```c
//Created By Ritwik Chandra Pandey
//On 4th Nov
//Collision Resolution Techniques: Linear Probing


#include <stdio.h>
#include <stdlib.h>

#define SIZE 10
int HashTable[SIZE];
int hash(int x) {
        return x % SIZE;
}
void insert(int x) {
        int index, start;
        index = hash(x);
        start = index;
        while(HashTable[index]!=-1){
                if(HashTable[index]==-1){
                        break;
                }
                index = (index+1)%SIZE;
                if(index == start){
                        printf("Hash table is full. So cannot insert the element.\n");
                        return;
                }
        }
        HashTable[index] = x;
        printf("Successfully inserted.\n");
}
void delete(int x) {
        int index, start;
        index = hash(x);
        start = index;
        while(HashTable[index]!=x){
                if(HashTable[index]==x){
                        break;
                }
                index = (index+1)%SIZE;
```

```c
                if(index==start){
                        printf("Element not found. So cannot delete the element.\n");
                        return;
                }
        }
        HashTable[index]=-1;
        printf("Successfully deleted.\n");

}
void search(int x) {
        int index, start;
        index = hash(x);
        start = index;
        while(HashTable[index]!=x){
                if(HashTable[index]==x){
                        break;
                }
                index = (index+1)%SIZE;
                if(index==start){
                        printf("Element not found.\n");
                        return;
                }
        }
        printf("Element found.\n");
}
void print() {
        int i;
        for(i=0; i<SIZE;i++){
                if(HashTable[i]!=-1){
                        printf("[%d]=>%d ",i,HashTable[i]);
                }
        }
        printf("\n");
}


int main() {
        int x, op, i = 0;
        for (i = 0; i < SIZE; i++)
```

```c
            HashTable[i] = -1;
    while (1) {
            printf("1.Insert 2.Delete 3.Search 4.Print 5.Exit\n");
            printf("Enter your option : ");
            scanf("%d", &op);
            switch (op) {
                    case 1:printf("Enter an element to be inserted : ");
                                    scanf("%d", &x);
                                    insert(x);
                                    break;
                    case 2:
                                    printf("Enter an element to be deleted : ");
                                    scanf("%d", &x);
                                    delete(x);
                                    break;
                    case 3:
                                    printf("Enter an element to be searched : ");
                                    scanf("%d", &x);
                                    search(x);
                                    break;
                    case 4:
                                    print();
                                    break;
                    case 5:exit(0);
            }
    }
}
```