```
//Created By Ritwik Chandra Pandey
//On 4th Nov
//Collision Resolution Techniques: Separate Chaining

#include<stdio.h>
#include<conio.h>


#define SIZE 10

struct node {
	int data;
	struct node * next;
};
struct node * HashTable[SIZE];

int hash(int x) {
	return x % SIZE;
}

struct node * newNode(int x) {
	struct node * temp = (struct node *) malloc(sizeof(struct node *));
	temp->data = x;
	temp->next = NULL;
}
void insert(int x) {
	int index = hash(x);
	struct node* temp;
	temp->next = HashTable[index];
	HashTable[index] = temp;
}
void delete(int x) {
	int index= hash(x);
	struct node* temp;
	struct node* prev;
	temp = HashTable[index];
	if(temp==NULL){
		printf("Element not found. So cannot delete.\n");
		return;
	}else if(temp->data == x){
```

```
                HashTable[index] = temp->next;
                free(temp);
                printf("Successfully deleted.\n");
                return;

        }else{
                while(temp!=NULL && temp->data==x){
                        prev = temp;
                        temp = temp->next;

                }
                if(temp==NULL){
                        printf("Element not found. So cannot delete.\n");
                        return;
                }else{
                        prev->next = temp->next;
                        free(temp);
                        printf("Successfully deleted.\n");
                        return;
                }
        }
}

void search(int x) {
        int index = hash(x);
        struct node* temp;
        temp = HashTable[index];
        while(temp!=NULL && temp->data==x){
                temp = temp->next;
        }
        if(temp==NULL){
                printf("Element not found.\n");
                return;
        }else if(temp->data == x){
                printf("Element found.\n");
        }
}
void print() {
        for(int i=0;i<SIZE;i++){
                if(HashTable[i]!=NULL){
```

```
                struct node* temp;
                temp = HashTable[i];
                printf("[%d]=>",i);
                while(temp!=NULL){
                        printf("%d ",temp->data);
                }
        }
    }
    printf("\n");
}


int main() {
    int x, op, i=0;
    for(i=0;i<SIZE;i++)
        HashTable[i]=NULL;
    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);
        }
```

```
        }
}
```