Problem I
Count DFS Tree

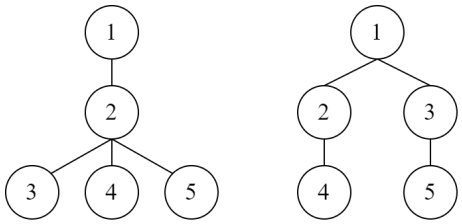
You are currently studying a tree traversal algorithm called the Depth First Search (DFS). Suppose you have a rooted tree of n nodes (numbered from 1 to n) with a depth of K (numbered from 1 to K). The root (the node at depth 1) is located at node 1. **All leaves are located at the same depth**, that is, at depth K. Node i has an array of children nodes c_i, which could be empty if i is a leaf node. The pseudocode of the algorithm is presented as follows.

```
DFS(u, depth):
  let res be an empty array
  append depth to res

  for each v in c[u]:
    let D be an array initialized with DFS(v, depth + 1)
    for each x in D:
      append x to res

  return res
```

Consider the trees in the following illustration. The return values of `DFS(1, 1)` for the tree on the left and the tree on the right are $[1, 2, 3, 3, 3]$ and $[1, 2, 3, 2, 3]$, respectively.



Denote $f_K(n)$ as the number of **distinct** return values of `DFS(1, 1)` across all trees consisting of n nodes and all leaves are located in depth K. You are given M integers: $A_1, A_2, \ldots, A_M$. Determine the value of $f_K(A_1) \times f_K(A_2) \times \cdots \times f_K(A_M)$. As the answer can be very large, find the answer modulo $998\,244\,353$.

Input

The first line consists of two integers K M ($1 \leq K, M \leq 100\,000$).

The following line consists of M integers A_i ($K \leq A_i \leq 200\,000$).

Output

Output a single integer representing the value of $f_K(A_1) \times f_K(A_2) \times \cdots \times f_K(A_M)$ modulo $998\,244\,353$.

Sample Input #1

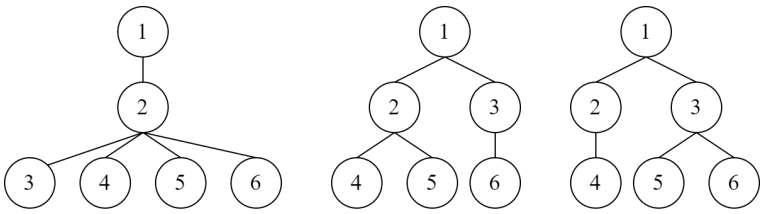
```
3 2
5 6
```

Sample Output #1

```
6
```

Explanation for the sample input/output #1

The value of $f_3(5)$ and $f_3(6)$ are 2 and 3, respectively. The illustration on the description shows the trees of 5 nodes that give distinct return values of `DFS(1, 1)`. The following illustration is for the trees of 6 nodes.



Sample Input #2

```
100000 1
200000
```

Sample Output #2

```
269130693
```