**ASSIGNMENT NO.7.**

**Aim :-** Insert the keys into a hash table of length m using open addressing using double hashing with h(k)=1+(k mod(m-1)).

**Program Code:-**

#include <bits/stdc++.h>

using namespace std;

#define TABLE\_SIZE 13

#define PRIME 7

class DoubleHash

{

int \*hashTable;

int curr\_size;

public:

bool isFull()

{

return (curr\_size == TABLE\_SIZE);

}

int hash1(int key)

{

return (key % TABLE\_SIZE);

}

int hash2(int key)

{

return (PRIME - (key % PRIME));

}

DoubleHash()

{

hashTable = new int[TABLE\_SIZE];

curr\_size = 0;

for (int i=0; i<TABLE\_SIZE; i++)

hashTable[i] = -1;

}

void insertHash(int key)

{

if (isFull())

return;

int index = hash1(key);

if (hashTable[index] != -1)

{

int index2 = hash2(key);

int i = 1;

while (1)

{

int newIndex = (index + i \* index2) %

TABLE\_SIZE;

if (hashTable[newIndex] == -1)

{

hashTable[newIndex] = key;

break;

}

i++;

}

}

else

hashTable[index] = key;

curr\_size++;

}

// function to display the hash table

void displayHash()

{

for (int i = 0; i < TABLE\_SIZE; i++)

{

if (hashTable[i] != -1)

cout << i << " --> "

<< hashTable[i] << endl;

else

cout << i << endl;

}

}

};

// Driver's code

int main()

{

int a[] = {19, 27, 36, 10, 64};

int n = sizeof(a)/sizeof(a[0]);

cout<<"inserted elements in hash table are as follows:";

// inserting keys into hash table

DoubleHash h;

for (int i = 0; i < n; i++)

h.insertHash(a[i]);

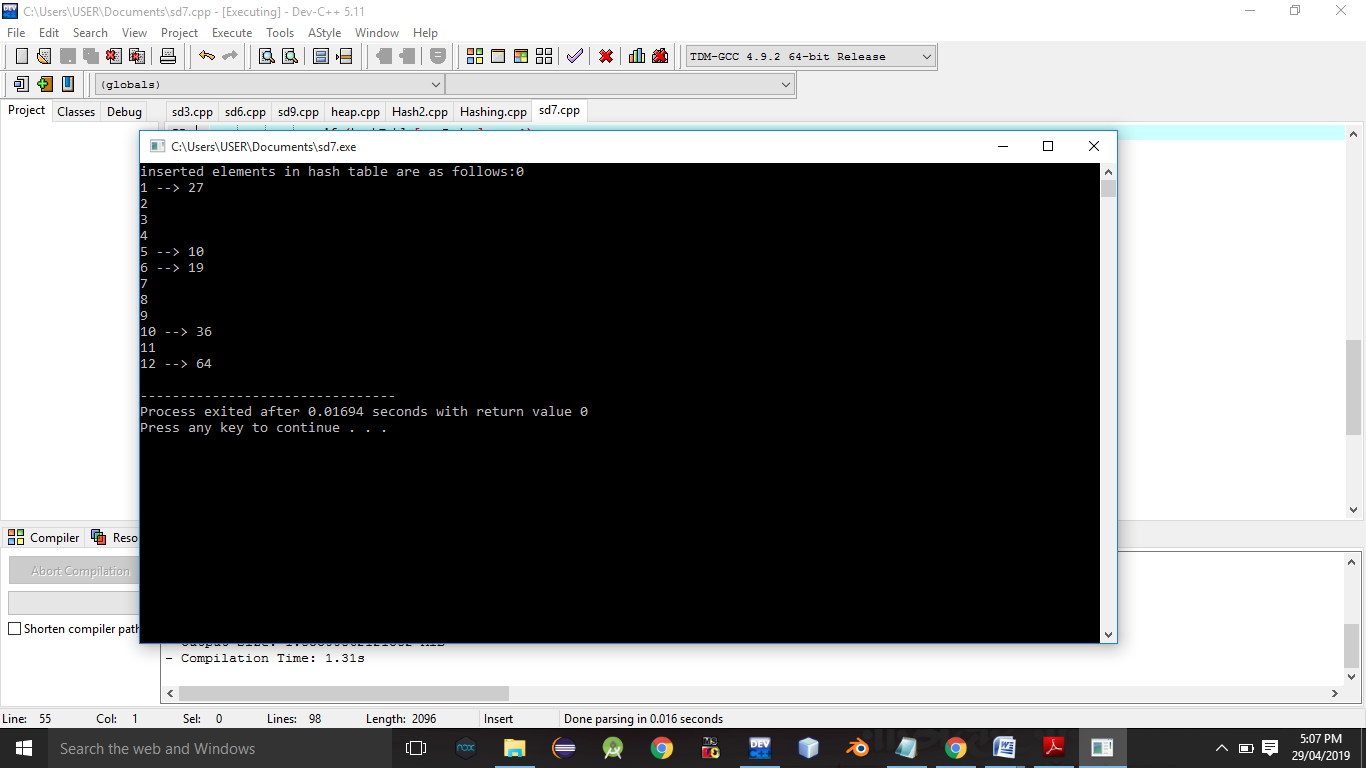
// display the hash Table

h.displayHash();

return 0;

}

**Output Screenshots:-**

****

**Conclusion:-** Thus,we have studied double hashing and hashing technique.