Due date: Friday, April 26

Submission Instructions:

- Each group must submit one solution to this assignment in class on the due date.
- Assignments must be neatly written or typed. If the grader cannot (easily) read an answer, she will give it a 0.
- Each group must submit only one solution. If multiple solutions are handed in, the grader will randomly discard all but one of them.
- Your submission must have the name of every group member clearly marked. If your name is not on it, you do not get credit.
- All arguments should be clear, concise, and of course correct. You will lose points for poor writing, bad grammar, etc...
- Use of the Internet for solving these problems is strictly forbidden, and will be treated as an honor code violation.
- Inter-group work: Members of different groups may discuss the questions. However:
  - With your submission you must provide a list of all non-group members with whom any member of your group discussed the assignment.
  - No written material may leave the inter-group discussion. If you talk with someone about the assignment, you must throw away and written notes at the end of the discussion.
Problem 1 (20 points):

We wish to implement a dictionary by using direct addressing on a huge array. At the start, the array entries may contain garbage, and initializing the entire array is impractical because of its size. Describe a scheme for implementing a direct-address dictionary on a huge array. Each stored object should use $O(1)$ space; the operations SEARCH, INSERT, and DELETE should take $O(1)$ time each; and initializing the data structure should take $O(1)$ time. (Hint: Use an additional array, treated somewhat like a stack whose size is the number of keys actually stored in the dictionary, to help determine whether a given entry in the huge array is valid or not.)
Problem 2 (20 points): Demonstrate what happens when we insert the keys 5; 28; 19; 15; 20; 33; 12; 17; 10 into a hash table with collisions resolved by chaining. Let the table have 9 slots, and let the hash function be \( h = k \mod 9 \). Show the solution by showing which slot each key goes to, and what order the items are in the linked list.
Problem 3 (20 points): Consider inserting the keys 10, 22, 31, 4, 15, 28, 17, 88, 59 into a hash table of length $m = 11$ using open addressing with the auxiliary hash function $h_0(k) = k$. Illustrate the result of inserting these keys using linear probing, using quadratic probing with $c_1 = 1$ and $c_2 = 3$, and using double hashing with $h_1(k) = k$ and $h_2(k) = 1 + (k \mod (m - 1))$. 