Problem-1

```
#include<iostream>
using namespace std;
struct LNode
{
    int data;
    LNode *next;
};
/*头插法，逆序输出*/（本题目采用此法）
void CreatList(LNode *head, int n)
{
    LNode *p;
    int x;
    head->next = NULL;
    for (int i = n; i > 0; i--)
    {
        p = new LNode;
        cin >> x;
        p->data = x;
        p->next = head->next;
        head->next = p;
    }
}
/*尾插法,正序输出*/
void CreatList(LNode *head)
{
    LNode *p, *q;
    int x;
    q = head;
    while (1)
    {
        cin >> x;
        if (x == 0)
            break;
        p = new LNode;
        p->data = x;
        q->next = p;
        q = p;
    }
    q->next = NULL;
}
void PrintList(LNode *head)
{
    LNode *p;
    p = head->next;
    while (p)
    {
        cout << p->data << " ";
        p = p->next;
    }
}
void InsertList(LNode *head, int n, int i, int x)
{
    LNode *p, *q;
    q = new LNode;
    int j = 0;
    p = head;
    while (j < i - 1 && p)
    {
        p = p->next;
        j++;
    }
    if (!p)
        cout << -1<<endl;
    else
    {
        q->data = x;
        q->next = p->next;
        p->next = q;
        PrintList(head);
        cout << endl;
    }
}
void DelList(LNode *head, int n, int j)
{
    LNode *p, *q;
    p = head;
    int i = 0;
    while (p->next&&i < j - 1)
    {
        p = p->next;
        i++;
    }
    if (!(p->next)
        cout << -1 << endl;
    else
    {
        q = p->next;
        p->next = q->next;
        PrintList(head);
        cout << endl;
    }
}
void SearchList(LNode *head, int y)
{
    LNode *p;
    p = head->next;
    int i = 1;
    while (p&&p->data != y)
    {
        p = p->next;
        i++;
    }
    if (p == NULL)
        cout << -1 << endl;
    else
        cout << i << endl;
}
int Length(LNode *head)
{
    LNode *p;
    p = head->next;
    int i = 0;
    while (p)
    {
        p = p->next;
        i++;
    }
    return i;
}
int main()
{
    LNode *head;
    head = new LNode;
    int n;
    cin >> n;
    CreatList(head, n);
    int i, x;
    cin >> i >> x;
    int j;
    cin >> j;
    int y;
    cin >> y;
    PrintList(head);
    cout << endl;
    InsertList(head, n, i, x);
    DelList(head, n, j);
    SearchList(head, y);
    cout << Length(head) << endl;
    return 0;
}
```

Problem-2

```
#include<iostream>
using namespace std;
struct LNode
{
    int data;
    LNode *next;
};
void CreatList(LNode *head, int n)
{
    LNode *p;
    int x;
    head->next = NULL;
    for (int i = n; i > 0; i--)
    {
        p = new LNode;
        cin >> x;
        p->data = x;
        p->next = head->next;
        head->next = p;
    }
}
void PrintList(LNode *head)
{
    LNode *p;
    p = head;
    p = p->next;
    while (p)
    {
        cout << p->data << ” ”;
        p = p->next;
    }
}
```

```
void Reverse(LNode *head)
{
    LNode *p, *q;
    p = head->next;
    head->next = NULL;
    while (p)
    {
        q = p;
        p = p->next;
        q->next = head->next;
        head->next = q;
    }
}
void Judge_3t(LNode *head)
{
    LNode *p;
    p = head->next;
    while (p)
    {
        if (p->data % 3 == 0)
            cout << p->data << " ";
        p = p->next;
    }
}
void destroy(LNode *head)
{
    LNode *p;
    p = head->next;
    while (p)
    {
        delete p;
        p = p->next;
    }
}
int main()
{
    LNode *head;
    head = new LNode;
    int n;
    cin >> n;
    CreatList(head, n);
    PrintList(head);
    cout << endl;
    Reverse(head);
    Judge_3t(head);
    return 0;
}
```

Problem-3

非正解但为值得思考的一种方法：

```
#include<iostream>
using namespace std;
struct LNode
{
    int data;
    LNode *next;
};
void PrintList(LNode *head)
{
    LNode *p;
    p = head;
    p = p->next;
    while (p)
    {
        cout << p->data << " ";
        p = p->next;
    }
}
void CreatList(LNode *head, int n)
{
    LNode *p;
    int x;
    head->next = NULL;
    for (int i = n; i > 0; i--)
    {
        p = new LNode;
        cin >> x;
        p->data = x;
        p->next = head->next;
        head->next = p;
    }
}
void Rev_same(LNode *head)
{
    LNode *p, *q, *r;
    p = head->next;
    head->next = NULL;
    while (p)
    {
        q = p;
        p = p->next;
        q->next = head->next;
        head->next = q;
        r = head->next;
        while (p&&r)
        {

            if (r->data == p->data&&p)
            {
                p = p->next;
                continue;
            }
            r = r->next;
        }
    }
}
int main()
{
    LNode *head;
    head = new LNode;
    int n;
    cin >> n;
    CreatList(head, n);
    Rev_same(head);
    PrintList(head);
    cout << endl;
    return 0;
}
```

正解：

```
#include<iostream>
using namespace std;
struct LNode
{
    int data;
    LNode *next;
};
void PrintList(LNode *head)
{
    LNode *p;
    p = head;
    p = p->next;
    while (p)
    {
        cout << p->data << " ";
        p = p->next;
    }
}
void CreatList(LNode *head, int n)
{
    LNode *p, *q;
    int x;
    q = head;
    for (int i = n; i > 0; i--)
    {
        p = new LNode;
        cin >> x;
        p->data = x;
        q->next = p;
        q = p;
    }
    q->next = NULL;
}
void Rev_same(LNode *head)
{
    LNode *p, *q, *r, *s;
    p = head->next;
    r = head;
    int flag = 0;
    while (p)
    {
        if (flag == 0)
        {
            q = p;
            p = p->next;
            q->next = NULL;
            r->next = q;
            r = q;
        }
        s = head->next;
        while (s && p)
        {
            if (s->data == p->data)
            {
                p = p->next;
                flag = 1;
                break;
            }
            flag = 0;
            s = s->next;
        }
    }
}
int main()
{
    LNode *head;
    head = new LNode;
    int n;
    cin >> n;
    CreatList(head,n);
    Rev_same(head);
    PrintList(head);
    cout << endl;
    return 0;
}
```

Problem-4

（本题目，我觉得将链表的数据域利用冒泡排序和将链表的数据域存储在数组然后冒泡排序效果是一样的，而为后者似乎更复杂，繁琐，但是前者却出现了TLE，why？？？）

```
#include<iostream>
using namespace std;
struct LNode
{
    int data;
    LNode *next;
};
void  fun(LNode *h)
{
    LNode *p;
    int *e;
    e = new int[10000];
    p = h->next;
    int i = 0;
    while (p)
    {
        p->data = e[i++];
        p = p->next;
    }
}
```

```
void CreatList(LNode *head)
{
    LNode *p;
    int x;
    head->next = NULL;
    while (1)
    {
        cin >> x;
        if (x == 0)
            break;
        p = new LNode;
        p->data = x;
        p->next = head->next;
        head->next = p;
    }
}
int Length(LNode *head)
{
    LNode *p;
    p = head->next;
    int i = 0;
    while (p)
    {
        p = p->next;
        i++;
    }
    return i;
}
void SortList(LNode *h)
{
    LNode *p;
    int *e;
    e = new int[10000];
    p = h->next;
    int i = 0;
    while (p)
    {
        e[i++] = p->data;
        p = p->next;
    }
    int t;
    for (int j = 0; j < i - 1; j++)
        for (int k = 0; k < i - 1 - j; k++)
            if (e[k] > e[k + 1] && e[k + 1] > 0)
            {
                t = e[k];
                e[k] = e[k + 1];
                e[k + 1] = t;
            }
    p = h->next;
    i = 0;
    while (p)
    {
        p->data = e[i++];
        p = p->next;
    }
    delete[]e;
}
void Cmb_List(LNode *ha, LNode *hb, LNode *hc)
{
    LNode *pa, *pb, *pc;
    pa = ha->next;
    pb = hb->next;
    pc = hc;
    while (pa  && pb)
    {
        if (pa->data < pb->data)
        {
            pc->next = pa;
            pc = pa;
            pa = pa->next;
        }
        else
        {
            pc->next = pb;
            pc = pb;
            pb = pb->next;
        }
    }
    pc->next = (pa) ? pa : pb;
}
void PrintList(LNode *head)
{
    LNode *p;
    p = head->next;
    while (p)
    {
        cout << p->data << " ";
        p = p->next;
    }
}
int main()
{
    LNode *ha, *hb, *hc;
    ha = new LNode;
    hb = new LNode;
    hc = new LNode;
    CreatList(ha);
    CreatList(hb);
    SortList(ha);
    SortList(hb);
    Cmb_List(ha, hb, hc);
    PrintList(hc);
    cout << endl;
    return 0;
}
```

Problem-5

(此题目中，要注意如何输出单向无结点循环链表元素:输出了其中的n-1个，剩下一个还要输出)

以下为源代码：

```
(p=head;
while(p->next!=head)
{
    cout<<p->data;
    p=p->next;
}
cout<<p->data;)
```

```
#include<iostream>
using namespace std;
struct LNode
{
    int data;
    LNode *next;
};
LNode* CreatList(int n)
{
    LNode *h;
    LNode *r, *p;
    h = new LNode;
    h = NULL;
    r = NULL;
    for (int i = 1; i <= n; i++)
    {
        p = new LNode;
        p->data = i;
        p->next = NULL;
        if (h == NULL)
        {
            h = p;
            r = h;
        }
        else
            r->next = p;
        r = p;
    }
    r->next = h;
    return h;
}
void Josephus(LNode *h, int m, int s)
{
    LNode *q, *p1;
    p1 = h;
    q = NULL;
    int i, j;
    for (i = 1; i < s; i++)
    {
        q = p1;
        p1 = p1->next;
    }
    while (p1->next != p1)
    {
        for (j = 1; j < m; j++)
        {
            q = p1;
            p1 = p1->next;
        }
        q->next = p1->next;
        cout << p1->data << " ";
        p1 = q->next;
    }
    cout << p1->data << endl;
```

```
}
int main()
{
    int n, s, m;
    LNode *h;
    cin >> n >> s >> m;
    h = CreatList(n);
    Josephus(h, m, s);
    return 0;
}
```