Codeforces Round #302 (Div. 1)

A. Writing Code

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Programmers working on a large project have just received a task to write exactly m lines of code. There are n programmers working on a project, the i-th of them makes exactly a_i bugs in every line of code that he writes.

Let's call a sequence of non-negative integers $v_1, v_2, \ldots, v_n$ a *plan*, if $v_1 + v_2 + \ldots + v_n = m$. The programmers follow the plan like that: in the beginning the first programmer writes the first v_1 lines of the given task, then the second programmer writes v_2 more lines of the given task, and so on. In the end, the last programmer writes the remaining lines of the code. Let's call a plan *good*, if all the written lines of the task contain at most b bugs in total.

Your task is to determine how many distinct *good* plans are there. As the number of plans can be large, print the remainder of this number modulo given positive integer mod.

Input

The first line contains four integers n, m, b, mod ($1 \le n, m \le 500$, $0 \le b \le 500$; $1 \le mod \le 10^9 + 7$) — the number of programmers, the number of lines of code in the task, the maximum total number of bugs respectively and the modulo you should use when printing the answer.

The next line contains n space-separated integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 500$) — the number of bugs per line for each programmer.

Output

Print a single integer — the answer to the problem modulo mod.

Examples

input
3 3 3 100 1 1 1
output
10

input
3 6 5 1000000007 1 2 3
output
0

input
3 5 6 11 1 2 1
output
0

B. Destroying Roads

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In some country there are exactly n cities and m bidirectional roads connecting the cities. Cities are numbered with integers from 1 to n. If cities a and b are connected by a road, then in an hour you can go along this road either from city a to city b, or from city b to city a. The road network is such that from any city you can get to any other one by moving along the roads.

You want to destroy the largest possible number of roads in the country so that the remaining roads would allow you to get from city s_1 to city t_1 in at most l_1 hours and get from city s_2 to city t_2 in at most l_2 hours.

Determine what maximum number of roads you need to destroy in order to meet the condition of your plan. If it is impossible to reach the desired result, print -1.

Input

The first line contains two integers n, m ($1 \le n \le 3000$, $n - 1 \le m \le \min\{3000, \frac{n(n-1)}{2}\}$) — the number of cities and roads in the country, respectively.

Next m lines contain the descriptions of the roads as pairs of integers a_i, b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$). It is guaranteed that the roads that are given in the description can transport you from any city to any other one. It is guaranteed that each pair of cities has at most one road between them.

The last two lines contains three integers each, s_1, t_1, l_1 and s_2, t_2, l_2, respectively ($1 \le s_i, t_i \le n$, $0 \le l_i \le n$).

Output

Print a single number — the answer to the problem. If the it is impossible to meet the conditions, print -1.

Examples

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

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

input
5 4 1 2 2 3 3 4 4 5 1 3 2 3 5 1
output
-1

C. Remembering Strings

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have multiset of n strings of the same length, consisting of lowercase English letters. We will say that those strings are easy to remember if for each string there is some position i and some letter c of the English alphabet, such that this string is the only string in the multiset that has letter c in position i.

For example, a multiset of strings {"abc", "aba", "adc", "ada"} are not easy to remember. And multiset {"abc", "ada", "ssa"} is easy to remember because:

- the first string is the only string that has character c in position 3;
- the second string is the only string that has character d in position 2;
- the third string is the only string that has character s in position 2.

You want to change your multiset a little so that it is easy to remember. For a_{ij} coins, you can change character in the j-th position of the i-th string into any other lowercase letter of the English alphabet. Find what is the minimum sum you should pay in order to make the multiset of strings easy to remember.

Input

The first line contains two integers n, m ($1 \le n, m \le 20$) — the number of strings in the multiset and the length of the strings respectively. Next n lines contain the strings of the multiset, consisting only of lowercase English letters, each string's length is m.

Next n lines contain m integers each, the i-th of them contains integers $a_{i1}, a_{i2}, \ldots, a_{im}$ ($0 \le a_{ij} \le 10^6$).

Output

Print a single number — the answer to the problem.

Examples

input
4 5 abcde abcde abcde abcde 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
output
3

input
4 3 abc aba adc ada 10 10 10 10 1 10 10 10 10 10 1 10
output
2

input
3 3 abc ada ssa 1 1 1 1 1 1 1 1 1
output
0

D. Road Improvement

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The country has n cities and $n - 1$ bidirectional roads, it is possible to get from every city to any other one if you move only along the roads. The cities are numbered with integers from 1 to n inclusive.

All the roads are initially bad, but the government wants to improve the state of some roads. We will assume that the citizens are happy about road improvement if the path from the capital located in city x to any other city contains at most one bad road.

Your task is — for every possible x determine the number of ways of improving the quality of some roads in order to meet the citizens' condition. As those values can be rather large, you need to print each value modulo $1\,000\,000\,007$ ($10^9 + 7$).

Input

The first line of the input contains a single integer n ($2 \le n \le 2 \cdot 10^5$) — the number of cities in the country. Next line contains $n - 1$ positive integers $p_2, p_3, p_4, \ldots, p_n$ ($1 \le p_i \le i - 1$) — the description of the roads in the country. Number p_i means that the country has a road connecting city p_i and city i.

Output

Print n integers $a_1, a_2, \ldots, a_n$, where a_i is the sought number of ways to improve the quality of the roads modulo $1\,000\,000\,007$ ($10^9 + 7$), if the capital of the country is at city number i.

Examples

input
3 1 1
output
4 3 3

input
5 1 2 3 4
output
5 8 9 8 5

E. Listening to Music

time limit per test: 7 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

Please note that the memory limit differs from the standard.

You really love to listen to music. During the each of next s days you will listen to exactly m songs from the playlist that consists of exactly n songs. Let's number the songs from the playlist with numbers from 1 to n, inclusive. The quality of song number i is a_i.

On the i-th day you choose some integer v ($l_i \le v \le r_i$) and listen to songs number $v, v + 1, \ldots, v + m - 1$. On the i-th day listening to one song with quality less than q_i increases your displeasure by exactly one.

Determine what minimum displeasure you can get on each of the s next days.

Input

The first line contains two positive integers n, m ($1 \le m \le n \le 2 \cdot 10^5$). The second line contains n positive integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i < 2^{30}$) — the description of songs from the playlist.

The next line contains a single number s ($1 \le s \le 2 \cdot 10^5$) — the number of days that you consider.

The next s lines contain three integers each l_i, r_i, x_i ($1 \le l_i \le r_i \le n - m + 1$; $0 \le x_i < 2^{30}$) — the description of the parameters for the i-th day. In order to calculate value q_i, you need to use formula: $q_i = x_i \oplus ans_{i-1}$, where ans_i is the answer to the problem for day i. Assume that $ans_0 = 0$.

Output

Print exactly s integers $ans_1, ans_2, \ldots, ans_s$, where ans_i is the minimum displeasure that you can get on day i.

Examples

input
5 3 1 2 1 2 3 5 1 1 2 1 3 2 1 3 3 1 3 5 1 3 1

output
2 0 2 3 1