Practical No.5

Create two separate Positive and Negative List form the Inputted List

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
typedef struct linklist
{
    int data;
    struct node *next;
}Node;
Node *creatlist(Node *,int );
void PostiveNumber(Node**,Node*);
void NegativeNumber(Node**,Node*);
void Display(Node *);
int main()
{
 int c;
 printf("Number of elements to be inputed : ");
 scanf("%d",&c);
 printf("\n\t\t\tEnter Positive and Negative Numbers\n ");
 Node *head1=creatlist(NULL,c);
 Node *p=head1,*pos=NULL,*neg=NULL;
 Node *t1,*t2;
 printf("\nList\n");
 Display(head1);
 PostiveNumber(&pos,head1);
 printf("\nList Of Positive Numbers\n");
 Display(pos);
 NegativeNumber(&neg,head1);
  printf("\nList of Negative numbers\n");
 Display(neg);
}
Node *creatlist(Node *head,int co)
{
    Node * ptr=head;
while(co>0)
    {
       Node* node = (Node*)malloc(sizeof(Node));
        printf("\nEnter the element : ");
        scanf("%d",&node->data);
        node->next=NULL;
        if(head==NULL)
        {
            head=node;
            co--;
        }
        else
        {
            ptr=head;
            while(ptr->next!=NULL)
                ptr=ptr->next;
            ptr->next=node;
            co--;
        }
    }
    return head;
```

#AJ

```
}
void PostiveNumber(Node **pos,Node *head)
{
    Node *ptr=*pos;
    while(head!=NULL)
    {
        int n=head->data;

        if(n>0)
            {
                Node* node = (Node*)malloc(sizeof(Node));
                node->next=NULL;
                node->data=n;
                 if(*pos==NULL)
                      {
                        *pos=node;

                      }
                else
                      {
                        ptr=*pos;
                        while(ptr->next!=NULL)
                            ptr=ptr->next;
                        ptr->next=node;
                       }
            }
           head=head->next;
    }

}
void NegativeNumber(Node **pos,Node *head)
{
    Node *ptr=*pos;
    while(head!=NULL)
    {
        int n=head->data;

        if(n<0)
            {
                Node* node = (Node*)malloc(sizeof(Node));
                node->next=NULL;
                node->data=n;
                 if(*pos==NULL)
                      {
                        *pos=node;

                      }
                else
                      {
                        ptr=*pos;
                        while(ptr->next!=NULL)
                            ptr=ptr->next;
                        ptr->next=node;
                       }
```

```
            }
            head=head->next;
    }

}

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