

```

#include <iostream>

using namespace std;

const int MinNumber=-1000000;

struct Node
{
    Node(int x)
    {
        value=x;
        before=NULL;
        next=NULL;
    }
    int value;
    Node* before;
    Node* next;
};

void Change(Node* a)
{
    Node *p=a;
    Node *second=a->next;
    Node *q=p->next->next;
    //从1到n,沿next路径
    while(q!=a&&q!=a->before)
    {
        p->next=q;
        //q->before=p;不能改before!!! 还要回来的
        p=q;
        q=q->next->next;
    }
    if(q==a)//总数为偶数
    {
        q=a->before;
    }
}

```

```

        p->next=q;
    }
else//总数为奇数
{
    p->next=q;

    p=q;

    q=q->before;

    p->next=q;
}

//经过以上操作，保证q为偶数
//从n到2,沿before路径
if(q!=second)//如果已经是第2个了，就不用再往前走了

    p=q->before->before;

else

    p=q;

while(p!=second)//一定会等于这个
{

    q->next=p;

    q=p;

    p=p->before->before;
}

if(p==second)//总数为偶数
{

    q->next=p;//把最后接上

    p->next=a;
}

//最后改造所有的before路径

p=a;

while(p->next!=a)
{

    q=p->next;

```

```

        q->before=p;

        p=p->next;

    }

    a->before=p;

}

int main()

{

    //由题目可得n>=3,先建立双向链表

    Node* ha=new Node(1);

    Node *p=ha;

    for(int i=2;i<=11;i++)

    {

        Node * t=new Node(i);

        p->next=t;

        t->before=p;

        p=t;

    }

    p->next=ha;

    ha->before=p;

    Change(ha);

    //正向输出

    p=ha;

    while(p->next!=ha)

    {

        cout<<p->value<<" ";

        p=p->next;

    }

    cout<<p->value<<endl;

    //逆向输出

    p=ha->before;

```

```
while(p!=ha)
{
    cout<<p->value<<" ";
    p=p->before;
}
cout<<p->value<<endl;
```

```
}
```