作业二

说明：除此文档外，另附有经过测试运行的 cpp 文件

1. 试编写算法，从顺序表中删除具有最小值的元素并由函数返回最小值，空出的位置由最后一个元素填补，若顺序表为空则显示出错信息并退出运行。

```
T DelMin()//1.删除最小值
    {
        if (size == 0)
        {
            cerr << "空表无最小值" << endl;
            exit;
        }
        T min = a[0];
        int n = 0;
        for (int i = 0; i < size; i++)
            if (a[i] < min)
                min = a[i], n = i;
        for (int i = 0; i < size; i++)
            if (a[i] == min)
                a[i] = a[size - 1];//空出的位置由最后一个元素填补
        return min;
    }
```

2. 试编写算法，从顺序表中删除具有给定值 *x* 的所有元素。

```
void DelX(T x)//2.删除x
    {
        bool b = 1;
        for (int i = 0; i < size; i++)
            if (a[i] == x)
            {
                for (int j = i; j < size - 1; j++)
                    a[j] = a[j + 1];
                i--;
                size--;
                b = 0;
            }
        if (b)
        {
            cerr << "表中无指定值" << endl;
            return;
        }
        }
```

3. 试编写算法，从有序表中删除其值在给定值 *s* 和 *t*(要求 *s* 小于 *t*)之间的所有元素。

```
void DelRange(T s, T t)//3.删除(s,t)中的元素
    {
        bool b = 1;
        if (s >= t)
        {
            cerr << "给定范围逻辑错误" << endl;
            return;
        }
        for (int i = 0; i < size; i++)
            if (s<a[i] && t>a[i])
            {
                for (int j = i; j < size - 1; j++)
                    a[j] = a[j + 1];
                i--;
                size--;
                b = 0;
            }
        if (b)
        {
            cerr << "表中无指定值" << endl;
            return;
        }
        }
```

4. 试编写算法，从顺序表中删除所有其值重复的元素，使所有元素的值均不同。如对于线性表(2，8，9，2，5，5，6，8，7，2)，则执行此算法后变为(2，8，9，5，6，7)。注意：表中元素未必是排好序的，且每个值的第一次出现应当保留。

```
void Unique()//4.删除重复
    {
        for (int i = 0; i < size; i++)
            for (int j = 0; j < i; j++)
                if (a[i] == a[j])
                {
                    for (int k = i; k < size - 1; k++)
                    {
                        a[k] = a[k + 1];
                    }
                    i--; size--;
                    break;
                }
    }
```

5.试编写算法，根据一个元素类型为整型的单链表生成两个单链表，使得第一个单链表中包含原单链表中所有元素值为奇数的结点，使得第二个单链表中包含原单链表中所有元素值为偶数的结点，原有单链表保持不变。

```
void Split(List<T>& even, List<T>& odd)
```

```
    {
        even.Clearnode(); odd.Clearnode();
        ListNode<T>* tmp = Head->next;
        while (tmp != nullptr)
        {
            if ((tmp->data) % 2)
                odd.add(tmp->data);
            else
                even.add(tmp->data);
            tmp = tmp->next;
        }
    }
```

6.设表 *L* 用数组表示，且各元素值递增有序。试写一算法，将元素 *x* 插入到表 *L* 的适当位置，使得表中元素仍保持递增有序。

```
void insertX(T x)//6.有序插入，不作排序
    {
        size++;
        int i = 0;
        int j = 0;
        for (i = 0; i < size - 1; i++)
            if (a[i] >= x)
            {
                for (j = size; j >= i; j--)
                    a[j] = a[j - 1];
                break;
            }
        a[j] = x;
    }
```

7. 已知一个单链表，设计一个复制单链表的算法。

```
void Copy(List<T>& one)
    {
        Clearnode();
        ListNode<T>* tmp = one.Head->next;
        while (tmp != nullptr)
        {
            add(tmp->data);
            tmp = tmp->next;
            size++;
        }
    }
```

8. 已知一个无序单链表，表中结点的 data 字段为正整数。设计一个算法按递增次序打印表中结点的值。

```
void Show_order()
    {
```

```
    List tmp(*this);
    tmp.Bsort();
    tmp.show();
}
void Bsort()//最简单的冒泡排序
{
    ListNode<T> *i, *j;

    for (i = Head->next; i!=nullptr; i=i->next)
        for (j = Head->next; j !=nullptr; j=j->next)
            if (i->data < j->data) {

                swap(i->data, j->data);
            }
    }
```