Doubly Linked List

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


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

void create(int A[],int n)
{
    struct Node *t,*last;
    int i;

    first=(struct Node *)malloc(sizeof(struct Node));
    first->data=A[0];
    first->prev=first->next=NULL;
    last=first;

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

void Display(struct Node *p)
{
    while(p)
    {
```

```
        printf("%d ",p->data);
        p=p->next;
    }
    printf("\n");
}

int Length(struct Node *p)
{
    int len=0;

    while(p)
    {
        len++;
        p=p->next;
    }
    return len;
}

void Insert(struct Node *p,int index,int x)
{
    struct Node *t;
    int i;

    if(index < 0 || index > Length(p))
        return;
    if(index==0)
    {
        t=(struct Node *)malloc(sizeof(struct Node));
        t->data=x;
        t->prev=NULL;
        t->next=first;
        first->prev=t;
        first=t;
    }
    else
    {
        for(i=0;i<index-1;i++)
            p=p->next;
        t=(struct Node *)malloc(sizeof(struct Node));
        t->data=x;
```

```
        t->prev=p;
        t->next=p->next;
        if(p->next)p->next->prev=t;
        p->next=t;

    }
}

int Delete(struct Node *p,int index)
{
    //struct Node *q;
    int x=-1,i;

    if(index < 1 || index > Length(p))
        return -1;

    if(index==1)
    {
        first=first->next;
        if(first)first->prev=NULL;

        x=p->data;
        free(p);
    }
    else
    {
        for(i=0;i<index-1;i++)
            p=p->next;
        p->prev->next=p->next;
        if(p->next)
            p->next->prev=p->prev;
        x=p->data;
        free(p);
    }
    return x;

}

void Reverse(struct Node *p)
```

```
{
    struct Node *temp;

    while(p!=NULL)
    {
        temp=p->next;
        p->next=p->prev;
        p->prev=temp;
        p=p->prev;
        if(p!=NULL && p->next==NULL)
            first=p;
    }
}

int main()
{
    int A[]={10,20,30,40,50};
    create(A,5);

    Reverse(first);

    Display(first);

    return 0;
}
```