Problem J. String Decomposition

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

For a string α and an integer n define α^n as the concatenation of n copies of α. For example, $\texttt{aab}^4 = \texttt{aabaabaabaab}$.

Each string S can be decomposed as $S = S_1^{d_1} S_2^{d_2} \dots S_k^{d_k}$. There can be several ways to make such decomposition. The weight of the decomposition is the sum $|S_1| + |S_2| + \dots + |S_k|$ where $|Z|$ is the length of the string Z.

Given S find its decomposition which has the minimal possible weight.

Input

The input file contains the string S. S contains only capital letters of the English alphabet, its length doesn't exceed 5 000.

Output

The first line of the output file must contain w — the minimal possible weight of the decomposition of S. Let k be the number of elements in the optimal decomposition. The following k lines must contain two elements each — S_i and d_i separated by a space.

If there are several optimal decompositions, describe any one.

Example

system input	system output
ABABAAABABA	5 AB 2 A 3 BA 2

Problem K. Chips

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

You have n bags of chips in a row. The i-th bag contains a_i chips. In one turn, you could choose l and r and take exactly one chips from the bags from l to r. It should be the case that each bag from l to r contains at least one chip.

Calculate the maximal number of empty bags after k turns.

Input

The first line of the input contains n and k ($1 \le n \le 2000$, $1 \le k \le 10^9$) — the number of bags and the number of turns.

Output

The sole line of the output should contain the answer on the problem.

Example

system input	system output
6 5 1 5 3 2 1 4	5

Problem L. Congress

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

One mathematician and one philisopher from each of n countries attend the congress. The congress room contains $2n$ chairs in a row. The chairs could be of three types: on a chair of the first type only the mathematician could seat, on a chair of the second type only the philisopher could seat, on a chair of the third type could seat anybody.

Also, the arrangement of the seats should satisfy two following conditions:

- No mathematician should be a neighbour of two philosophers and no philisopher should be a neighbour of two mathematicians.
- The mathematician and philosopher of the same country could not be neighbours.

Your task is to calculate the total number of correct arrangements modulo $10^9 + 7$.

Input

The first line of the input contains t — the number of tests in the input.

Description of each test consists of two lines. The first line contains n — the number of contries. The second line contains $2n$ integers from 1 to 3. The integer 1 means that the chair is for the mathematician, the integer 2 means that the chair is for the philosopher and the integer 3 means that the chair is for anybody.

Output

Output for each test the total number of correct arrangements modulo $10^9 + 7$ on a separate line.

Example

system input	system output
2 2 3 3 3 3 2 1 1 1 1	8 0