

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

```
#include <stdio.h>
#include<stdlib.h>
#define TABLE_SIZE 10

int h[TABLE_SIZE]={NULL};

void insert()
{
    int key,index,i,flag=0,hkey;
    printf("\nEnter a value to insert into hash table\n");
    scanf("%d",&key);
    hkey=key%TABLE_SIZE;
    for(i=0;i<TABLE_SIZE;i++)
    {
        index=(hkey+i)%TABLE_SIZE;

        if(h[index] == NULL)
        {
            h[index]=key;
            break;
        }
    }

    if(i == TABLE_SIZE)

        printf("\nElement cannot be inserted\n");
}

void search()
{
    int key,index,i,flag=0,hkey;
    printf("\nEnter search element\n");
    scanf("%d",&key);
    hkey=key%TABLE_SIZE;
    for(i=0;i<TABLE_SIZE; i++)
    {
        index=(hkey+i)%TABLE_SIZE;
        if(h[index]==key)
        {
            printf("value is found at index %d",index);
        }
    }
}
```

```

        break;
    }
}
if(i == TABLE_SIZE)
    printf("\n value is not found\n");
}
void display()
{

    int i;

    printf("\nelements in the hash table are \n");

    for(i=0;i< TABLE_SIZE; i++)

        printf("\nat index %d \t value = %d",i,h[i]);

}
main()
{
    int opt,i;
    while(1)
    {
        printf("\nPress 1. Insert\t 2. Display \t3. Search \t4.Exit \n");
        scanf("%d",&opt);
        switch(opt)
        {
            case 1:
                insert();
                break;
            case 2:
                display();
                break;
            case 3:
                search();
                break;
            case 4:exit(0);
        }
    }
}

```

```
D:\data structures lab\hashing using linear probing.c - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
[enque,deque,display.c vafes.h inorder,preorder,postorder.c hashing using linear probing.c]
1 // 15.. write a C program to implement hashing using Linear Probing method
2 #include <stdio.h>
3 #include <stdlib.h>
4 #define TABLE_SIZE 10
5
6 int h[TABLE_SIZE]={NULL};
7
8 void insert()
9 {
10
11 int key,index,i,flag=0,hkey;
12 printf("\nEnter a value to insert into hash table\n");
13 scanf("%d",&key);
14 hkey=key%TABLE_SIZE;
15 for(i=0;i<TABLE_SIZE;i++)
16 {
17 index=(hkey+i)%TABLE_SIZE;
18 if(h[index] == NULL)
19 {
20 h[index]=key;
21 break;
22 }
23 }
24
25 if(i == TABLE_SIZE)
26 printf("\nElement cannot be inserted\n");
27 }
28
29 void search()
30 {
31
32 int key,index,i,flag=0,hkey;
33 printf("\nEnter search element\n");
34 scanf("%d",&key);
35 hkey=key%TABLE_SIZE;
36 for(i=0;i<TABLE_SIZE;i++)
37 {
38 index=(hkey+i)%TABLE_SIZE;
39 if(h[index]==key)
40 {
41 printf("value is found at index %d",index);
42 }
43 }
44 }
```

Line: 58 Col: 30 Sel: 0 Lines: 84 Length: 1557 Insert Done parsing in 0.016 seconds

```
D:\data structures lab\hashing using linear probing.c - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
[enque,deque,display.c vafes.h inorder,preorder,postorder.c hashing using linear probing.c]
41 index=(hkey+i)%TABLE_SIZE;
42 if(h[index]==key)
43 {
44 printf("value is found at index %d",index);
45 break;
46 }
47 if(i == TABLE_SIZE)
48 printf("\n value is not found\n");
49 }
50 void display()
51 {
52
53 int i;
54
55 printf("\nelements in the hash table are \n");
56
57 for(i=0;i< TABLE_SIZE; i++)
58 {
59 printf("\nat index %d \t value = %d",i,h[i]);
60 }
61 }
62
63 main()
64 {
65 int opt,i;
66 while(1)
67 {
68 printf("\nPress 1. Insert \t 2. Display \t 3. Search \t 4.Exit\n");
69 scanf("%d",&opt);
70 switch(opt)
71 {
72 case 1:
73 insert();
74 break;
75 case 2:
76 display();
77 break;
78 case 3:
79 search();
80 break;
81 case 4:
82 exit(0);
83 }
84 }
```

Line: 58 Col: 30 Sel: 0 Lines: 84 Length: 1557 Insert Done parsing in 0.016 seconds

```
Press 1. Insert 2. Display 3. Search 4.Exit
1
Enter a value to insert into hash table
10
Press 1. Insert 2. Display 3. Search 4.Exit
1
Enter a value to insert into hash table
20
Press 1. Insert 2. Display 3. Search 4.Exit
1
Enter a value to insert into hash table
30
Press 1. Insert 2. Display 3. Search 4.Exit
2
elements in the hash table are
at index 0 value = 10
at index 1 value = 20
at index 2 value = 30
at index 3 value = 0
at index 4 value = 0
at index 5 value = 0
at index 6 value = 0
at index 7 value = 0
at index 8 value = 0
at index 9 value = 0
Press 1. Insert 2. Display 3. Search 4.Exit
3
Enter search element
20
value is found at index 1
Press 1. Insert 2. Display 3. Search 4.Exit
4
Process exited after 46.64 seconds with return value 0
Press any key to continue . . .
```