Question 1: Hash Tables .... [320 points]

(i) **[10 points] Complexity Analysis:**
Explain why constants are important in practice.

(ii) **[10 points] Naive Hash Table:**
List two limitations of a naive hash table.

(iii) **[10 points] Hash Function:**
Define a hash function. List two desirable properties of a hash function.

(iv) **[10 points] Hashing Scheme:**
Define a hashing scheme. List two desirable properties of a hashing scheme.

(v) **[10 points] Hash Function:**
List an assembly instruction that can be used to accelerate hashing.

(vi) **[10 points] Hashing Scheme:**
Distinguish between static and dynamic hashing schemes.

(vii) **[10 points] Hashing Scheme:**
Which one is faster in general: static or dynamic hashing schemes? Why?

(viii) **[10 points] Linear Probe Hashing:**
Explain how `INSERT` operation works in linear probe hashing.

(ix) **[10 points] Linear Probe Hashing:**
Explain how `DELETE` operation works in linear probe hashing.

(x) **[10 points] Linear Probe Hashing:**
Distinguish between these two solutions for handling `DELETE` operations: Tombstone and Movement of Keys.

(xi) **[10 points] Linear Probe Hashing:**
Explain how to support non-unique keys in a hash table.

(xii) **[10 points] Robin Hood Hashing:**
Explain how `INSERT` operation works in Robin Hood Hashing.

(xiii) **[10 points] Robin Hood Hashing:**
List a benefit and a limitation of Robin Hood Hashing in comparison to Linear Probe Hashing.

(xiv) **[10 points] Cuckoo Hashing:**
Explain how `INSERT` operation works in Cuckoo Hashing.

(xv) **[10 points] Cuckoo Hashing:**
Can a cycle arise in Cuckoo Hashing? Illustrate with an example.

(xvi) **[10 points] Cuckoo Hashing:**
How are cycles resolved in Cuckoo Hashing?

(xvii) **[10 points] Chained Hashing:**
Explain how `INSERT` operation works in chained hashing.

(xviii) **[10 points] Chained Hashing:**
How can we synchronize concurrent operations in a chained hash table?

(xix) **[10 points] Extendible Hashing:**
Explain how `INSERT` operation works in extendible hashing.

(xx) **[10 points] Extendible Hashing:**
List a benefit and a limitation of Extendible Hashing in comparison to Chained Hashing.

(xxi) **[10 points] Extendible Hashing:**
Explain why the number of bits that the hash table uses to map hashes to buckets changes over time.

(xxii) **[10 points] Extendible Hashing:**
Explain the purpose of Global and Local counters in extendible hashing. Can they be different from each other? Why?

(xxiii) **[10 points] Linear Hashing:**
Explain how `INSERT` operation works in linear hashing.

(xxiv) **[10 points] Linear Hashing:**
List a benefit and a limitation of Linear Hashing in comparison to Extendible Hashing.

(xxv) **[10 points] Linear Hashing:**
Explain the purpose of split pointer in linear hashing.

(xxvi) **[10 points] Linear Hashing:**
Why do we need multiple hashes to find the right bucket for a given key?

(xxvii) **[10 points] Linear Hashing:**
List two overflow criterion that may be used in linear hashing.

(xxviii) **[10 points] Linear Hashing:**
Explain how `DELETE` operation works in linear hashing.

(xxix) **[10 points] Linear vs Extendible Hashing:**
Distinguish between linear and extendible hashing.

(xxx) **[10 points] Linear vs Extendible Hashing:**
How are they related to each other: linear and extendible hashing? Which technique is more flexible? Why?

(xxxi) **[10 points] Hash Tables:**
Illustrate how a hash table is used within the DBMS.

(xxxii) **[10 points] Hash Tables:**
Why are hash tables typically only used for managing in-memory data and **not** on-disk data?