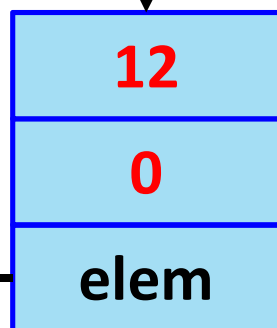
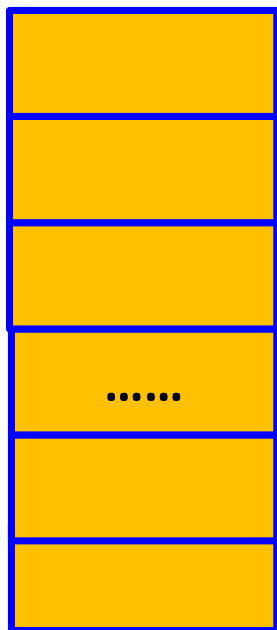


2.2 顺序表建立和判空

SeqList slist



2. 申请顺序表数组空间



1. 申请结构体List空间

slist->Max = 12

slist->n = 0

slist->elem

```
typedef int DataType;
struct List
{
    int Max;//最大元素个数
    int n;//实际元素个数
    DataType *elem;//首地址
};
typedef struct List *SeqList;
//顺序表类型定义
```

2.2 顺序表建立和判空

算法2-3

```
typedef int DataType;

struct List
{
    int Max;//最大元素个数
    int n;//实际元素个数
    DataType *elem;//首地址
};

typedef struct List *SeqList;
//顺序表类型定义

1 SeqList SetNullList_Seq(int m)//创建空顺序表
2 {
3     SeqList slist = (SeqList)malloc(sizeof(struct List));
4     //申请结构体List空间
5     if(slist != NULL){
6         slist->elem = (DataType*)malloc(sizeof(DataType)*m);
7         //申请顺序表空间，大小为m个DataType空间
8         if(slist->elem)
9         {
10             slist->Max = m;//顺序表的最大值
11             slist->n = 0; //顺序表长度赋值为0
12             return(slist);
13         }
14         else free(slist);
15     }
16     printf("out of space!!\n");
17     return NULL;
18 }
```

2.2 顺序表建立和判空 算法2-4

算法思路：通过检查顺序表的长度可以容易地判断。
顺序表长度为0则是空表，如果是空返回1，否则返回0。

```
1 int IsNullList_seq(SeqList slist)//判断顺序表是否为空
2 {
3     return(slist->n == 0);
4 }
```