

```

1: //Display a Linked List
2:
3: #include <stdio.h>
4: #include <stdlib.h>
5:
6: struct Node
7: {
8:     int data;
9:     struct Node *next;
10: }*first=NULL;
11:
12: void create(int A[],int n)
13: {
14:     int i;
15:     struct Node *t,*last;
16:     first=(struct Node *)malloc(sizeof(struct Node));
17:     first->data=A[0];
18:     first->next=NULL;
19:     last=first;
20:
21:     for(i=1;i<n;i++)
22:     {
23:         t=(struct Node*)malloc(sizeof(struct Node));
24:         t->data=A[i];
25:         t->next=NULL;
26:         last->next=t;
27:         last=t;
28:     }
29: }
30:
31: struct Node * LSearch(struct Node *p,int key)
32: {
33:     struct Node *q;
34:
35:     while(p!=NULL)
36:     {
37:         if(key==p->data)
38:         {
39:             q->next=p->next;

```

```
40: p->next=first;
41: first=p;
42: return p;
43: }
44: q=p;
45: p=p->next;
46: }
47: return NULL;
48:
49: }
50:
51: struct Node * RSearch(struct Node *p,int key)
52: {
53:     if(p==NULL)
54:         return NULL;
55:     if(key==p->data)
56:         return p;
57:     return RSearch(p->next,key);
58:
59: }
60:
61: int main()
62: {
63:     struct Node *temp;
64:     int A[]={3,5,7,10,25,8,32,2};
65:     create(A,8);
66:     temp=Search(first,3);
67:     printf("%d",temp->data);
68:
69:     return 0;
70: }
```