Hashing Implementation

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

struct hash *hashTable= NULL;
int eleCount=0;

struct node {
int key, age;
char name[100];
struct node *next;
};

struct hash {
struct node *head;
int count;
};

struct node *createNode(int key,char*name,int age){
struct node *newnode;
newnode=(struct node *)malloc(sizeof(struct node));
newnode->key = key;
newnode->age = age;
strcpy(newnode->name, name);
newnode->next = NULL;
returnnewnode;
}

Void insertToHash(int key,char*name,int age){
int hashIndex= key %eleCount;
struct node *newnode=createNode(key, name, age);
/* head of list for the bucket with index "hashIndex" */
if(!hashTable[hashIndex].head){
hashTable[hashIndex].head=newnode;
hashTable[hashIndex].count=1;
return;
}
/* adding new node to the list */
newnode->next =(hashTable[hashIndex].head);
/*
     * update the head of the list and no of
     * nodes in the current bucket
```

```
43.      */
44. hashTable[hashIndex].head=newnode;
45. hashTable[hashIndex].count++;
46. return;
47. }
48.
49. Void deleteFromHash(int key){
50. /* find the bucket using hash index */
51. Int hashIndex= key %eleCount, flag =0;
52. struct node *temp,*myNode;
53. /* get the list head from current bucket */
54. myNode=hashTable[hashIndex].head;
55. if(!myNode){
56. printf("Given data is not present in hash Table!!\n");
57. return;
58. }
59.     temp =myNode;
60. while(myNode!= NULL){
61. /* delete the node with given key */
62. if(myNode->key == key){
63.             flag =1;
64. if(myNode==hashTable[hashIndex].head)
65. hashTable[hashIndex].head=myNode->next;
66. else
67.                 temp->next =myNode->next;
68.
69. hashTable[hashIndex].count--;
70. free(myNode);
71. break;
72. }
73.         temp =myNode;
74. myNode=myNode->next;
75. }
76. if(flag)
77. printf("Data deleted successfully from Hash Table\n");
78. else
79. printf("Given data is not present in hash Table!!!!\n");
80. return;
81. }
82.
83. Void searchInHash(int key){
84. Int hashIndex= key %eleCount, flag =0;
85. struct node *myNode;
```

```
myNode=hashTable[hashIndex].head;
if(!myNode){
printf("Search element unavailable in hash table\n");
return;
}
while(myNode!= NULL){
if(myNode->key == key){
printf("VoterID  : %d\n",myNode->key);
printf("Name     : %s\n",myNode->name);
printf("Age      : %d\n",myNode->age);
            flag =1;
break;
}
myNode=myNode->next;
      }
      if(!flag)
      printf("Search element unavailable in hash table\n");
      return;
      }

      void display(){
      struct node *myNode;
      inti;
      for(i=0;i<eleCount;i++){
      if(hashTable[i].count==0)
      continue;
      myNode=hashTable[i].head;
      if(!myNode)
      continue;
      printf("\nData at index %d in Hash Table:\n",i);
      printf("VoterID     Name          Age   \n");
      printf("--------------------------------\n");
      while(myNode!= NULL){
      printf("%-12d",myNode->key);
      printf("%-15s",myNode->name);
      printf("%d\n",myNode->age);
      myNode=myNode->next;
      }
      }
      return;
      }

```

```
129.      int main(){
130.      int n,ch, key, age;
131.      char name[100];
132.      printf("Enter the number of elements:");
133.      scanf("%d",&n);
134.      eleCount= n;
135.      /* create hash table with "n" no of buckets */
136.      hashTable=(struct hash *)calloc(n,sizeof(struct hash));
137.      while(1){
138.      printf("\n1. Insertion\t2. Deletion\n");
139.      printf("3. Searching\t4. Display\n5. Exit\n");
140.      printf("Enter your choice:");
141.      scanf("%d",&ch);
142.      switch(ch){
143.      case1:
144.      printf("Enter the key value:");
145.      scanf("%d",&key);
146.      getchar();
147.      printf("Name:");
148.      fgets(name,100,stdin);
149.                  name[strlen(name)-1]='\0';
150.      printf("Age:");
151.      scanf("%d",&age);
152.      /*inserting new node to hash table */
153.      insertToHash(key, name, age);
154.      break;
155.
156.      case2:
157.      printf("Enter the key to perform deletion:");
158.      scanf("%d",&key);
159.      /* delete node with "key" from hash table */
160.      deleteFromHash(key);
161.      break;
162.
163.      case3:
164.      printf("Enter the key to search:");
165.      scanf("%d",&key);
166.      searchInHash(key);
167.      break;
168.      case4:
169.                  display();
170.      break;
171.      case5:
```

```
172.        exit(0);
173.        default:
174.        printf("U have entered wrong option!!\n");
175.        break;
176.        }
177.        }
178.        return0;
179.        }
```

Output:

```
$ gccHashTablesLL.c
$ ./a.out

  Enter the number of elements:3

  1. Insertion 2. Deletion
  3. Searching 4. Display
  5. Exit
  Enter your choice:1
  Enter the key value:3
Name:Sally
  Age:23

  1. Insertion 2. Deletion
  3. Searching 4. Display
  5. Exit
  Enter your choice:1
  Enter the key value:33
Name:Harry
  Age:25

  1. Insertion 2. Deletion
  3. Searching 4. Display
  5. Exit
  Enter your choice:1
  Enter the key value:7
Name:Nick
  Age:30

  1. Insertion 2. Deletion
  3. Searching 4. Display
  5. Exit
  Enter your choice:1
  Enter the key value:35
Name:Raj
  Age:28

  1. Insertion 2. Deletion
  3. Searching 4. Display
  5. Exit
  Enter your choice:4
```

```
  Data at index 0 in Hash Table:
VoterID     Name          Age
  ------------------------------
  33          Harry          25
  3           Sally          23

  Data at index 1 in Hash Table:
VoterID     Name          Age
  ------------------------------
  7           Nick           30

  Data at index 2 in Hash Table:
VoterID     Name          Age
  ------------------------------
  35          Raj            28

  1. Insertion 2. Deletion
  3. Searching 4. Display
  5. Exit
  Enter your choice:2
  Enter the key to perform deletion:33
  Data deleted successfully from Hash Table

  1. Insertion 2. Deletion
  3. Searching 4. Display
  5. Exit
  Enter your choice:4

  Data at index 0 in Hash Table:
VoterID     Name          Age
  ------------------------------
  3           Sally          23

  Data at index 1 in Hash Table:
VoterID     Name          Age
  ------------------------------
  7           Nick           30

  Data at index 2 in Hash Table:
VoterID     Name          Age
  ------------------------------
  35          Raj            28

  1. Insertion 2. Deletion
  3. Searching 4. Display
  5. Exit
  Enter your choice:3
  Enter the key to search:35
VoterID  : 35
  Name     : Raj
  Age      : 28

  1. Insertion 2. Deletion
  3. Searching 4. Display
  5. Exit
```

```
Enter your choice:5
```