My signature testifies that I have neither given nor received aid on this exam.

Name:_____ SUID:_____

CIS 351 Quiz 5

15 Nov 2017, Section M001

Question	Max	Score
1	8	
2	2	
3	5	
4	2	
5	8	
Total	25	

1. (8 pts) Consider the following hash function for strings; `ht` is the hash table.

```
int hash(String s) {
    int val = 0;

    for (int i = 0; i < 8 && i < s.length(); i += 2) {
        val += (int) s.charAt(i);
    }

    return (val % ht.size());
}
```

List four strings that would produce the same hash index for a table of size 32.

The hash is simply the sum of the even numbered characters in the string (0, 2, 4, and 6). Thus, any string either with these characters being the same, or even one that's shorter, would work. I was looking for at least 8 character-long strings, but didn't ask that, so some people gave "cheapie" answers.

Sample answers: "AaBaCaDa", "AbBbCbDb", "AcBcCcDc", "AdBdCdDd"
"A.B.C.D.", "B.C.D.A.", "C.D.B.A.", "D.A.B.C."

Cheap answer: "Aa", "Ab", "A" , "Ac"

2. (2 pts.) How often is the statement "For any hash function, it is possible to derive a set of many elements that all hash to the same value" true?

   (a) sometimes
   (b) always
   (c) never

   This is always true (answer (b)), assuming the universe of keys is $>>$ the number of slots in the table. As I didn't state that assumption, I accepted either (a) or (b).

3. (5 pts.) A university assigns each of its student a unique ID number the first time they register for a course. These are sequential integers that started many years ago and are now in the hundreds of thousands. Suppose we have a class of first year students and we want to assign them hash codes based on their student numbers. Does it make more sense to use a mod (%) or logical right shift ($>>>$) operator to map from the student number to the hash table index? Explain why.

   We want to use the mod operator. Because student ID numbers are allocated sequentially, almost all the students for a given year will have the same leading (leftmost) digts in the student ID numbers, but will differ in the trailing (rightmost) digits.

   Mod discards the leading digits; right shift discards the trailing digits. Therefore, mod is the better choice to avoid collisions.

4. (2 pts) In `ChainedHashTable.remove()`, the iterator is used to...

   (a) process each hash table slot in succession.
   (b) find the next slot with a `del` entry.
   (c) walk the list of elements with identical hash indices.
   (d) avoid collisions

   (c), walk the list of elements with identical hash indices.

5. (8 pts.) Consider the following hash table of size 10, with a hash function `hash(k) = k % 10`, and using linear probing with a stride of 1. After 6 insertions, the following table results:

slot	key
0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

Which of the following sequences of key value insertions could have resulted in the table shown above?

(a) 46, 42, 34, 52, 23, 33

(b) 34, 42, 23, 52, 33, 46

(c) 46, 34, 42, 23, 52, 33

(d) 42, 46, 33, 23, 34, 52

Things to note: 42, 23, 34, and 46 are in the correct slots; 52 and 33 are displaced. (A) Thus, 42 must come before 52, and 23 must come before 33. (B) Furthermore, 52 must come after 23 and 34, because otherwise it would have taken one of their slots. (C) Likewise, 33 must come last, because otherwise it would have landed in some earlier slot.

From (A) or (C) we can eliminate option (d). From (C) we can eliminate option (b). From (B) we can eliminate option (a), so option (c) is correct, and testing the insertion verifies it.