You will submit your solution to this assignment to the Curator System (as `HW4`). Your solution must be either a plain text file (e.g., NotePad) or a MS Word document; submissions in other formats will not be graded.

Except as noted, credit will only be given if you show relevant work.

---

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 + 5) * (key + 5);
  x = x / 16;
  x = x + key;
  x = x % 11;
  return x;
}
```

   a) [20 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) [20 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) [20 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. [20 points] A pseudo-random generator is a software function that, when called N times, returns a sequence of N integer values $\{V_1, V_2, V_3, \ldots, V_N\}$ that have certain nice statistical properties, like being uniformly distributed over some range. A pseudo-random generator is "primed" by supplying an initialization value called a *seed*. Different seed values result in different sequences of values, but for a given seed value the resulting sequence of values is always the same.

   Could a pseudo-random generator `rand()` could be used as part of a probing strategy to resolve collisions in a hash table? If no, explain why not? If so, describe an effective solution that would yield different probe sequences for different key values that collide in the same home slot.

3. [20 points] Most programming languages that have a `switch` statement limit the variable used to specify the switch target to be of some integer type. BTW, the point of a `switch` statement is that the cost of jumping to the correct case is Theta(1).

   However, Java 7 adds support for `switch` statements that use string variables to specify the switch target. Explain clearly, how the Java compiler can accomplish this.