5.2 数组的顺序表示和实现

类型特点:

1）只有引用型操作，没有加工型操作；

2）数组是多维的结构，而存储空间是一个一维的结构。

有两种顺序映象的方式:

1）以行序为主序（低下标优先）；

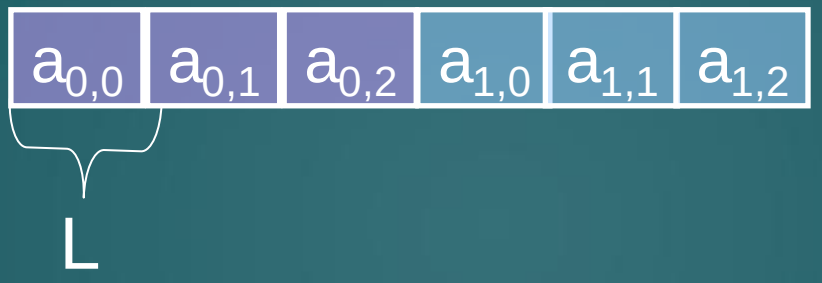
2）以列序为主序（高下标优先）。

例如：　　以“行序为主序”的存储映象

二维数组

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$

存储结构



二维数组A中任一元素$a_{i,j}$的存储位置

$$LOC(i,j) = LOC(0,0) + (b_2 \times i + j) \times L$$

称为**基地址**或基址

推广到一般情况，可得到 n 维数组数据元素存储位置的映象关系

$$LOC(j_1, j_2, \ldots, j_n) = LOC(0,0,\ldots,0) + \sum c_i j_i$$

其中 $c_n = L$，$c_{i-1} = b_i \times c_i$，$1 < i \leq n$。

称为 n 维数组的映象函数。数组元素的存储位置是其下标的线性函数。

```
// - - - - 数组的顺序存储表示 - - - -
#include < stdarg.h >
    //标准头文件，提供宏va_start、va_arg和va_end,用于存取变
    长参数表
#define MAX_ARRAY_DIM 8
   //假设数组维数的最大值为8
typedef struct{
     ElemType *base ;
     //数组元素地址，由InitArray分配
     int dim;
    //数组维数
     int *bounds;
     //数组维界基址，由InitArray分配
     int *constants;
     //数组映像函数常量基址，由InitArray分配
}Array;
```

```
// - - - - 基本操作的函数原型说明 - - - -
Status InitArray( Array &A, int dim... );
//若维数dim和随后的各维长度合法，则构造相应的数组A，并
    返回OK。
 int DestroyArray (Array &A);
//销毁数组A
 int StrCompare (Array &A, ElemType &e, ... )
//A是n维数组，e为元素变量，随后是n各下标值。
//若各下标不超界，则将e的值赋给所指定的A的元素，并返回
    OK。
```