Circular Linked List

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

struct Node
{
    int data;
    struct Node *next;
}*Head;


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

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

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

```
    printf("\n");
}

void RDisplay(struct Node *h)
{
    static int flag=0;
    if(h!=Head || flag==0)
    {
        flag=1;
        printf("%d ",h->data);
        RDisplay(h->next);
    }
    flag=0;
}

int Length(struct Node *p)
{
    int len=0;
    do
    {
        len++;
        p=p->next;

    }while(p!=Head);
    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;
        if(Head==NULL)
        {
```

```
            Head=t;
            Head->next=Head;
        }
        else
        {
            while(p->next!=Head)p=p->next;
            p->next=t;
            t->next=Head;
            Head=t;
        }

    }
    else
    {
        for(i=0;i<index-1;i++)p=p->next;
        t=(struct Node *)malloc(sizeof(struct Node));
        t->data=x;
        t->next=p->next;
        p->next=t;

    }
}

int Delete(struct Node *p,int index)
{
    struct Node *q;
    int i,x;

    if(index <0 || index >Length(Head))
        return -1;
    if(index==1)
    {
        while(p->next!=Head)p=p->next;
        x=Head->data;
        if(Head==p)
        {
            free(Head);
            Head=NULL;
        }
        else
```

```
        {
            p->next=Head->next;
            free(Head);
            Head=p->next;
        }
    }
    else
    {
        for(i=0;i<index-2;i++)
            p=p->next;
        q=p->next;
        p->next=q->next;
        x=q->data;
        free(q);
    }
    return x;
}

int main()
{
    int A[]={2,3,4,5,6};
    create(A,5);

    Delete(Head,8);

    RDisplay(Head);
    return 0;
}
```