Rajalakshmi Engineering College

*241901104*

*241901104*

*241901104*

Name: Shafraz.R

*241901104*

Email: 241901104@rajalakshmi.edu.in

Roll no: 241901104

Phone: 6380360049

Branch: REC

Department: l CSE (CS) FB

Batch: 2028

Degree: B.E - CSE (CS)

NeoColab\_REC\_CS23231\_DATA STRUCTURES

*241901104*

*241901104*

REC\_DS using C\_Week 7\_COD\_Question 2

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Priya is developing a simple student management system. She wants to store roll numbers in a hash table using Linear Probing, and later search for specific roll numbers to check if they exist.

*241901104*

*241901104*

*241901104*

*241901104*

Implement a hash table using linear probing with the following operations:

Insert all roll numbers into the hash table.For a list of query roll numbers, print "Value x: Found" or "Value x: Not Found" depending on whether it exists in the table.

The first line contains two integers, n and table\_size — the number of roll numbers to insert and the size of the hash table.

*Input Format*

*241901104*

*241901104*

*241901104*

*241901104*

The second line contains n space-separated integers — the roll numbers to

The third line contains an integer q — the number of queries.

The fourth line contains q space-separated integers — the roll numbers to search for.

insert.

*241901104*

*241901104*

*241901104*

*241901104*

*Output Format*

The output print q lines — for each query value x, print: "Value x: Found" or "Value x: Not Found"

Refer to the sample output for formatting specifications.

*241901104*

*241901104*

*241901104*

*241901104*

*Sample Test Case*

Input: 5 10

21 31 41 51 61

3

31 60 51

Output: Value 31: Found

Value 60: Not Found

*241901104*

*241901104*

*241901104*

Value 51: Found

*241901104*

*Answer*

#include <stdio.h>

#define MAX 100

// You are using GCC

void initializeTable(int table[], int size) {

//Type your code here

for (int i= 0; i < size; i++) table[i] =-1;

}

int linearProbe(int table[], int size, int num) {

*241901104*

*241901104*

int index= num% size; int start =index; while (table[index] != -1) { index =(index + 1) % size; if (index==start) break;

*241901104*

*241901104*

*241901104*

*241901104*

*241901104*

*241901104*

}

return index;

}

void insertIntoHashTable(int table[], int size, int arr[], int n) {

//Type your code here for (int i= 0;i<n;i++) {

int index=linearProbe(table,size,arr[i]); table[index] =arr[i];

}

}

int searchInHashTable(int table[], int size, int num) { int index = num %size; int start = index; while(table[index]!= -1) {

*241901104*

*241901104*

*241901104*

*241901104*

if(table[index] ==num) return 1;

index =(index +1) %size; if(index ==start) break;

}

*241901104*

*241901104*

return 0; } int main() { int n, table\_size;

*241901104*

scanf("%d %d", &n, &table\_size);

*241901104*

int arr[MAX], table[MAX]; for (int i = 0; i < n; i++)

scanf("%d", &arr[i]);

initializeTable(table, table\_size); insertIntoHashTable(table, table\_size, arr, n);

*241901104*

*241901104*

int q, x;

scanf("%d", &q); for (int i = 0; i < q; i++) { scanf("%d", &x);

*241901104*

*241901104*

if (searchInHashTable(table, table\_size, x)) printf("Value %d: Found\n", x); else

*241901104*

*241901104*

*241901104*

*241901104*

printf("Value %d: Not Found\n", x);

}

return 0; }

*Status :* Correct *Marks : 10/10*

*241901104*

*241901104*

*241901104*

*241901104*

*241901104*

*241901104*

*241901104*

*241901104*

*241901104*

*241901104*

*241901104*

*241901104*