**PROGRAM 8)** Write a program to implement functions of Dictionary using Hashing.

CODE:

#include<bits/stdc++.h>

using namespace std;

const int Table\_size = 200;

class HashTableEntry {

public:

int k;

int v;

HashTableEntry(int k, int v) {

this->k= k;

this->v = v;

}

};

class HashMapTable {

private:

HashTableEntry \*\*t;

public:

HashMapTable() {

t = new HashTableEntry \* [Table\_size];

for (int i = 0; i< Table\_size; i++) {

t[i] = NULL;

}

}

int hashFunc(int k) {

return k % Table\_size;

}

void insert(int k, int v) {

int h = hashFunc(k);

while (t[h] != NULL && t[h]->k != k) {

h = hashFunc(h + 1);

}

if (t[h] != NULL)

delete t[h];

t[h] = new HashTableEntry(k, v);

}

int search(int k) {

int h = hashFunc(k);

while (t[h] != NULL && t[h]->k != k) {

h = hashFunc(h + 1);

}

if (t[h] == NULL)

return -1;

else

return t[h]->v;

}

void deleteEle(int k) {

int h = hashFunc(k);

while (t[h] != NULL) {

if (t[h]->k == k)

break;

h = hashFunc(h + 1);

}

if (t[h] == NULL) {

cout<<"No Element found at key "<<k<<endl;

return;

} else {

delete t[h];

}

cout<<"Element Deleted"<<endl;

}

~HashMapTable() {

for (int i = 0; i < Table\_size; i++) {

if (t[i] != NULL)

delete t[i];

delete[] t;

}

}

};

int main() {

HashMapTable hash;

int k, v;

int c;

while (1) {

cout<<"1.Insert"<<endl;

cout<<"2.Search"<<endl;

cout<<"3.Delete"<<endl;

cout<<"4.Exit"<<endl;

cout<<"Enter your choice: ";

cin>>c;

switch(c) {

case 1:

cout<<"Enter element to be inserted: ";

cin>>v;

cout<<"Enter key at which element to be inserted: ";

cin>>k;

hash.insert(k, v);

break;

case 2:

cout<<"Enter key of the element to be searched: ";

cin>>k;

if (hash.search(k) == -1) {

cout<<"No element found at key "<<k<<endl;

continue;

} else {

cout<<"Element at key "<<k<<" : ";

cout<<hash.search(k)<<endl;

}

break;

case 3:

cout<<"Enter key of the element to be deleted: ";

cin>>k;

hash.deleteEle(k);

break;

case 4:

exit(1);

default:

cout<<"\nEnter correct option\n";

}

}

return 0;

}

OUTPUT:



