Problem L. LIS on Grid

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given a grid with n rows and m columns. n rows are numbered from 1 to n from top to bottom, and m columns are numbered from 1 to m from left to right. Let's denote the cell at the intersection of the i-th row and the j-th column by (i, j).

You have to color some of these cells in black. More precisely, in i-th column you should color exactly a_i cells in black.

After you color cells of your choice, your penalty will be calculated as the length of the longest increasing subsequence of black cells. More precisely, it will be equal to the largest k, for which there exists a sequence of cells $(r_1, c_1), (r_2, c_2), \ldots, (r_k, c_k)$, such that:

- $1 \le r_1 < r_2 < \ldots < r_k \le n$
- $1 \le c_1 < c_2 < \ldots < c_k \le m$
- All cells (r_i, c_i) are black

Find the smallest possible penalty you can get.

Input

The first line contains a single integer t ($1 \le t \le 10^5$) — the number of test cases. The description of test cases follows.

The first line of each test case contains two integers n, m ($1 \le n, m \le 2 \cdot 10^5, m \cdot n \le 2 \cdot 10^5$) — the numbers of rows and columns correspondingly.

The second line of each test case contains m integers $a_1, a_2, \ldots, a_m$ ($1 \le a_i \le n$), indicating that you have to color exactly a_i cells in column i.

It is guaranteed that the sum of $m \cdot n$ over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, in the first line output a single integer — the smallest penalty you can achieve.

In the next n lines, output n strings. i-th string should have length m and consist only of characters `.` and `#`. If j-th character of i-th string is `#`, cell (i, j) is colored black, otherwise it's colored white.

Example

standard input	standard output
4 2 4 1 1 1 1 3 3 3 3 3 4 4 4 3 2 1 4 5 2 3 4 3 2	1 .... #### 3 ### ### ### 2 ###. #... #### ##.. 2 ..### .#### ####. ###..