Rajalakshmi Engineering College

Name: B. Sunetra
Email: Sunetra.b2024.aids@rajalakshmi.edu.in
Roll No: 2116241801282
Phone: 9150958405
Branch: REC
Department: AIDS FD
Batch: 2028
Degree: BE - AI&DS

Week - 4
Exp: 1
Date: 29/5/25

REC. DS using C Week 7 C.OD Question 1

Section - 1 Coding

1. Problem Statement

Ravi is building a basic hash table to manage student roll no.'s for quick lookup. He decides to use linear probing.

The hash function is: index = roll_number % table_size. On collision, check subsequent indexes (i+1, i+2, ...) until an empty slot.

You need to insert a list of n students into hash table. Print the final state of hash table.

Input:
The first line contains 2 integers n, table_size.
The second line contains n space seperated integers.

Output:
The output should print a single line with table size space seperated integers representing the final state.

Sample Test Case.

4 7

50 700 76 85

700 50 85 -1 -1 -1 76.

Answer

```
# include <stdio.h>
# define MAX 100
# include <stdio.h>
# define MAX 100.
void initialize Table (int table [ ], int size ) {
    for (int i = 0, i < size ; i++ ) {
        table [i] = -1
    }
}
int linear Probe (int table [ ], int size, int num ) {
    int index = num % size
    while (table [index ] ! = -1 ) {
        index = (index + 1) % size ,
    }
    return index .
}
Void insertIntoHashTable (int table [ ], int size, int arr [ ], int n ) {
    for (int i = 0 ; i < n ; i++ ) {
        int index = linearProbe (table , size, arr [i ])
        table [index ] = arr [i ] ;
    }
}
```

```
void printTable ( int table[], int size ){
    for(int i=0; i<size, i++){
        printf("%d ", table[i]);
    }
    printf("\n");
}

int main(){
    int n, table_size;
    scanf("%d %d", &n, &table_size);
    int arr[MAX]
    int table[MAX]
    for(int i=0; i<n; i++)
        scanf("%d", &arr[i])

    initializeTable(Table, Table_size);
    insertIntoHashTable(table, Table_size, arr, n);
    printTable(Table, Table_size);
    return 0;

}
```

RESULT: Thus, a program to display final state representing integers is implemented and verified successfully using C program

Rajalakshmi Engineering College.

Name: B. Sunelia
email: Sunelia.b.2024.aids@rajalakshmi.edu.in
RollNo: 2116 241801282
Phone: 9150958405
Branch: REC
Department: AIDS FD
Batch: 2028
Degree: Bt AI&DS

Week-7
Exp-2
Date 29/5/25

REC_DS_using C\Week-7_COD. Question 2

1. Problem Statement:

Priya is developing a simple student management system. She wants to store roll numbers in hashtable using linear probing.

Implement a hash table using linear probing with:
insert a roll no. into hash table. For a list of query for roll numbers, print ("Value x: Found" or "Value x: Not Found" depending on whether it exists in the table.

Input:
The first line contains 2 integers, n, table_size.
The second line - n space-separated integers.
The third line - integer q
The fourth line - q space-separated integers.

Sample Test Case

5 10
21 31 41 51 61
3
31 60 51
Output: Value 31: Found

Value 60 : Not found.
Value 51 : Found

Answer.

```
# include <stdio.h>
# define MAX 100.
Void initialize Table (int lable[], int size ){
      for (int i=0; i< size ; ++i ){
            table [i] = = -1;
      }
}
int linear Probe (int lable [], int size, int num) {
    int index = num % size;
      while (table [index] ! = -1) {
       index = (index +1) % size;
}
  retun index;
}
void insert into Hash Table ( int fable [], int size, int arr [], int n){
   for (int i=0 ; i <n ; i++) {
     int index = linear Probe (table, size, arr [i];
     table [index] = arr [i];
  }
}
int search in Hash Table (int table [], int size, int num) {
     int index = num % size;
     int start index = index;
  while (table [index] ! =-1) {
     If (table [index] == num) {
} retun 1;
```

```
        index = (index + 1) % size;
        if (index == start index) {
            break;
        }
    }
    return 0;
}

int main () {
    int n, table_size;
    scanf ("%d %d", &n, &table_size);

    int arr [MAX], table [MAX];
    for(int i=0; i<n; i++)
        scanf ("%d", &arr[i]);

    initialize Table (table, table_size);
    insert into Hash Table (table, table_size, arr, n);

    int q, x;
    scanf ("%d", &q);
    for (int i = 0; i<q; i++) {
        scanf ("%d", &x);
        if (search In Hash Table (table, table_size, x))
            Printf ("Value %d : found\n", x);
        else
            Printf ("Value %d : Not found \n", x);
    }
    return 0;
}
```

RESULT: ~~Status~~ Thus a program to check if a value is found or not is implemented and verified successfully using C program.

Rajalakshmi Engineering College.

Name : B. Suneetha
Email : suneetha.b2024.aids@rajalakshmi.edu.in
Roll No : 2116 241801282
Phone : 9150958405
Branch : REC
Department : AIDS FD
Batch : 2028
Degree : BE AI & DS

Week-7
EXP1-3
Date : 30/5/25

REC_DS using C_Week 7_COD_ Question 3

1. Problem statement :

In a messaging application, users maintain a contact list with names and corresponding phone numbers. Develop a program to manage this contact list using a dictionary implemented with hashing.

The program allows users to add contacts, delete and check if a specific contact exists.

Input :
The first line contains an integer n.
each of the next n lines consists of 2 strings seperated by a space.
The last line contains a string K, representing the contact.

OUTPUT :
The first line prints, " The given Key is removed ".
The next n-1 lines print the updated contact
Key : X ; Value : Y , X- contact name, Y- phone number

Sample Test Case.

Input : 3
Alice 1234567890
Bob 9876543210
Charlie 4567890123
Bob

Output : The given Key is removed!
Key : Alice ; Value : 1234567890
Key : Charlie , Value : 4567890123.

Answer.

```
#include <stdio.h>
#include <string.h>
#define MAX 50

typedef struct {
    char name[11];
    char phone[20];
} contact;

void insertContacts(Contact contacts[], int *count, int n) {
    for(int i=0; i<n; i++) {
        scanf("%s %s", contacts[*count].name,
              contacts[*count].phone);
        (*count)++;
    }
}
```

```
void manageContact (Contact contacts [], int * count, char Key[])
{
    int found :0
    for (int i : 0 ; i < * count ; i ++ ){
        if ( strcmp ( contacts [i].name, key ) == 0) {
            found = 1 ;
            Printf ("The given key is removed! \n") ;

        for(int j = i ; j < * count - 1 ; j++) {
            contacts [j] = contacts [j+1] ;

        }
        ( * count )-- ;
        break ;
        }
    }
    if (! found) {
        Printf ("The given key is not found ! \n");
        }
    }
    void printContacts (Contact contacts [], int count) {
        for(int i=0; i< count; i++) {
        printf (" key : %s ; value : %s \n", contacts [i].name .
                                            contacts [i]. phone ;
        }
    }
    int main () {
        int n ;
    Scanf ("%d", &n);
```

```
    Contact contacts[MAX];
    int count = 0;

    insertContacts(contacts, &count, n);

    char key[11];
    scanf("%s", key);

    manageContact(contacts, &count, key);
    PrintContacts(contacts, count);
    return 0;
}
```

RESULT:

Thus a program to remove a key is implement and successfully verified using C program.

Rajalakshmi Engineering College

Name: B. Suneela
email: suneela.b.2024.aids@rajalakshmi.edu.in
RollNo: 2116 241801282
Phone: 91509 58405
Branch: REC
Department: AIDS FD
Batch: 2028
Degree: BE AIDS

Week-7
EXP-4
Date 30/5/25

REC_DS using C-Week 7_COD-Question 4.

1. Problem Statement:

Develop a program using hashtag to manage a fruit contest where each fruit is assigned a unique name and corresponding score. The program allows the organizer to input the number of fruits.

It should enable them to check if a specific fruit, identified by its name. If the fruit is registered, the program should display its score.

Input:
The first line consists of integer N.
The following N lines contain a string K and integer V.

Output:
If T exists in the dictionary, print "Key "T" exists in the dictionary".
print "Key "T" does not exist in the dictionary.

Sample Test Case

Input : 2
Banana . 2
apple 1
Banana

Output : Key "Banana" does not exist in the dictionary.

Answer.

```
# include <stdio.h>
# include <string.h>.
  # define MAX 15

    typedef struct {
        char name [20];
        int score;
    } fruit;

void insert fruits ( fruits fruits [ ]; int * count, int n) {
        for int i =0; i<n ; i++) {
            scanf ("%s %d", fruits [*count].name, & fruits [*count]. score];
            (* count) ++;
    }
  }
    void search fruit (fruit fruits [ ], int count, char Key [ ]) {
        for (int i=0; i< count ; i++) {
          if (strcmp (fruits [i].name, key) ==0) {
          Printf (" Key \"%s\" exists in the dictionary. \n", Key);
          return;
    }
  }
```

RESULT: Thus a program to check if a Key exists in a dictionary and implemented and successfully verified using C Program

Rajalakshmi Engineering College

Name : B. Sunekaa
email : Sunekaa.b.2024.aids@rajalakshmi.edu.in
Roll No : 2116 241801 282
Phone : 9150955405
Branch : REC
Department : AIDS FD
Batch : 2028
Degree : BE AI&DS

Week-7
Experiment : 45
Date : 30/5/25

REC_DS using C. Week 7 - COD. Question 45

1. Problem Statement :

You are provided with a collection of numbers, each represented by an array of integers. However, there's a unique scenario, within this array, one element occurs an odd No. of times while all other elements occur an even number of times. Your objective is to identify and return the element that occurs an odd No. of times.

Utilize mid-square hashing by squaring elements & extracting middle digits for hash codes.

Hash function squared: Key * Key.

Input
7
22 33 44 5
Output:
5

Explanation :

The hash function and the calculated hash indices for each element

2 → hash(2*2) = 4

3 → hash(3*3)%100 = 9

The hash table records the occurrence of each element's index.

index 4 : 2 occurrences

index 9 : 2 occurrences

index 16 : 2 occurrences

index 25 : 1 occurrence

Among the elements, the integer 5 occurs once and odd No. of times.

Input :

First line of input consists of N space seperated integers.

Second line consists of N space seperated integers.

Output

The output prints a single integer representing the element.

Sample Test case :

Input 7

2 2 3 3 4 4 5

Output : 5

Answer.

```
# include <stdio.h>
# include <string.h>
# include <stdlib.h>
# include <stdbool.h>
# define MAX_SIZE 100.

unsigned int hash (int key, int tablesize) {
        return (key * key) % tablesize;
    }

int getOddOccurence (int arr[], int size) {
        int hashTable [MAX_SIZE] = {0};

for (int i=0; i<size; i++) {
        int index = hash (arr[i], MAX_SIZE);
        hashTable [index]++;
    }
        for (int i=0; i<size; i++) {
            int index = hash (arr[i]), MAX_SIZE);
            if (hashTable [index] % 2 != 0) {
                return arr[i];
            }
        }
        return -1;
}
```

```
int main() {
    int n;
    scanf("%d", &n);
    int arr[MAX_SIZE];
    for(int i=0; i<n; i++){
        scanf("%d", &arr[i]);
    }
    printf("%d\n", getOddOccurrence(arr, n));
    return 0;
}
```

RESULT : Thus a program displaying a single integer to represent odd No of times is implemented and verified successfully using C program.