Problem 1. First-Come-First-Serve with deadline

(Time Limit: 3 seconds)

Problem Description

There are N jobs to be processed, where $N < 100{,}000$, and these jobs must be completed before a given deadline. The CEO wants to buy more machines to meet the deadline requirement. A job cannot be partitioned and must be processed in one machine. The i-th job takes $t(i)$ time no matter which machine you choose to process it. An important principle is first-come-first-serve. Suppose that the jobs have been sorted by their incoming time. For $i < j$, job j cannot start before the time that job i starts. Given a deadline, write a program to compute the minimum number of machines to complete all the jobs in time.

Input Format

The input of a test case consists of two lines. The first line contains two integers N and D, which are the number of jobs and the deadline, respectively. The second line consists of N integers, which are $t(0), t(1), \ldots, t(N-1)$. Any two consecutive numbers in the same line are separated by a space. You can assume that $1 \leq N \leq 100000$ and all the input and output numbers in this problem are 32-bit signed integers.

Output Format

Output the minimum number of machines to complete all the jobs in time in one line. If it is impossible, output -1.

Example

Sample Input:	**Sample Output:**
5 8 2 3 1 5 4	2
4 5 5 5 2 6 4	-1

Problem 2. Problem Hill Subsequences

(Time Limit: 3 seconds)

Problem Description

A sequence (a_1, a_2, … a_k) is a hill sequence if the following conditions are satisfied.

1. k>2
2. There exists an integer t such that $1<t<k$, $a_1<a_2<\ldots<a_t$ and $a_t>a_{t+1}>\ldots>a_k$.

For examples, (1,2,3,2,1) and (1,2,3,1) are hill sequences, but neither of (1,2,2,1) and (1,2,3,1,2) is a hill sequence.

A hill subsequence (b_1, …, b_k) of a sequence (a_1, a_2, … a_n) is a hill sequence induced by a monotonically increasing sequence (i_1, …, i_k) of indices where b_j is the i_j-th term of (a_1, a_2, … a_n). Frank wonders how many increasing sequences of indices inducing a hill subsequence are there. For examples, there are two hill subsequences of (1,2,2,1), and they are induced by (1,2,4) and (1,3,4) respectively

Please write a program to compute the answer for Frank. Output the answer modulo 10^9+7.

Technical Specification

- T, the number of test cases, is at most 25.
- n, , the length of the sequence, is at most 10^5.
- a_1, a_2, … a_n are non-negative integers no more than 10^9.

Input Format

The first line of the input file contains an integer T indicating the number of test cases. Each test case contains 2 lines. The first line contains a positive integer n indicating the length of the sequence a_1, a_2, … a_n. The second line contains n non-negative integers a_1, a_2, … a_n separated by spaces.

Output Format

For each test case, please output the number of increasing sequences of indices inducing a hill subsequence modulo 10^9+7.

Example

Sample Input:	Sample Output:
2 4 1 2 2 1 5 1 2 3 2 1	2 11

Problem 3. Random Number Game

(Time Limit: 3 seconds)

Problem Description

A player is engaged in playing random number games. A random number game of size K is played with the following rule. In each round, the player draws an integer from the interval $[1, K]$ by using a random number generator. Assume that the game stops after N-round playing. Then, the game trace is an integer sequence of length N. Note that the same integer may occur many times in the trace. We say that a game trace is a winning sequence for the player if for any integer x appearing in the sequence, the first occurrence of x-1 in the sequence appears before the last occurrence of x. Formally, if the last position of x in the sequence is i_x and the first position of x-1 in the sequence is j_{x-1}, then $j_{x-1} < i_x$. For example, [2,3,1,2] is a winning sequence of length 4 for a game of size 3 because the first (unique) occurrence of 1 appears before the last occurrence of 2 and the first occurrence of 2 appears before the last (unique) occurrence of 3. To assess the chance of winning, the player would like to compute the number of winning sequences of length N for a random number game of size K.

Input Format

The input contains several test cases; each test case is formatted as follows. The first line contains two integers N and Q, representing respectively the length of the game trace and the number of queries to answer ($1 \leq N \leq 5000$ and $1 \leq Q \leq 1000$). The second line contains Q integers $K_1, K_2, \ldots, K_Q$, indicating that the size of the game is K_i in the i-th query ($1 \leq K_i \leq 10^9$ for $i = 1, 2, \ldots, Q$).

Output Format

For each test case, output a line with Q integers, such that the i-th integer represents the number of winning sequences of length N for a random number game of size K_i. Since this number can be very large, output the remainder of dividing it by $10^9 + 7$.

Example

Sample Input:	Sample Output:
1 1 1 6 4 2 5 8 10 1000 3 120 25 200	1 58 719 720 720 928381737 887265570 454636459

Problem 4. Overlap Computation

(Time Limit: 1 second)

Problem Description

High-throughput DNA sequencing has become the major platform for deciphering the genetic secrets in all lives. DNA sequencing randomly shears the genome into pieces and sequences each piece in parallel. A huge amount of sequences will be generated, but the original genome is still not available. In order to reconstruct the original genome, efficient overlap computation among these sequences is required. However, because DNA sequence may contain errors, including insertions, deletions, and mismatches, the overlap computation must tolerate these errors. In order to measure the degree of overlap, we define a scoring function: match=+2, mismatch=-3, and insertion/deletion=-1. For each consecutive run of insertions/deletions, an additional gap opening cost -2 will be charged. Given two sequences, you are asked to compute the overlap with maximum score. The two flanking non-overlapping subsequences will not be charged for score. Note that the overlap could be left-shifted, right-shifted, or entirely contained depending on which one provides the maximum score.

```
TATACGTTGCG              ACG--TGCGTTA    ACGTGCGTTA
   ACG-TGAGTTA        TTTACGTTTGCG          TGCG
2*3-2-1+2*3-3=6       2*7-2-2*2=10          2*4=8
```

Given a scoring function and two sequences, you are asked to compute the maximum overlap score.

Technical Specification

- The length of each sequence ranges from 10 to 1000.
- The scoring function is match: +2, mismatch: -3, insertion/deletion: -1, gap opening: -2.
- The number of test cases is at most 10.

Input Format

The first line of the input file contains an integer indicating the number of test cases. Each test case consists of two lines, where the first and second line stores the first and second sequences.

Output Format

For each test case, output the maximum overlap score in each line.

Example

Sample Input:	Sample Output:
3 TATACGTTGCG ACGTGAGTTA ACGTGCGTTA TTTACGTTTGCG ACGTGCGTTA TGCG	6 10 8

Problem 5. Seen or not Seen?

(Time Limit: 3 seconds)

Problem Description

Recently, Peter, a famous architect, encounters a problem that is extremely difficult to solve by himself. Therefore, Peter asks for your help. The problem is described below.

In this problem, n line segments on the plane are given. There is no intersection and overlap between any pair of line segments. Each line segment has a color. Peter wants to know the total length of each color that can be seen from the north.

Figure 1(a) shows an example, in which there are five line segments L_1, L_2, …, L_5 and their colors are, respectively, 1, 1, 2, 2, 3. Figure 1(b) shows, in bold lines, the portions of line segments that can be viewed from the north. In this example, L_1 and L_2 are not sheltered by any line segments, so the whole L_1 and L_2, each of length 2, are visible; L_3 is sheltered by L_1 and its visible portion is of length $\sqrt{2} + 2\sqrt{2}$; similarly, the visible portion of L_4 is of length $\sqrt{2} + 2\sqrt{2}$; and L_5 is sheltered by L_3 and L_4 and its visible portion is of length 1. As a result, the total visible lengths of colors 1, 2, 3 are, respectively, 2 + 2 = 4, $3\sqrt{2} + 3\sqrt{2} = 6\sqrt{2}$, and 1.

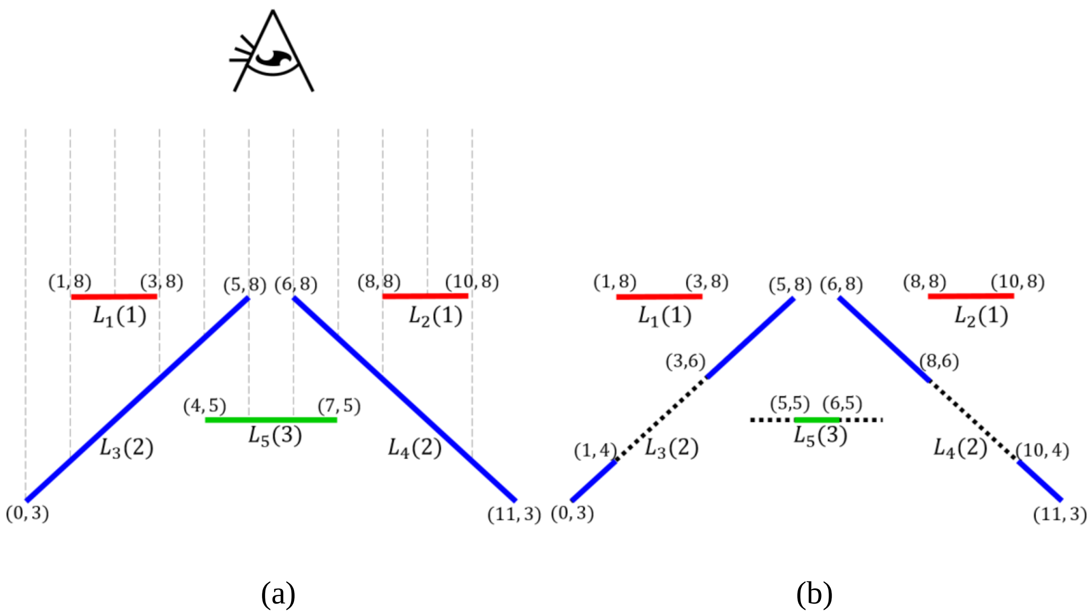

Figure 1. An example.

Technical Specification

- There are at most 10 test cases.
- The number, n, of line segments is an integer between 1 and 3×10^5.
- Coordinates of the endpoints of line segments are integers between 0 and 10^8.
- The number of different colors is an integer between 1 and 10^4.

Input Format

The first line of the input file contains an integer indicating the number of test cases. Each test case begins with a line containing two integers n and r, where $1 \le n \le 3\times10^5$ and $1 \le r \le 10^4$. The integer n indicates the number of line segments; and the integer r is the number of different colors, represented by integers 1, 2, ..., r. Then, n lines follow, each containing five integers x_1, y_1, x_2, y_2, c, indicating that there is a line segment of color c, whose endpoints are (x_1, y_1) and (x_2, y_2), where $0 \le x_1 < x_2 \le 10^8$, $0 \le y_1, y_2 \le 10^8$, and $1 \le c \le r$.

Output Format

For each test case, print a line containing three real numbers indicating the smallest, the median, and the maximum of $m_1, m_2, ..., m_r$, where m_c is the total visible length of color c, $1 \le c \le r$. These values must be exact to two digits to the right of the decimal point. If there are k numbers, here the median is uniquely defined to be the $\lfloor(k + 1)/2\rfloor$-th smallest number. For example, the median of (10, 5, 10, 6, 7) is 7, and the median of (2, 10, 9, 5) is 5.

Example

Sample Input:	Sample Output:
2 5 3 1 8 3 8 1 8 8 10 8 1 0 3 5 8 2 6 8 11 3 2 4 5 7 5 3 3 10 0 3 9 3 1 5 1 11 1 3 7 2 16 2 2	1.00 4.00 8.49 0.00 0.00 9.00