**15. Write a C program to implement hashing using Linear Probing method**

#include <stdio.h>

#include <stdlib.h>

#define SIZE 10

int hashTable[SIZE];

void initializeHashTable() {

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

hashTable[i] = -1; // -1 indicates empty slot

}

}

int hashFunction(int key) {

return key % SIZE;

}

void insert(int key) {

int index = hashFunction(key);

int originalIndex = index;

while(hashTable[index] != -1) {

index = (index + 1) % SIZE;

if(index == originalIndex) {

printf("Hash table is full! Cannot insert %d\n", key);

return;

}

}

hashTable[index] = key;

printf("Inserted %d at index %d\n", key, index);

}

void search(int key) {

int index = hashFunction(key);

int originalIndex = index;

while(hashTable[index] != key) {

if(hashTable[index] == -1) {

printf("%d not found in hash table\n", key);

return;

}

index = (index + 1) % SIZE;

if(index == originalIndex) {

printf("%d not found in hash table\n", key);

return;

}

}

printf("%d found at index %d\n", key, index);

}

void display() {

printf("Hash Table:\n");

printf("Index\tValue\n");

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

printf("%d\t", i);

if(hashTable[i] == -1) {

printf("Empty\n");

} else {

printf("%d\n", hashTable[i]);

}

}

}

int main() {

initializeHashTable();

int choice, key;

while(1) {

printf("\nHashing with Linear Probing:\n");

printf("1. Insert\n2. Search\n3. Display\n4. Exit\n");

printf("Enter choice: ");

scanf("%d", &choice);

switch(choice) {

case 1:

printf("Enter key to insert: ");

scanf("%d", &key);

insert(key);

break;

case 2:

printf("Enter key to search: ");

scanf("%d", &key);

search(key);

break;

case 3:

display();

break;

case 4:

exit(0);

default:

printf("Invalid choice!\n");

}

}

return 0;

}