Assignment No-A02

Name-Kavita Balivada

Roll No. – SECOMPA05

Sub- DSAL

Program:

//Assignment No 2

/\*Problem Statement-

Implement all the functions of a dictionary (ADT) using hashing and handle collisions using

chaining with/without replacement. Data: Set of (key, value) pairs, Keys are mapped to values,

Keys must be comparable, Keys must be unique. Standard Operations: Insert (key, value), Find

(key), Delete(key).\*/

#include <iostream>

#include <string.h>

using namespace std;

class HashFunction

{

    typedef struct hash

    {

        long key;

        char name[10];

    } hash;

    hash h[10];

public:

    HashFunction();

    void insert();

    void display();

    int find(long);

    void Delete(long);

};

HashFunction::HashFunction()

{

    int i;

    for (i = 0; i < 10; i++)

    {

        h[i].key = -1;

        strcpy(h[i].name, "NULL");

    }

}

void HashFunction::Delete(long k)

{

    int index = find(k);

    if (index == -1)

    {

        cout << "\n\tKey Not Found";

    }

    else

    {

        h[index].key = -1;

        strcpy(h[index].name, "NULL");

        cout << "\n\tKey is Deleted";

    }

}

int HashFunction::find(long k)

{

    int i;

    for (i = 0; i < 10; i++)

    {

        if (h[i].key == k)

        {

            cout << "\n\t" << h[i].key << " is Found at " << i << " Location With Name " << h[i].name;

            return i;

        }

    }

    if (i == 10)

    {

        return -1;

    }

    return 0;

}

void HashFunction::display()

{

    int i;

    cout << "\n\t\tKey\t\tName";

    for (i = 0; i < 10; i++)

    {

        cout << "\n\th[" << i << "]\t" << h[i].key << "\t\t" << h[i].name;

    }

}

void HashFunction::insert()

{

    char ans, n[10], ntemp[10];

    long k, temp;

    int v, hi, cnt = 0, flag = 0, i;

    do

    {

        if (cnt >= 10)

        {

            cout << "\n\tHash Table is FULL";

            break;

        }

        cout << "\n\tEnter a Telephone No: ";

        cin >> k;

        cout << "\n\tEnter a Client Name: ";

        cin >> n;

        hi = k % 10; // hash function

        if (h[hi].key == -1)

        {

            h[hi].key = k;

            strcpy(h[hi].name, n);

        }

        else

        {

            if (h[hi].key % 10 != hi)

            {

                temp = h[hi].key;

                strcpy(ntemp, h[hi].name);

                h[hi].key = k;

                strcpy(h[hi].name, n);

                for (i = hi + 1; i < 10; i++)

                {

                    if (h[i].key == -1)

                    {

                        h[i].key = temp;

                        strcpy(h[i].name, ntemp);

                        flag = 1;

                        break;

                    }

                }

                for (i = 0; i < hi && flag == 0; i++)

                {

                    if (h[i].key == -1)

                    {

                        h[i].key = temp;

                        strcpy(h[i].name, ntemp);

                        break;

                    }

                }

            }

            else

            {

                for (i = hi + 1; i < 10; i++)

                {

                    if (h[i].key == -1)

                    {

                        h[i].key = k;

                        strcpy(h[i].name, n);

                        flag = 1;

                        break;

                    }

                }

                for (i = 0; i < hi && flag == 0; i++)

                {

                    if (h[i].key == -1)

                    {

                        h[i].key = k;

                        strcpy(h[i].name, n);

                        break;

                    }

                }

            }

        }

        flag = 0;

        cnt++;

        cout << "\n\t..... Do You Want to Insert More Key: y/n";

        cin >> ans;

    } while (ans == 'y' || ans == 'Y');

}

int main()

{

    long k;

    int ch, index;

    char ans;

    HashFunction obj;

    do

    {

        cout << "\n\t\*\* Telephone (ADT) \*\*";

        cout << "\n\t1. Insert\n\t2. Display\n\t3. Find\n\t4. Delete\n\t5. Exit\n\t";

        cout << "\n\t..... Enter Your Choice: ";

        cin >> ch;

        switch (ch)

        {

        case 1:

            obj.insert();

            break;

        case 2:

            obj.display();

            break;

        case 3:

            cout << "\n\tEnter a Key Which You Want to Search: ";

            cin >> k;

            index = obj.find(k);

            if (index == -1)

            {

                cout << "\n\tKey Not Found";

            }

            break;

        case 4:

            cout << "\n\tEnter a Key Which You Want to Delete: ";

            cin >> k;

            obj.Delete(k);

            break;

        case 5:

            break;

        }

        cout << "\n\t..... Do You Want to Continue in Main Menu:y/n ";

        cin >> ans;

    } while (ans == 'y' || ans == 'Y');

}

Output:

