**🔹 Theory: Hashing with and without Replacement**

A **hash table** is a data structure used for fast data retrieval. It uses a **hash function** to compute an index into an array, where data can be stored or retrieved efficiently.

In a **telephone directory**, each entry consists of a telephone number (used as a key) and the client’s name (used as the value).

**Collision Handling Methods Used:**

1. **With Replacement (Chaining with Replacement):**
   * If the calculated hash index is occupied by an entry whose key does **not** belong to that index, the new entry **replaces** the existing one.
   * The displaced entry is then reinserted into the hash table using linear probing.
2. **Without Replacement:**
   * If a collision occurs, the new entry is inserted at the next available position using **linear probing**, **without replacing** the existing one.

**Advantages of With Replacement:**

* Ensures better clustering around actual home positions.
* Improves search efficiency for entries likely to be found at their original hash index.

**🔹 Algorithm**

**1. Insert With Replacement**

plaintext

CopyEdit

Input: key (telephone number), name

Output: Insert into hash table

1. Compute index = key % size

2. If table[index] is empty:

Insert at index

3. Else:

If table[index].key % size != index:

// Replace the record

Temp = table[index]

Insert new entry at index

Reinsert Temp using linear probing

Else:

Find next empty slot using linear probing and insert

**2. Insert Without Replacement**

plaintext

CopyEdit

Input: key, name

Output: Insert into hash table

1. Compute index = key % size

2. If table[index] is empty:

Insert at index

3. Else:

Find next empty slot using linear probing and insert

**3. Display Table**

plaintext

CopyEdit

For each slot in the table:

If slot is not empty:

Print index, key, name

Else:

Print "Empty"

**4. Find Record**

plaintext

CopyEdit

Input: key

Output: Display record if found

1. Compute index = key % size

2. While table[index] is not empty:

If table[index].key == key:

Print record and exit

index = (index + 1) % size

3. If loop ends without finding:

Print "Record not found"

**5. Delete Record**

plaintext

CopyEdit

Input: key

Output: Delete record if found

1. Use same steps as Find Record

2. If found, set slot to default empty values

