

```

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
// 线性顺序表的类定义
// Author: Melissa M.CAO
// Belong: Section of software theory, School of Computer Engineering & Science,
Shanghai University
// Version: 1.0
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

```

```
#pragma once
```

```

template <class datatype> class SeqSearchList;
template<class datatype> class SeqSortList;

```

//注意: 大小写, 尤其是"delete"和>Delete", delete 是系统默认函数, 而 Delete 是自定义函数

```

template <class datatype> class SeqList
{
    friend class SeqSearchList<datatype>;
    friend class SeqSortList<datatype>;

private:
    static const int maxlen = 40;
    // datatype* data;
    int len;
    int maxsize;
public:
    datatype* data; //----图的邻接表中用到, 修改太麻烦, 临时改为 public
    SeqList(int size = maxlen); //构造空表, 未申请空间
    SeqList(int size, int curlen, datatype *a, int t); //构造空表, 按当前长度申请空间, 每个存储空间按序赋值为 a
    ~SeqList() { if (data) delete[]data; }
    int Length() const { return len;}
    datatype Get (int i)const;
    int Locate( datatype &item ) const;
    int find(datatype &x)const;
    int Insert ( const datatype &item,int i );
    datatype Delete( const int i , bool f);
    datatype Delete(datatype &item );
    int IsEmpty()const;
    int IsFull()const;
    void Clear();
    void Display();

```

```

void Display(int low, int high);    //显示部分数据
int AppendItem( const datatype &item); //在顺序表的最后追加一个元素
datatype GetData(int i) {return data[i];}
void SetData (datatype d) { data[i] = d;}
void Initial() { len = 0;}
//////////扩展操作//////////
int Next(datatype &x);
int Prior(datatype &x);
int Union(SeqList<datatype> &B);      //与表 B 合并，且数据不重复
int InterSection(SeqList<datatype> &B); //与表 B 求交集
//////////书上习题//////////
datatype DelMinMaxElm(int select); //删除最大或最小元素，并用最后一个元
素代替
void DeleteALL(datatype &item, int select); //删除表中所有的 Item 值，
select 为算法选择
void DeleteAllRepeat();                //所有重复的元素只保留一
个
void DeleteBetweenST(datatype s, datatype t); //有序表中删除 s 和 t 之间
的元素
};

```