Hash tables

A – Theory part

A.1. Consider the following hash table of size **M = 9** and a hash function **h(x) = x mod M**. Collisions are resolved using *linear probing*. The hash table has a fixed size and no elements have been deleted yet.

0	1	2	3	4	5	6	7	8
9	18		12	3	14	4	21	

In what order were the keys entered into the hash table? Please choose **all c**orrect answers.

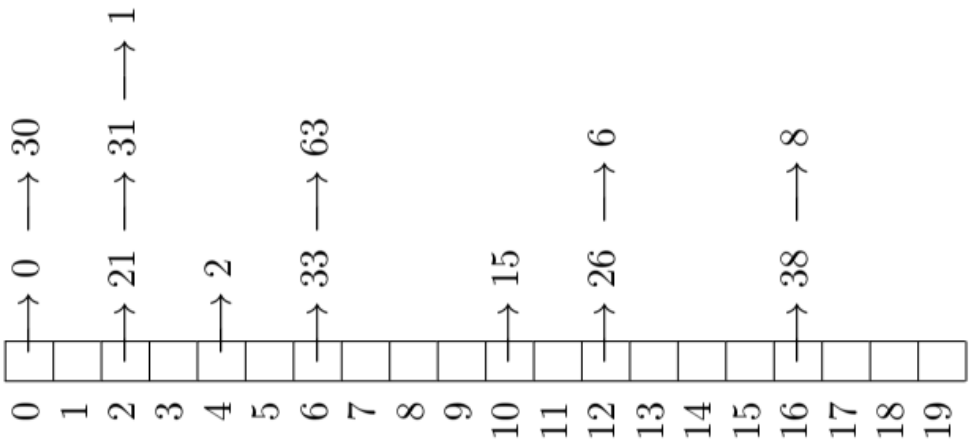
a. 9, 14, 4, 18, 12, 3, 21
b. 12, 3, 14, 18, 4, 9, 21
c. 12, 14, 3, 9, 4, 18, 21
d. 9, 12, 14, 3, 4, 21, 18
e. 12, 9, 18, 3, 14, 21, 4

A.2. Consider the following hash table of size **M = 11** and a hash function **h(x) = x mod M**. Keys 0, 1, 8, 9 are already in the hash table. Add keys 52, 44, 56, 53, 61, 64 to the hash table, respectively, using one of the following collision resolution strategies.

a. Linear probing
b. Quadratic probing
c. Double hashing with the secondary hash function **h'(x) = 7 – (x mod 7)**

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

A.3. Suppose there is a hash table containing several elements, as shown in the figure below



You observe the content of the hash table and discover a major problem with the hash table.

a. What is the problem with the hash table?

b. Describe a hash function that can lead to this problem.
c. Suggest another hash function to overcome the above issue.

A.4. Consider a hash table of size **M** and a hash function **h(x) = x mod M**. Collisions are resolved using *separate chaining*. For each of the following cases, identify the time complexity for adding **n** keys into the table.

a. Each element of the hash table is an *unordered linked list*, and a new key is added to the top of the list.
b. Each element of the hash table is an *ordered linked list*, and a new key is added in the appropriate position to keep the order.

A.5. Which of the following techniques is LEAST suitable for collision resolution?

a. Make the hash function output a random value
b. Design each element of the hash table as a one-way linked list
c. Detect empty positions in the hash table using various formulas (uniform hashing)
d. Increase the size of the hash table

A.6. You need to implement a database for a start-up telecommunications company. The database contains customer information, including names and phone numbers. With the following operations, the company wants the average execution time to be as small as possible: (1) add a new customer, (2) find a customer's phone number, and (3) delete a customer out of the database.

Choose one of the following data structures: (a) An extremely large static array, (b) A dynamically allocated array, (c) A linked list, and (d) A hash table.

List the issues to consider when designing a database with the chosen data structure.

A.7. Compare two collision resolution strategies, *separate chaining* and *open addressing*.

B – Coding part

Download the file containing English words from the below link

https://github.com/dwyl/english-words/blob/master/words_alpha.txt

Write a program that reads every word in the given file and stores them in a hash table. The program allows the user to choose one of the following collision resolution strategies.

1) Linking method, optionally using linked list type
2) Linear detection
3) Quadratic detection
4) Double hashing

The hash table size and the hash function (as well as the secondary hash function) are designed by your own.

The user enters a search term, called **S**. The program searches for the word **S** in the hash table with the previously selected collision resolution strategy. If found, answer "S is an English word." If not found, answer "S is not an English word." (Replace S with the corresponding word).

Measure the time taken to perform each search on the hash table. For each collision resolution strategy, perform at least 10 searches and calculate the average execution time.

Comment on the distribution of keys (words) and search time for each collision resolution strategy.