

## Data Structures and Algorithms

### Lab Journal - Lab 9

Name: \_\_\_\_\_

Enrollment #: \_\_\_\_\_

Class/Section: \_\_\_\_\_

#### Objective

This lab session is intended to introduce students to the Hashing and concepts related to Hash Table.

#### Task 1 :

Give answers to the following.

1.	<p>Consider the following hash function</p> <p><math>\text{Hash}(i) = i \% 10</math></p> <p>Map the values into hash Table</p> <p>12, 22, 24, 14, 48, 58</p>	<table style="margin: auto;"> <tr><td>0</td><td style="border: 1px solid black; width: 50px; height: 20px;"></td></tr> <tr><td>1</td><td style="border: 1px solid black; width: 50px; height: 20px;"></td></tr> <tr><td>2</td><td style="border: 1px solid black; width: 50px; height: 20px;"></td></tr> <tr><td>3</td><td style="border: 1px solid black; width: 50px; height: 20px;"></td></tr> <tr><td>4</td><td style="border: 1px solid black; width: 50px; height: 20px;"></td></tr> <tr><td>5</td><td style="border: 1px solid black; width: 50px; height: 20px;"></td></tr> <tr><td>6</td><td style="border: 1px solid black; width: 50px; height: 20px;"></td></tr> <tr><td>7</td><td style="border: 1px solid black; width: 50px; height: 20px;"></td></tr> <tr><td>8</td><td style="border: 1px solid black; width: 50px; height: 20px;"></td></tr> <tr><td>9</td><td style="border: 1px solid black; width: 50px; height: 20px;"></td></tr> </table> <p style="text-align: center;">Hash Table</p>	0		1		2		3		4		5		6		7		8		9	
0																						
1																						
2																						
3																						
4																						
5																						
6																						
7																						
8																						
9																						
2.	<p>From part 1, it is shown that for some cases i.e. remainder of 48 and 58 would be 8 so in that case collusion occurs. What are three techniques to resolve the collision problem?</p> <p>a)</p> <p>b)</p> <p>c)</p>																					

3.	What is the advantage of hashing over linear search and binary search ?
4.	<p>From part 1, map the values in to hash table by considering seperate chaining method.</p> <p>12, 22 ,24,14,48,58</p>

**Task 2 :**

Implement the following exercises.

**Exercise 1**

Implement the following functions and insert a given set of integers into a hash table.

```
//Hash Function to generate the index
```

```
int hash(int key) ;
```

```
//A function that accepts the hash table and key to be inserted
and inserts the "key" at appropriate location in the table. Use
linear probing to resolve collisions. The returned values is the
index at which the key is inserted.
```

```
int linear_probing(int HashTable[], int key) ;
```

```
//A function that inserts values in the table and resolves
collisions using quardatice probing.
```

```
int quardatic_probing(int HashTable[], int key) ;
```

**HINTS :**

Quardatic probing can be implemented like :

```
for (i = 0 ; i% MAX != pos ; i++)
    pos = (pos + i * i) % MAX ;
```

In the main program, take a number of values from the user and insert them in the hash table using linear probing. Insert the same values in another hash table using quardatic probing.

**Exercise 2**

Consider the following class to store integers in nodes of a list.

```
class Node{
    public :
        int key ;
        Node * next ;
} ;
```

And the following hash table :

```
Node * hashtable[MAX] ;
```

Implement the following functions.

```
//Hash function that generate hash index
int hashfunction(int key);
```

```
//Intialize the array of pointers to NULL
void initialize();
```

```
// Insert a value in the hash table; You need to create a node,
insert the value in the node and place the node at appropriate
location.
```

```
void insert(int k);
```

```
// Display the complete data in the hash table
void display();
```

```
// Search a given value in the list
void search(int v);
```

Implement the given exercises and get them checked by your instructor. If you are unable to complete the tasks in the lab session, deposit this journal alongwith your programs (printed or handwritten) before the start of the next lab session.

S No.	Exercise	Checked By:
1.	Exercise 1	
2.	Exercise 2	

+++++