Linked List

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

struct Node
{
    int data;
    struct Node *next;
}*first=NULL;

void create(int A[],int n)
{
    int i;
    struct Node *t,*last;
    first=(struct Node *)malloc(sizeof(struct Node));
    first->data=A[0];
    first->next=NULL;
    last=first;

    for(i=1;i<n;i++)
    {
        t=(struct Node*)malloc(sizeof(struct Node));
        t->data=A[i];
        t->next=NULL;
        last->next=t;
        last=t;
    }
}

void Display(struct Node *p)
{
    while(p!=NULL)
    {
        printf("%d ",p->data);
        p=p->next;
    }
```

```
}

void RDisplay(struct Node *p)
{
    if(p!=NULL)
    {
        RDisplay(p->next);
        printf("%d ",p->data);

    }
}

int count(struct Node *p)
{
    int l=0;
    while(p)
    {
        l++;
        p=p->next;
    }
    return l;
}

int Rcount(struct Node *p)
{
    if(p!=NULL)
        return Rcount(p->next)+1;
    else
        return 0;
}

int sum(struct Node *p)
{
    int s=0;

    while(p!=NULL)
    {
        s+=p->data;
        p=p->next;
    }
```

```
    return s;
}

int Rsum(struct Node *p)
{
    if(p==NULL)
        return 0;
    else
        return Rsum(p->next)+p->data;
}

int Max(struct Node *p)
{
    int max=INT32_MIN;

    while(p)
    {
        if(p->data>max)
            max=p->data;
        p=p->next;
    }
    return max;

}

int RMax(struct Node *p)
{
    int x=0;

    if(p==0)
        return INT32_MIN;
    x=RMax(p->next);
    if(x>p->data)
        return x;
    else
        return p->data;
}

struct Node * LSearch(struct Node *p,int key)
{
```

```
    struct Node *q;

    while(p!=NULL)
    {
        if(key==p->data)
        {
            q->next=p->next;
            p->next=first;
            first=p;
            return p;
        }
        q=p;
        p=p->next;
    }
    return NULL;

}

struct Node * RSearch(struct Node *p,int key)
{
    if(p==NULL)
        return NULL;
    if(key==p->data)
        return p;
    return RSearch(p->next,key);

}


int main()
{
    struct Node *temp;
    int A[]={3,5,7,10,25,8,32,2};
    create(A,8);

    temp=LSearch(first,25);
    temp=LSearch(first,8);
    if(temp)
        printf("Key is Found %d\n",temp->data);
    else
```

```
        printf("Key not found\n");

    Display(first);

    return 0;
}
```