Prepare your answers to the following questions in a plain text file or <u>typed</u> MS Word document. Submit your file to the Curator system by the posted deadline for this assignment. No late submissions will be accepted.

1. Consider a hash table consisting of $M = 11$ slots, and suppose nonnegative integer key values are hashed into the table using the hash function h1():

```
int h1 (int key) {
  int x = (key + 7) * (key + 7);
  x = x / 16;
  x = x + key;
  x = x % 11;
  return x;
}
```

a) [10 points] Suppose that collisions are resolved by using linear probing. The integer key values listed below are to be inserted, in the order given. Show the home slot (the slot to which the key hashes, before any probing), the probe sequence (if any) for each key, and the final contents of the hash table after the following key values have been inserted in the given order:

Key Value	Home Slot	Probe Sequence
43		
23		
1		
0		
15		
31		
4		
7		
11		
3		

Final Hash Table:

Slot	0	1	2	3	4	5	6	7	8	9	10
Contents											

b)   [10 points] Suppose that collisions are resolved by using quadratic probing, with the probe function:

$$\left(k^2 + k\right)/2$$

The integer key values listed below are to be inserted, in the order given. Show the home slot (the slot to which the key hashes, before any probing), the probe sequence (if any) for each key, and the final contents of the hash table after the following key values have been inserted in the given order:

Key Value	Home Slot	Probe Sequence
43		
23		
1		
0		
15		
31		
4		
7		
11		
3		

Final Hash Table:

Slot	0	1	2	3	4	5	6	7	8	9	10
Contents											

c)	[10 points] Suppose that collisions are resolved by using double hashing (see the course notes), with the secondary hash function Reverse(key), which reverses the digits of the key and returns that value; for example, Reverse(7823) = 3287.

The integer key values listed below are to be inserted, in the order given. Show the home slot (the slot to which the key hashes, before any probing), the probe sequence (if any) for each key, and the final contents of the hash table after the following key values have been inserted in the given order:

Key Value	Home Slot	Probe Sequence
43		
23		
1		
0		
15		
31		
4		
7		
11		
3		

Final Hash Table:

Slot	0	1	2	3	4	5	6	7	8	9	10
Contents											

2. Consider implementing a hash table for an application in which we will build an initial hash table by inserting a substantial collection of records. After this, we expect that the number of insertions and the number of deletions performed to be roughly the same, although there may be long runs of consecutive insertions or consecutive deletions. Furthermore, the table will use a probe strategy to resolve any collisions that occur during insertion, and therefore we will "tombstone" cells from which a record has been deleted.

   a) [15 points] If we implement the hash table described above, then when we search for a record, we cannot conclude the record is not in the table until we have found an empty cell in the table, not just a tombstone. (We will ensure that the table never reaches the state that there are no empty cells.) Explain carefully why the search cannot stop when a tombstone is encountered.

   But, as more and more deletions occur, the number of tombstones will affect the performance of the hash table, since those cells are essentially locked into an unusable state. Therefore, we could adopt the following strategy on insertion. If we pass a tombstoned cell during an insertion search, we will remember the first such cell we have seen, and once the insertion search concludes at an empty cell, we will place the new record into that tombstoned cell.

   b) [10 points] If we implement insertion as described above, would searches for absent records (as described in part a) still need to proceed until an empty cell is found? Justify your conclusion carefully.

   c) [10 points] If we implemented insertion as described above (recycling tombstones), one advantage is that the table will not fill up as quickly. Describe another potential advantage of recycling tombstones.

3. Suppose you are implementing a hash table and are trying to choose between using a probing strategy and using chaining (each slot is actually a linked list that can hold as many records as needed). You will use the same hash function and hash table size no matter which strategy you select.

   a) [10 points] A *primary collision* occurs when two records map to the same home slot in the table. Will the number of primary collisions be lower if you choose probing or if you choose chaining? Justify your conclusion carefully.

   b) [10 points] One objection to using chaining is that you'll have to perform a linear traversal of the linked list to find a record, and linear traversals are slow. Considering that, and assuming that a small but significant number of primary collisions will occur, would searching be more efficient if you chose <u>linear probing</u> instead of chaining? Justify your conclusion carefully.

   c) [10 points] Reconsider part b under the assumption that you'd choose quadratic probing as an alternative to chaining.