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`.

  Test cases:

  ```
  l1 = [1,2,0,4,2,1,40,-5]
  l2 = [0,3,2,1,32,3,4,0]
  largerIndex(l1) should return [1, 1, -1, 1, -1, -1, 1, -1].
  largerIndex(l2) should return [0, 1, 0, -1, 1, -1, -1, -1].
  ```

- **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`.

  Test cases:

  ```
  squareUpTo(10) should return [0, 1, 4, 9].
  squareUpTo(100) should return [0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100].
  ```

- **Problem 3**:
  Write a function `flip1in3()` that uses only "fair coins" to generate a "biased coin" with success probability 1/3. That is, this function returns `False` with probability 2/3 and returns `True` with probability 1/3. To simulate a "fair coin", use `random.randint(0,1)`.

- **Problem 4**:
  Write a function `duplicates(c)` that takes as input a list `c` of integers. 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.

  Test cases:

  ```
  l3 = [1,2,5,3,6,2,4,5]
  l4 = [1,3,5,5,1,4,3]
  duplicates(l3) should return [2,5].
  duplicates(l4) should return [1,3,5].
  ```