Reverse a Linked List

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

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

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



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 Reverse1(struct Node *p)
{
    int *A,i=0;
    struct Node *q=p;
    A=(int *)malloc(sizeof(int)*count(p));

    while(q!=NULL)
    {
        A[i]=q->data;
        q=q->next;
        i++;
    }
    q=p;
    i--;
    while(q!=NULL)
    {
        q->data=A[i];
        q=q->next;
        i--;
    }
}

void Reverse2(struct Node *p)
{
    struct Node *q=NULL,*r=NULL;

    while(p!=NULL)
    {
        r=q;
        q=p;
        p=p->next;
        q->next=r;
    }
    first=q;
}

void Reverse3(struct Node *q,struct Node *p)
{
    if(p)
    {
        Reverse3(p,p->next);
        p->next=q;
    }
    else
        first=q;
}


int main()
{
    int A[]={10,20,40,50,60};
    create(A,5);


    Reverse1(frist);
    Display(frist);

    return 0;
}
```