

C语言程序设计

第九章 指针

第9章 指针(Pointer)

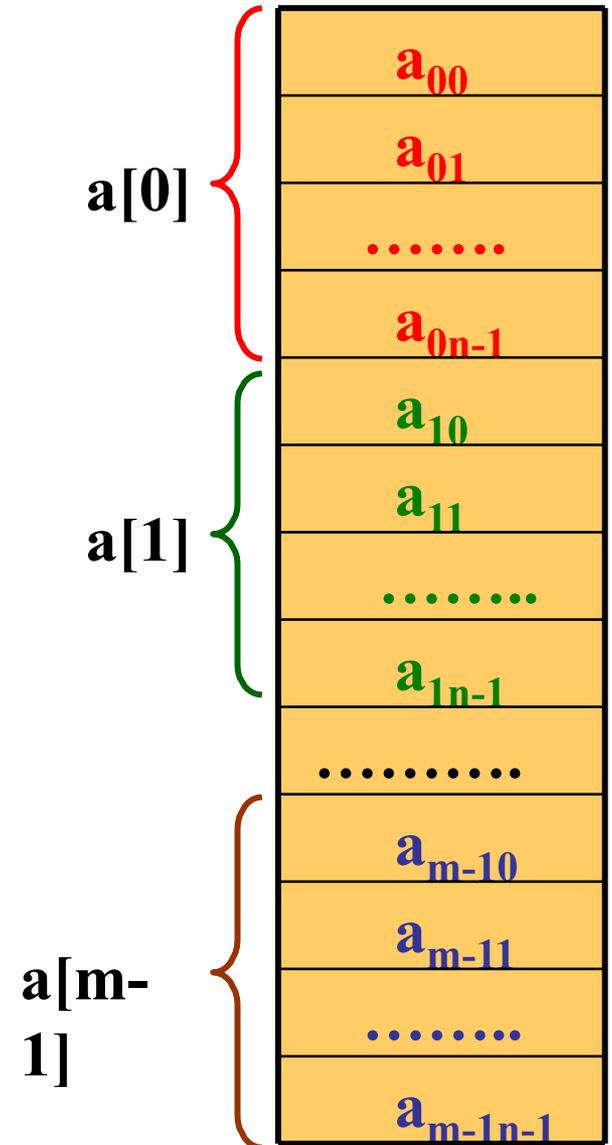
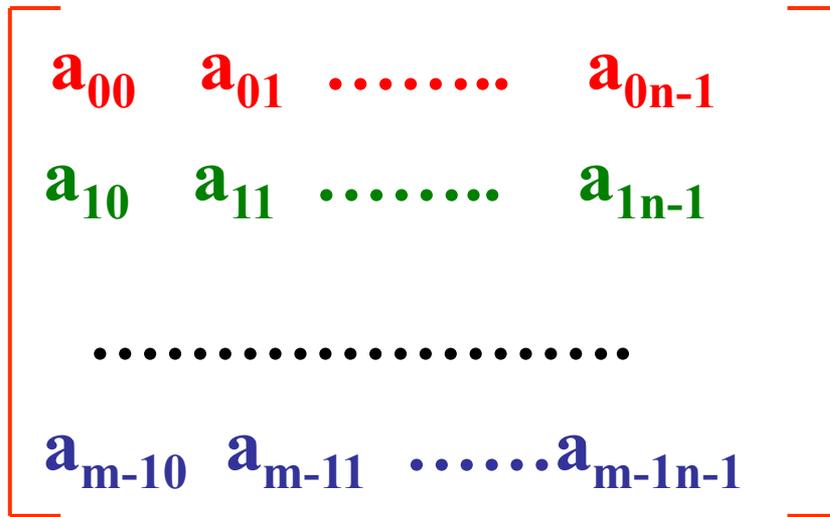
- 9.1 指针概述
- 9.2 指针与一维数组
- 9.3 指针与二维数组
- 9.4 指针与字符串
- 9.5 指针与函数
- 9.6 指针数组与多级指针

第九章 指针

➤ 9.3 指针与二维数组

C语言将二维数组看作一维数组，
其每个数组元素又是一个一维数组。
二维数组：按行的顺序存放。

$a[m][n]=$



第九章 指针

➤ 9.3 指针与二维数组

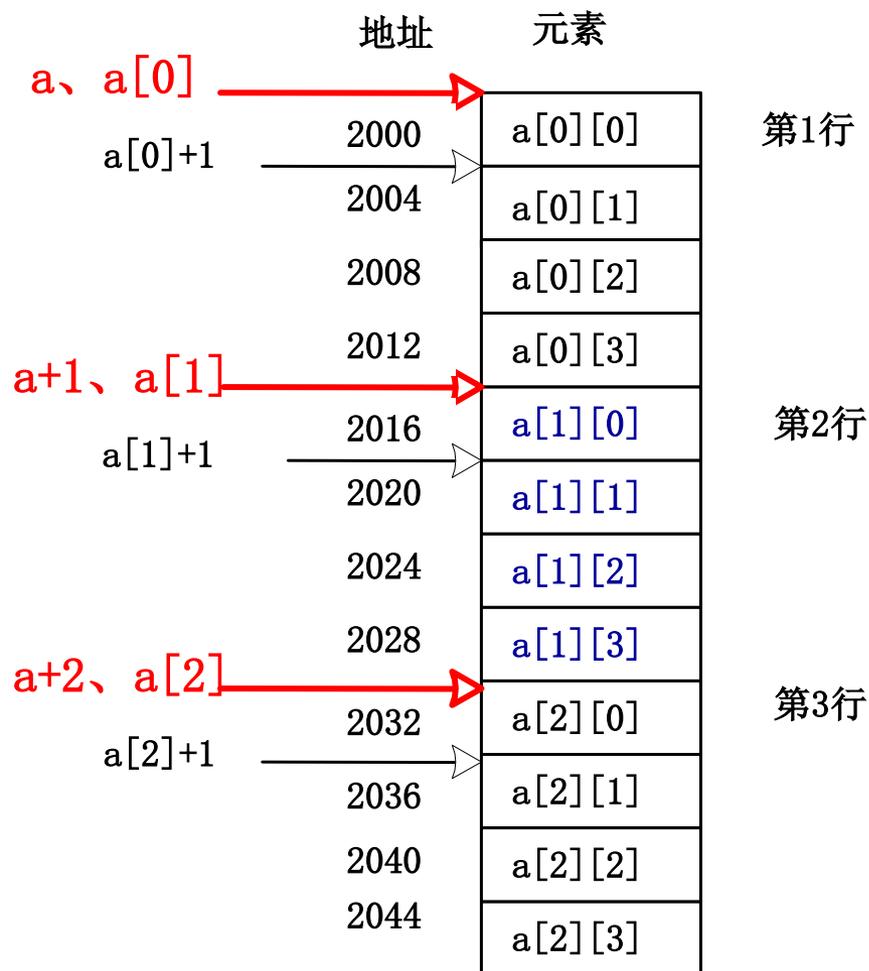
```
int a[3][4]={  
    {1,2,3,4},  
    {5,6,7,8},  
    {9,10,11,12}  
};
```

a 是一个数组名。

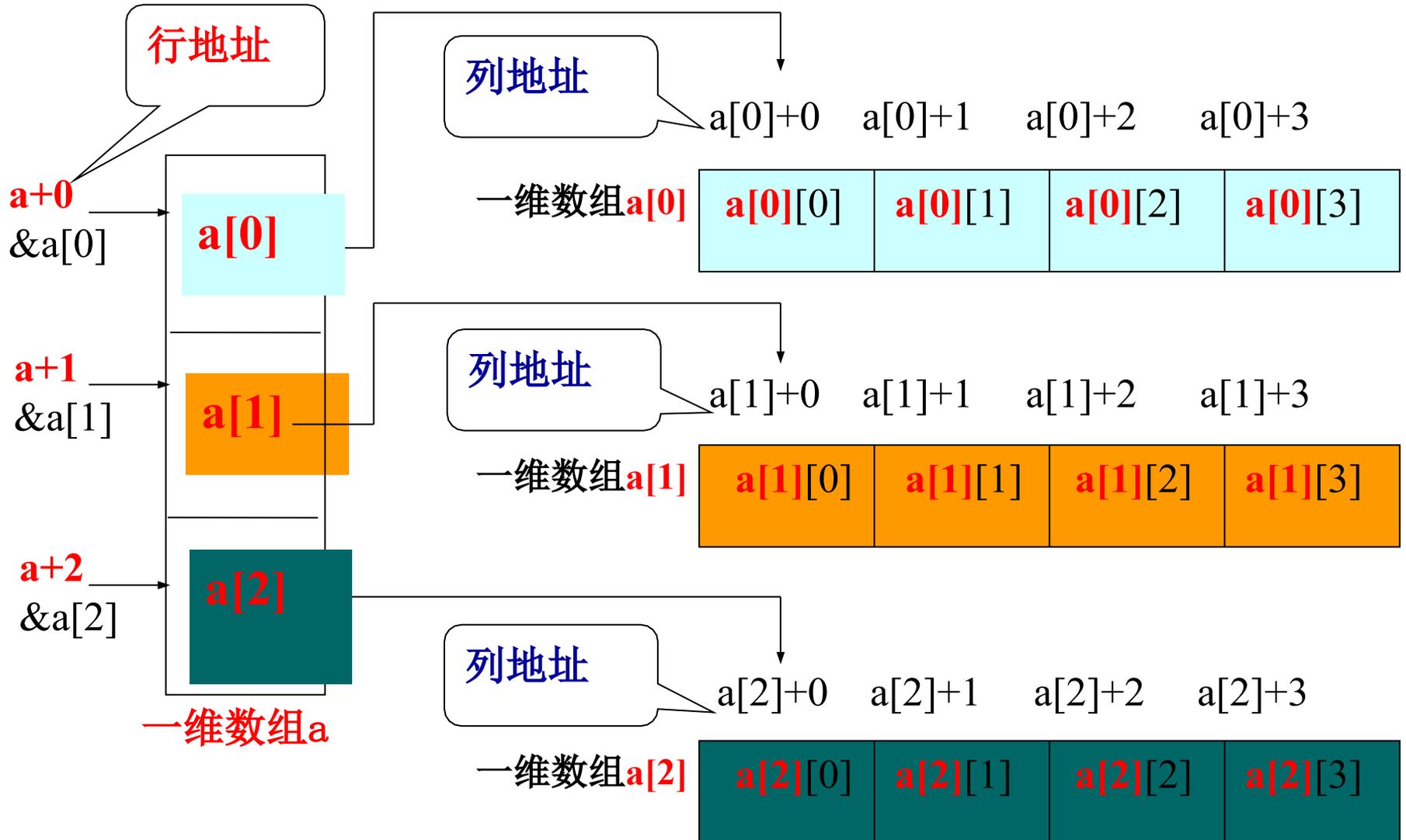
数组 **a** 包含3个元素：

a[0]、**a[1]**、**a[2]**。

每个元素又是一个一维数组，它包含4个元素。



第九章 指针



第九章 指针

二维数组元素 $a[i][j]$ 的定位: (指向数组元素的指针变量)

1. 先找到其所在的行

由于二维数组按行存放, 由二维数组的首地址 a , 可以确定第 i 行的首地址为 $a+i$ 。

第 0 行首地址为: a 或 $a+0$ 或 $a[0]$

第 i 行首地址为: $a+i$ \longleftrightarrow $a[i]$ \longleftrightarrow $*(a+i)$

2. 再找到其所在的列

行确定后, 只需进行行内再寻址, 就可确定列地址。

第 i 行第 j 列元素的地址为: $a[i] + j$ \longleftrightarrow $*(a+i)+j$

所以, 第 i 行第 j 列元素的值应为:

$*(*(a+i)+j)$ 或 $*(a[i]+j)$ 或 $a[i][j]$

第九章 指针

元素 $a[i][j]$ 的地址的几种等价的引用方式

- $\&a[i][j]$
- $a[i]+j$
- $*(a+i)+j$
- $\&(*a+i)[j]$

元素 $a[i][j]$ 的几种等价的引用方式

- $a[i][j]$
- $*(a[i]+j)$
- $*(*(a+i)+j)$
- $(*a+i)[j]$

第九章 指针

行指针变量:

行指针变量 也称为指向一维数组的指针变量，
一般定义形式为:

类型标识符 (*指针变量名) [常量表达式];

常量表达式: 表示一行含有多少个元素。在定义行指针变量时，必须确定每一行所含的元素个数。

```
int (*p)[4], (*q)[6], a[3][4], b[2][6];
```

```
p=a; /* p指向a的第1行的地址 */
```

```
q=b; /* q指向b的第1行的地址 */
```

p、q 虽然都是行指针变量，但类型不同。

p 指向的行只能有 4 个 int 型元素，

q 指向的行只能有 6 个 int 型元素。

第九章 指针

G:\ *D:\教学\C语言\2013—2014学年

```
1 2 3 14 5 6 7 8 9 10 11 12
 1  2  3 14
 5  6  7  8
 9 10 11 12
max=14,i=1,j=4
Press any key to continue_
```

```
for(j=0;j<4;j++)
    scanf("%d",&*(a+i+j));
for(i=0;i<3;i++)
{ for(j=0;j<4;j++)
    printf("%3d",&*(a+i+j));
  printf("\n"); }
maxp=*a;
```

【例9.6】输入一个 3×4 的矩阵到二维数组中，输出该矩阵，并且输出它的最大值元素及行列号。

```
for(i=0;i<3;i++)
    for(j=0;j<4;j++)
        if(*maxp<*(a+i+j))
            maxp=*(a+i+j);
i=(maxp-a)/4;
j=maxp-a;
printf("max=%d, i=%d, j=%d\n",*maxp, i+1, j+1);
}
```

第九章 指针

D:\教学\C语言\2013—2014学年

```
1 2 3 4
2 3 4 5
4 5 6 7
5 6 7 8
1 2
10 16
Press any key to continue_
```

```
for(i=0;i<4;i++)
    for(j=0;j<4;j++)
        scanf("%d", *(a+i)+j );
scanf("%d%d",&n,&m);
p=a+n-1; /* p 指向第n行 */
for(s1=0,j=0;j<4;j++)
    s1 +=>(*p+j);
```

【例9.7】输入一个4×4的矩阵到二维数组中，分别计算出二维数组中指定行，指定列的元素之和。

```
p=a;
for(s2=0,i=0;i<4;i++)
    s2 += (*(p+i)+m-1);
/* (*(p+i)+m-1)
   即 a[i][m-1] */
printf("%d %d\n",s1,s2);
}
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/177005000111010001>