

Spring Carnival (colors)


The Spring Carnival has arrived and N tourists have reached to the event. We know that all the tourists must wear a hat which is in one of the K colors.

The carnival organizers have given each tourist a hat and asked them to sit in a line to take a group picture. However, the picture was lost, but fortunately an observer recorded for each person how many people in front of them had a hat of the same color as the hat they were wearing.

For example, if the array is $[0, 1, 0, 1, 2, 3]$, the second person had one person in front of them with the same hat color, while the fifth person had two people with the same hat color in front of them.

Assuming that the carnival organizers have an unlimited number of hats of each of the K colors, find out how many different ways exist which could have been used to assign hats to people such that the array the observer recorded would be the same as the array we would record for that given configuration?

Since the answer can be big, print it modulo $10^9 + 7$.

 Among the attachments of this task you may find a template file `colors.*` with a sample incomplete implementation.


Input

The input will contain on the first line N and K .

The next line will contain the array V .

Output

The output will contain a single number, representing the answer modulo $10^9 + 7$.





 The *modulo* operation ($a \bmod m$) can be written in C/C++/Python as `(a % m)` and in Pascal as `(a mod m)`. To avoid the *integer overflow* error, remember to reduce all partial results through the modulus, and not just the final result!
Notice that if $x < 10^9 + 7$, then $2x$ fits into a C/C++ `int` and Pascal `longint`.

Constraints

- $1 \leq N \leq 200\,000$.
- $1 \leq K \leq 100\,000\,000$.
- $0 \leq V_i \leq i$, for $i = 0 \dots N - 1$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 1** (0 points) Examples.

- **Subtask 2** (26 points) $N, K \leq 8$.

- **Subtask 3** (35 points) $N, K \leq 1000$.

- **Subtask 4** (39 points) No additional limitations.


Examples

input	output
6 4 0 1 0 1 2 3	24
4 3 0 1 0 0	6

Explanations

For the second sample test case, the arrays we can generate are:

- [1, 1, 2, 3]
- [1, 1, 3, 2]
- [2, 2, 1, 3]
- [2, 2, 3, 1]
- [3, 3, 1, 2]
- [3, 3, 2, 1]