

```

1: //Count and Sum 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: int count(struct Node *p)
32: {
33:     int l=0;
34:     while(p)
35:     {
36:         l++;
37:         p=p->next;
38:     }
39:     return l;

```

```
40: }
41:
42: int Rcount(struct Node *p)
43: {
44:     if(p!=NULL)
45:         return Rcount(p->next)+1;
46:     else
47:         return 0;
48: }
49:
50: int sum(struct Node *p)
51: {
52:     int s=0;
53:
54:     while(p!=NULL)
55:     {
56:         s+=p->data;
57:         p=p->next;
58:     }
59:     return s;
60: }
61:
62: int Rsum(struct Node *p)
63: {
64:     if(p==NULL)
65:         return 0;
66:     else
67:         return Rsum(p->next)+p->data;
68: }
69:
70: int main()
71: {
72:     int A[]={3,5,7,10,25,8,32,2};
73:     create(A,8);
74:
75:     printf("count '%d\n'",count(first));
76:     printf("Sum '%d\n'",sum(first));
77:     return 0;
78: }
```