DATA STRUCTURES AND ALGORITHMS LABORATORY

Group A :- Assignment No. 1

Name :- Ojus Pravin Jaiswal

Roll No. :- SACO19108

Division :- A

DATA STRUCTURE LABORATORY

Course Objectives:

- To understand practical implementation and usage of non linear data structures for solving problems of different domain.
- To strengthen the ability to identify and apply the suitable data structure for the given real world problems.
- To analyze advanced data structures including hash table, dictionary, trees, graphs, sorting algorithms and file organization.

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Understand the ADT/libraries, hash tables and dictionary to design algorithms for a specific problem.

CO2: Choose most appropriate data structures and apply algorithms for graphical solutions of the problems.

CO3: Apply and analyze non linear data structures to solve real world complex problems.

CO4: Apply and analyze algorithm design techniques for indexing, sorting, multi-way searching, file organization and compression.

CO5: Analyze the efficiency of most appropriate data structure for creating efficient solutions for engineering design situations.

DATA STRUCTURE LABORATORY

Group A

Assignment 1

Title:

Consider telephone book database of N clients. Make use of a hash table implementation to quickly look up client‘s telephone number. Make use of two collision handling techniques and compare them using number of comparisons required to find a set of telephone numbers

Objectives:

1. To understand concept of Hashing
2. To understand to find record quickly using hash function.
3. To understand concept & features of object oriented programming.

Outcome:

1. Identify object oriented Programming features
2. Implement the concept of hash table.

Theory:

Hashing is a technique that uses fewer key comparisons and searches the element in O(n) time in the worst case and in O(1) time in the average case.

The task is to implement functions of phone directory:

- create_record
- display_record
- search_record
- delete_record

Following data will be taken from the client:

Name, Telephone number

Approach:

We are creating a hash table, and inserting records. For deleting, searching, or updating an entity. If the record is not found, then an appropriate message is displayed.

Collision is the major problem in the hashing technique. In open addressing (closed hashing), all collisions are resolved in the prime area i.e., the area that contains all of the home addresses.

When a collision occurs, the prime area addresses are searched for an open or unoccupied element using linear probing.

Steps for inserting entities in a hash table:
1. If the location is empty, directly insert the entity.
2. If mapped location is occupied then keep probing until an empty slot is found. Once an empty slot is found, insert the entity.

- Create Record: This method takes details from the user like, Name and Telephone number and create new record in the hashtable.
- Display Record: This function is created to display all the record of the diary.
- Search Record: This method takes the key of the record to be searched. Then, it traverses the hash table, if record id matches with the key it displays the record detail.
- Delete Record: This method takes the key of the record to be deleted. Then, it traverses the hash table, if record id matches then the record is deleted

Software Required: Python

Input: *Name, Telephone number*.

Program:

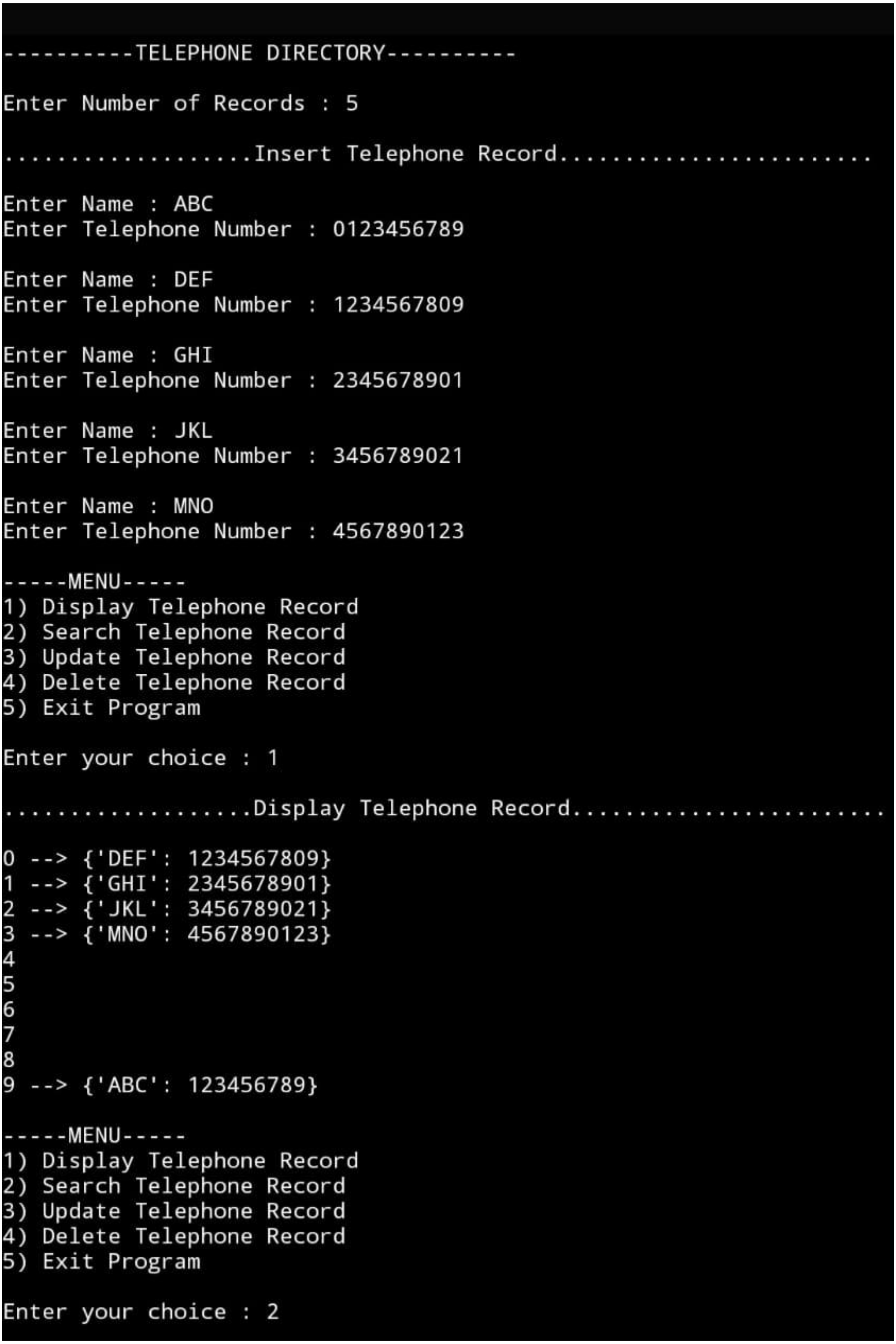
```
# Create HashTable
HashTable = [[] for _ in range(10)]

# Display records
def display_hash(hashTable):

    for i in range(len(hashTable)):
        print(i, end = " ")

        for j in hashTable[i]:
            if j !={}:
              print("-->", end = " ")
              print(j, end = " ")

        print()
```

```
# Hash function
def Hashing(keyvalue):
        return keyvalue % len(HashTable)



# Insert Function to add
# values to the hash table
#Chaining
def insert(Hashtable, keyvalue, value):

        hash_key = Hashing(keyvalue)
        Hashtable[hash_key].append({value:keyvalue})

# Search record
def find(Hashtable, value):
    f=0
    for i in range( 1, 10):
        for j in Hashtable[i]:
            if value in j:
                print (j)
                f=1
                return j
                break
    if f==0:
        print("Not Found")
        return 0

# update record
def update(Hashtable, keyvalue, value):
    pos={}
    hash_key = Hashing(keyvalue)
    pos=find(Hashtable,value)
    if pos !={}:
        del(pos[value])
    Hashtable[hash_key].append({value:keyvalue})
```

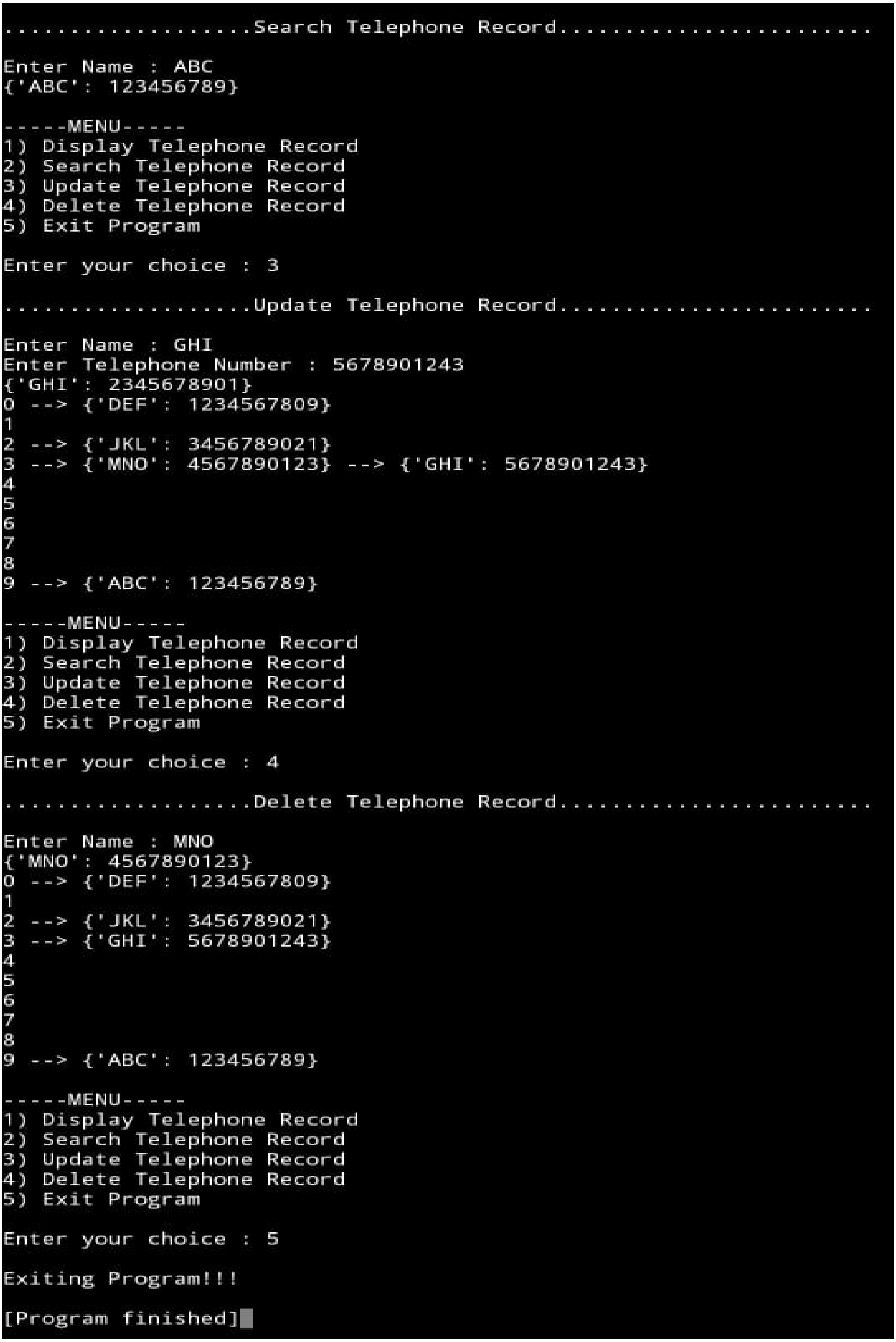
```
# delete record
def delete(Hashtable, value):
    pos={}
    pos=find(Hashtable,value)
    if pos !={}:
        del(pos[value])

# Insert Function to add
# values to the hash table
#linear probing
def insert_l(Hashtable, keyvalue, value):

        hash_key = Hashing(keyvalue)
        key=hash_key
        while Hashtable[hash_key] !=[]:
            hash_key = Hashing(hash_key+1)
        if Hashtable[hash_key] ==[]:
            Hashtable[hash_key].append({value:keyvalue})

# Driver Code
print("----------TELEPHONE DIRECTORY----------\n")
n = int(input("Enter Number of Records : "))
print ("\n..................Insert Telephone Record.......................")
for i in range(0,n) :
        name = input("\nEnter Name : ")
        phone_no = int(input("Enter Telephone Number : "))
        insert_l(HashTable, phone_no, name)

while True :
        print("\n-----MENU-----\n1) Display Telephone Record\n2) Search Telephone Record\n3) Update Telephone Record\n4) Delete Telephone Record\n5) Exit Program\n")
        c=int(input("Enter your choice : "))
```

```
if c== 1 :
        print ("\n..................Display Telephone Record.......................\n")
        display_hash (HashTable)

elif c==2 :
        print ("\n..................Search Telephone Record.......................\n")
        name = input("Enter Name : ")
        find(HashTable,name)

elif c==3 :
        print ("\n..................Update Telephone Record.......................\n")
        name = input("Enter Name : ")
        phone_no = int(input("Enter Telephone Number : "))
        update(HashTable, phone_no, name)
        display_hash (HashTable)

elif c==4 :
        print ("\n..................Delete Telephone Record.......................\n")
        name = input("Enter Name : ")
        delete(HashTable,name)
        display_hash (HashTable)

elif c==5 :
        print("\nExiting Program!!!")
        exit(0)

else :
        print("\nWrong choice entered!!!")
```

Output:



```
----------TELEPHONE DIRECTORY----------

Enter Number of Records : 5

..................Insert Telephone Record.......................

Enter Name : ABC
Enter Telephone Number : 0123456789

Enter Name : DEF
Enter Telephone Number : 1234567809

Enter Name : GHI
Enter Telephone Number : 2345678901

Enter Name : JKL
Enter Telephone Number : 3456789021

Enter Name : MNO
Enter Telephone Number : 4567890123

-----MENU-----
1) Display Telephone Record
2) Search Telephone Record
3) Update Telephone Record
4) Delete Telephone Record
5) Exit Program

Enter your choice : 1

..................Display Telephone Record.......................

0 --> {'DEF': 1234567809}
1 --> {'GHI': 2345678901}
2 --> {'JKL': 3456789021}
3 --> {'MNO': 4567890123}
4
5
6
7
8
9 --> {'ABC': 123456789}

-----MENU-----
1) Display Telephone Record
2) Search Telephone Record
3) Update Telephone Record
4) Delete Telephone Record
5) Exit Program

Enter your choice : 2
```



```
..................Search Telephone Record........................

Enter Name : ABC
{'ABC': 123456789}

-----MENU-----
1) Display Telephone Record
2) Search Telephone Record
3) Update Telephone Record
4) Delete Telephone Record
5) Exit Program

Enter your choice : 3

..................Update Telephone Record........................

Enter Name : GHI
Enter Telephone Number : 5678901243
{'GHI': 2345678901}
0 --> {'DEF': 1234567809}
1
2 --> {'JKL': 3456789021}
3 --> {'MNO': 4567890123} --> {'GHI': 5678901243}
4
5
6
7
8
9 --> {'ABC': 123456789}

-----MENU-----
1) Display Telephone Record
2) Search Telephone Record
3) Update Telephone Record
4) Delete Telephone Record
5) Exit Program

Enter your choice : 4

..................Delete Telephone Record........................

Enter Name : MNO
{'MNO': 4567890123}
0 --> {'DEF': 1234567809}
1
2 --> {'JKL': 3456789021}
3 --> {'GHI': 5678901243}
4
5
6
7
8
9 --> {'ABC': 123456789}

-----MENU-----
1) Display Telephone Record
2) Search Telephone Record
3) Update Telephone Record
4) Delete Telephone Record
5) Exit Program

Enter your choice : 5

Exiting Program!!!

[Program finished]
```

Conclusion: This program statement implements Hash Table Concept.