

1. 顺序表的结构定义
2. 顺序表的基本操作
 - 2.1 顺序表的初始化
 - 2.2 创建初始顺序表
 - 2.3 插入数据元素
 - 2.4 删除数据元素
 - 2.5 按元素值的查找
 - 2.6 求指定位置元素
 - 2.7 打印顺序表
3. 例题

1. 顺序表的结构定义

```
1  #include <iostream>
2  using namespace std;
3
4  #define MAX 100
5  typedef struct SqList{
6      int data[MAX]; // 存储表中元素的数组
7      int length;    // 顺序表长度
8  } SqList;
```

2. 顺序表的基本操作

2.1 顺序表的初始化

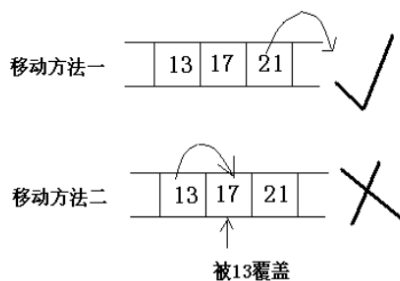
```
1  void init(SqList &L){
2      L.length = 0;
3  }
```

2.2 创建初始顺序表

```
1  void createSqList(SqList &sqList, int a[], int len){
2      sqList.length = 0;
3      for(int i=0; i<len; i++){
4          sqList.data[i]=a[i];
5          sqList.length++;
6      }
7  }
```

2.3 插入数据元素

找到插入位置之后，将插入位置及其以后的元素向后移动一个元素的位置即可。**先移动最右边的元素**，如果先移动最左边的元素则右边的元素会被左边的元素所覆盖。如下图所示：



```
1  int insert(SqList &L, int p, int e){
2      // 第p个位置插入元素
3      //位置错误或者表长已经达到
4      if(p<1 || p>L.length+1 || L.length==MAX){
5          return 0;
6      }
7      int i;
8      // 元素后移
9      for(i=L.length-1; i>=p; i--){
10         L.data[i+1] = L.data[i];
11     }
12     L.data[p-1] = e;
13     L.length++;
14 }
15
```

2.4 删除数据元素

删除表中下标为 p 的元素，只需将其后边的元素逐个往前移动一个位置，将 p 位置上的元素覆盖掉，就达到了删除的目的。

```
1  int delElem(SqList &L, int p, int &e){
2      if(p<1 || p>L.length){
3          return 0;
4      }
5      int i;
6      e = L.data[p-1];
7      // 元素前移
8      for(i=p-1; i<L.length-1; i++){
9          L.data[i] = L.data[i+1];
10     }
11     L.length--;
12     return 1;
13 }
```

2.5 按元素值的查找

```
1  int locateElem(SqList L, int e){
2      int i;
3      for(i=0; i<L.length; i++){
```

```

4         if(L.data[i] == e){
5             return i;
6         }
7     }
8     return 0;
9 }

```

2.6 求指定位置元素

```

1 int getElem(SqList L, int p, int &e){
2     if(p<1 || p>L.length){
3         return 0;
4     }
5     e = L.data[p];
6     return 1
7 }

```

2.7 打印顺序表

```

1 void printSqList(char * name, SqList sqList){
2     printf("顺序表 %s : [", name);
3     for(int i=0; i<sqList.length; i++){
4         printf(" %d ",sqList.data[i]);
5     }
6     printf("]\n");
7 }

```

3. 例题

- 已知一个顺序表 L，其中的元素递增有序排列，设计一个算法插入一个元素 x (x 为 int 型)后保持该顺序表仍然递增有序排列(假设插入操作总能成功，即插入后表长不会大于 MAX)。

```

1  /**
2   * 返回第一个比 x 大的元素的位置
3   *
4   * @param L
5   * @param x
6   * @return
7   */
8  int findElement1(SqList L, int x){
9      int p;
10     for(p=0; p<L.length; p++){
11         if(L.data[p] > x){
12             return p;
13         }
14     }
15     return p;
16 }
17
18 /**
19  * 将元素 x 插入表中

```

```
20  * @param L
21  * @param x
22  */
23 void insert1(SqList &L, int x){
24     int p;
25     p = findElement1(L, x);
26
27     // 其他元素后移
28     for(int i=L.length; i>=p; i--){
29         L.data[i+1] =L.data[i];
30     }
31
32     L.data[p] = x;
33     L.length++;
34 }
```