1. **Keys vs hash index: (10 points:)**

   Yes/No: In a hash table with chained elements for handling collisions, do elements in a chain have the same key?

   No

2. **Hashing: (40 points:)**

   Demonstrate what happens when we insert the keys 5, 28, 19, 15, 20, 33, 12, 17, 10, 11 into a hash table with collisions resolved by chaining. Let the table have 9 slots, and let the hash function be h(k) = k mod 9. Draw the contents of the hash table after all the keys have been inserted.

k	h(k)
5	5
28	1
19	1
15	6
20	2
33	6
12	3
17	8
10	1
11	2

Hash table	
key	value
0	
1	10, 19, 28
2	11, 20
3	12
4	
5	5
6	33, 15
7	
8	17

3. **Open Addressing: (40 points:)**

   Consider inserting the keys 10, 22, 31, 4, 15, 28, 17, 56, 87, 16 into a hash table of length m = 11 using open addressing with the auxiliary hash function h'(k) = k. Show the contents of the hash table using quadratic probing with c1 = 1 and c2 = 3. Recall that h(k,i) = (h'(k) + c1*i + c2*i^2) mod m.

h(k,i) = (k + i +3i^2) mod 11

	i=0	i=1	i=2	i=3
k	h(k,i)			
10	10			
22	0			
31	9			
4	4			
15	4	8		
28	6			
17	6	10	9	3
56	1			
87	10	3	2	
16	5			

Hash table	
key	value
0	22
1	56
2	87
3	17
4	4
5	16
6	28
7	
8	15
9	31
10	10

4. **Open Ended Question: (10 points:)**

   The social security number in the US consists of 9 digits and are unique keys assigned to individuals, and are not re-used even when the person holding a number dies. Of the billion minus 1 possible keys, around 450 million have already been assigned. Associated with each social security number is information about us e.g. name, birthday, address, etc.

   (a) (5 points:) Because social security numbers are (suppose to) uniquely identify an individual, they make excellent keys. If information about us are stored in a dictionary as key:value pairs in a hash table with chaining, **how many slots should the hash table have** if we are willing to search through the chain an average of 100 elements. That is a load factor of 100.

   m = 450M/100 = 4.5M

   (b) (5 points:) With recent data breaches and potential for identity thefts, many are requesting new social security numbers. That is, while a social security number uniquely identifies a person, a person can have more than one social security number. In a recent study conducted by ID Analytics, they found that around 20 million Americans have multiple social security numbers for various reasons – e.g. due to identity theft a person may request a new social security number. When issuing a new number to a person, the Social Security Administration will cross reference the two numbers to point to the same person.

   Explain how you would handle such cases in hash tables with chaining?

   add a link field to connect 2 nodes even if they're in different slots in hash table. downside is this will require space for another 450M pointers even if just a few are needed. do accept other solutions if they make sense even if not efficient.