Instructions: Name your file `hw1.py` and submit on CCLE. Add comments to each function.

- **Problem 1**:
  Write a function `largerIndex(c)` that takes as input a list `c` of numbers, and returns a new list `k`, such that `k[i]=1` if `c[i]>i`, `k[i] = 0` if `c[i]=i`, `k[i] = -1` if `c[i]<i`.

- **Problem 2**:
  Write a function `squareUpTo(n)` that takes as input a positive integer n, and returns a list of all the square numbers up to (and possibly including) `n`.

- **Problem 3**:
  Write a function `flip1in3()` that uses only fair coins to generate a biased coin with success probability 1/3.

- **Problem 4**:
  Write a function `duplicates(c)` that takes as input a list `c` of integers, where `1 <= c[i] <= n` (n is size of list). Some elements appear twice and others appear once. The function outputs all the elements as a list that appear twice in the list `c`. The elements in the output should preserve the original order.

- **Problem 5**:
  Write a function `longestpath(d)` that finds the length of a longest path, $(a : b) \to (b : c) \to \cdots$, in a dictionary `d`. It counts each pointer from a key to a value as one step. For example, the path $(a : b) \to (b : c)$ has length 2. To avoid cycles, we do not allow any key to appear more than once in a path (as a key).