**Minimum number of jumps**

Given an array of **N** integers **arr[]** where each element represents the max length of the jump that can be made forward from that element. Find the minimum number of jumps to reach the end of the array (starting from the first element). If an element is **0**, then you cannot move through that element.  
  
**Note:**Return -1 if you can't reach the end of the array.

**Example 1:**

**Input:**

N = 11

arr[] = {1, 3, 5, 8, 9, 2, 6, 7, 6, 8, 9}

**Output:** 3

**Explanation:**

First jump from 1st element to 2nd

element with value 3. Now, from here

we jump to 5th element with value 9,

and from here we will jump to the last.

**Example 2:**

**Input :**

N = 6

arr = {1, 4, 3, 2, 6, 7}

**Output:** 2

**Explanation:**

First we jump from the 1st to 2nd element

and then jump to the last element.

**Your task:**  
You don't need to read input or print anything. Your task is to complete function **minJumps()** which takes the array **arr** and it's size **N** as input parameters and returns the minimum number of jumps. If not possible return -1.

**Expected Time Complexity:**O(N)  
**Expected Space Complexity:**O(1)

**Constraints:**  
1 ≤ N ≤ 107  
0 ≤ arri ≤ 107