-_

//linklsit add display and sum

```
#include <stdio.h>
#include<stdlib.h>
struct Node
{

  int data;

  struct Node *next;

};

void display(struct Node*);
int R_SUM(struct Node*);
int SUM(struct Node*);

int main()

{
 struct Node *head,*n1,*n2;
 head=(struct Node *)malloc(sizeof(struct Node));
 n1=(struct Node *)malloc(sizeof(struct Node));
 n2=(struct Node *)malloc(sizeof(struct Node));
 head->data=10;
 head->next=n1;
 n1->data=20;
 n1->next=n2;
 n2->data=40;
 n2->next=NULL;
 display(head);
 int ansR=R_SUM(head);
 printf("\nsum of elemnts with recursion is %d\n",ansR);
 int ans=R_SUM(head);
 printf("\nsum of elemnts with recursion is %d\n",ans);
 return 0;

}

void display(struct Node *p){
  if(p!=NULL){   //Base condition
    printf("%d->",p->data);
    display(p->next);//recursion

  }
}
int R_SUM(struct Node *p){
  if(p!=NULL){   //Base condition

    return p->data+R_SUM(p->next);

  }

}
int SUM(struct Node *p){
  int sum=0;
```

```
   while(p!=NULL){    //Base condition
     sum=p->data+sum;
     p=p->next;
   }
   return sum;
}
```

-_-

//**link list search**

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

typedef struct node{
   int data;
   struct node *next;
}NODE;

void display(NODE*);
NODE *Lsearch(NODE*,int);

int main(){

   NODE *head=(NODE*)malloc(sizeof(NODE));
   head->data=10;
   NODE *N1=(NODE*)malloc(sizeof(NODE));
   head->next=N1;
   N1->data=20;
   NODE *N2=(NODE*)malloc(sizeof(NODE));
   N1->next=N2;
   N2->data=30;
   N2->next=NULL;
   display(head);
   NODE *srch=Lsearch(head,20);
   printf("\n%d",*srch);
   return 0;
}

void display(NODE *ptr){
   if(ptr != NULL){
      printf("%d->",ptr->data);
      display(ptr->next);
   }
}

NODE *Lsearch(NODE *ptr,int key){
   while(ptr!=NULL){
      if(key == ptr->data){
         return ptr;
      }
      ptr=ptr->next;
   }
   printf("\nnot found\n");
   return NULL;

}
```

-_

// **linklist insert**

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

typedef struct node{
   int data;
   struct node *next;
}NODE;

NODE *head=NULL;// defining globally bcoz any change in the node using fn will no be reflected in the main;

void display(NODE*);
NODE *Lsearch(NODE*,int);
void insert(NODE*,int,int);

int main(){

   head=(NODE*)malloc(sizeof(NODE));
   head->data=10;
   NODE *N1=(NODE*)malloc(sizeof(NODE));
   head->next=N1;
   N1->data=20;
   NODE *N2=(NODE*)malloc(sizeof(NODE));
   N1->next=N2;
   N2->data=30;
   N2->next=NULL;
   display(head);
   NODE *srch=Lsearch(head,20);
   printf("\n%d\n",*srch);
   insert(head,3,15);
   display(head);
   return 0;
}

int Ncount(NODE *ptr){
   int count=0;
   while(ptr!=NULL){
      count++;
      ptr=ptr->next;
   }
   return count;
}
void display(NODE *ptr){
   if(ptr != NULL){
      printf("%d->",ptr->data);
      display(ptr->next);
   }
}

NODE *Lsearch(NODE *ptr,int key){
   while(ptr!=NULL){
      if(key == ptr->data){
         return ptr;
      }
      ptr=ptr->next;
```

```
    }
    printf("\nnot found\n");
    return NULL;

}
void insert(NODE *ptr,int index,int x){
    NODE *new;
    int i;
    if(index<0||index>Ncount(ptr)){
        printf("invalid position\n");
    }
    new=(NODE*)malloc(sizeof(NODE));
    new->data=x;
    if(index==0){
        new->next=head;
        head=new;
    }
    else{
        for(i=0;i<index-1;i++){
            ptr=ptr->next;
        }
        new->next=ptr->next;
        ptr->next=new;
    }
}
```

-_

//**<u>linklist create node using function</u>**

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

typedef struct node{
    int data;
    struct node *next;
}NODE;

NODE *head=NULL;// defining globally bcoz any change in the node using fn will no be reflected in the main;

void display(NODE*);
NODE *Lsearch(NODE*,int);
void insert(NODE*,int,int);
void create(int *,int);

int main(){

    int A[]={10,20,30,40,50};
    create(A,5);
    display(head);
    NODE *srch=Lsearch(head,20);
    printf("\n%d\n",*srch);
    insert(head,3,15);
    display(head);
    return 0;
}

void create(int *arr,int n){
```

```
  NODE *t,*last;
  head=(NODE*)malloc(sizeof(NODE));
  head->data=arr[0];
  head->next=NULL;
  last=head;
  for(int i=1;i<n;i++){
    t=(NODE*)malloc(sizeof(NODE));
    t->data=arr[i];
    t->next=NULL;
    last->next=t;
    last=t;

  }
}
int Ncount(NODE *ptr){
  int count=0;
  while(ptr!=NULL){
    count++;
    ptr=ptr->next;
  }
  return count;
}
void display(NODE *ptr){
  if(ptr != NULL){
    printf("%d->",ptr->data);
    display(ptr->next);
  }
}

NODE *Lsearch(NODE *ptr,int key){
  while(ptr!=NULL){
    if(key == ptr->data){
      return ptr;
    }
    ptr=ptr->next;
  }
  printf("\nnot found\n");
  return NULL;

}
void insert(NODE *ptr,int index,int x){
  NODE *new;
  int i;
  if(index<0||index>Ncount(ptr)){
    printf("invalid position\n");
  }
  new=(NODE*)malloc(sizeof(NODE));
  new->data=x;
  if(index==0){
    new->next=head;
    head=new;
  }
  else{
    for(i=0;i<index-1;i++){
      ptr=ptr->next;
    }
```

```
    new->next=ptr->next;
    ptr->next=new;
  }
}
```
