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

#include <stdio.h>

#define SIZE 10

int hashTable[SIZE];

void init() {

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

hashTable[i] = -1;

}

}

int hash(int key) {

return key % SIZE;

}

void insert(int key) {

int index = hash(key);

int i = 0;

while (hashTable[(index + i) % SIZE] != -1 && i < SIZE) {

i++;

}

if (i == SIZE) {

printf("Hash Table Full\n");

return;

}

hashTable[(index + i) % SIZE] = key;

}

int search(int key) {

int index = hash(key);

int i = 0;

while (hashTable[(index + i) % SIZE] != -1 && i < SIZE) {

if (hashTable[(index + i) % SIZE] == key)

return (index + i) % SIZE;

i++;

}

return -1;

}

void display() {

printf("\nHash Table:\n");

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

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

}

}

int main() {

int choice, key, pos;

init();

while (1) {

printf("\n1. Insert\n2. Search\n3. Display\n4. Exit\nEnter 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);

pos = search(key);

if (pos != -1)

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

else

printf("Key %d not found\n", key);

break;

case 3:

display();

break;

case 4:

return 0;

default:

printf("Invalid Choice\n");

}

}

}

