

Southeastern European Regional Programming Contest
Bucharest, Romania – Vinnytsya, Ukraine
October 21, 2017

Problem B
Bricks

Input File: B.in
Output File: standard output
Time Limit: 0.*2* seconds (C/C++)
Memory Limit: *512* megabytes

Let us consider a zone with **N** boxes, initially empty, numbered from **1** to **N**. We have **M** events given in chronological order. An event is described by a number "**p**": a brick falls at position **p**. If the box at that position is empty, the brick has to stay there. Otherwise, let's consider the full interval of consecutive occupied boxes which contains the box labeled **p**. We have two options: we can put the new brick either on the left side or the right side of the interval (if they exists). The left side of an interval **[a,b]** is position **a - 1**, while the right side is position **b + 1**.

Explanation:
We are given a binary string of length **N**, where **0** denotes a free box and **1** denotes an occupied box: **001110111100**. If a brick falls at position **8** (or any other position inside the interval **[7,10]**), we can place this new brick either at position **6**, or at position **11**. If it falls at position **2** (which is unoccupied), the brick must stay there.

Task:
Given **N**, **M** and a set of **M** events (in chronological order), determine the number of distinct configurations in which we can place the **M** bricks at the **N** possible positions (boxes). The answer should be computed **modulo 1.000.000.007**. The bricks are unordered, so their order does not matter. If we consider the binary interpretation (from the explanation), you are asked to find out how many distinct binary string you can form.

Constraints:
- $1 \leq \mathbf{M} \leq 100.000$
- $1 \leq \mathbf{M} \leq \mathbf{N} \leq 1.000.000$
- $1 \leq \mathbf{p} \leq \mathbf{N}$

Input
- First line: **N**, **M**
- Second line: **M** numbers denoting the **M** events (the values for **p**)

Output
- One number denoting the number of distinct configurations you can obtain **modulo 1.000.000.007**.

Sample input	Sample output
5 4 2 2 4 4	3