Practical – 28

AIM: In an array of 20 elements, arrange 15 different values, which are generated randomly between 1,00,000 to 9,99,999. Use hash function to generate key using linear probing, quadratic probing and double hashing to avoid collision.

H(k) = 2k + 3 and m = 20.

Write a program to input and display the final values of array.

* Program

#include<bits/stdc++.h>

using namespace std;

vector<long int> hashArr(20, 0);

int noOfCollision = 0;

void insertData(long int, int);

int hashIndex(long int, int);

int main(){

int lb = 100000, ub = 999999;

for(int i = 0; i < 20; i++){

hashArr[i] = 0;

}

for(int i = 0; i < 15; i++){

long int randomNum = (rand() % (ub-lb+1)) + lb;

cout << i+1 << " : " << randomNum << endl;

insertData(randomNum, 0);

}

cout << endl << "After hashing " << endl;

for(int i = 0; i < 20; i++){

cout << i << " " << hashArr[i] << endl;

}

cout << "Number of collision : " << noOfCollision << endl;

}

void insertData(long int val, int i){

int index = 0;

do{

index = (hashIndex(val, i++)) % 20;

}while(hashArr[index] != 0);

hashArr[index] = val;

}

int hashIndex(long int val, int i){

if(i > 0){

noOfCollision++;

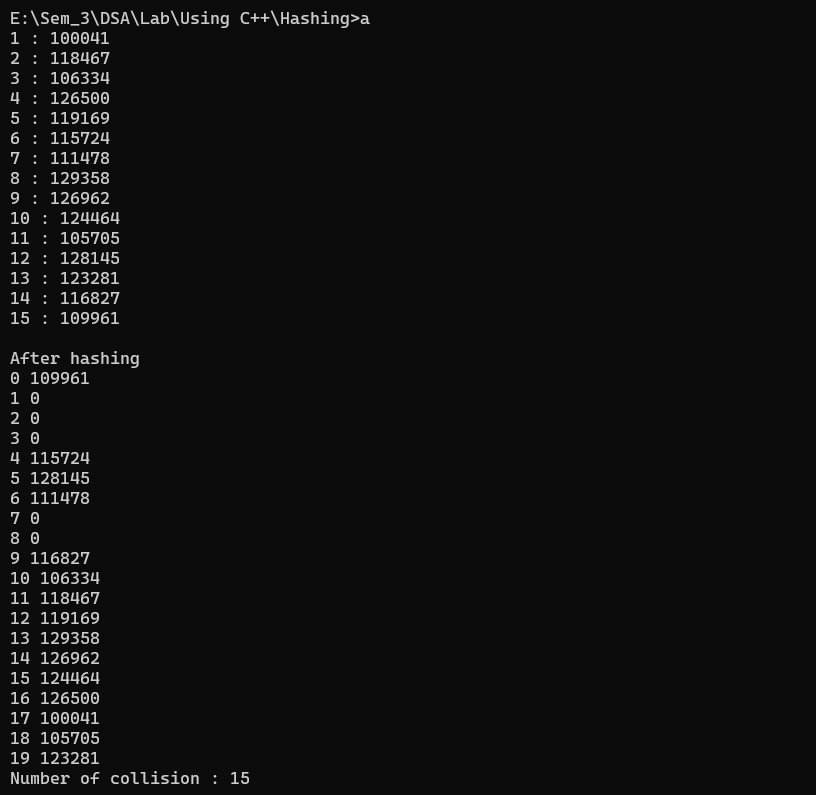
}

int h = (val % 18) + 2 + i;

return h;

}

* Output



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

Student Signature Faculty Signature Marks