

第10章 指针

本章主要内容：

1. 指针的基本概念

2. 各种类型的指针：

数组的指针、字符串的指针、
函数的指针、指针的指针等

指针的优点：

- (1) 可有效地表示复杂的数据结构；
- (2) 动态分配内存；
- (3) 方便地使用字符串，数组；
- (4) 直接处理内存地址等。

指针的缺点：

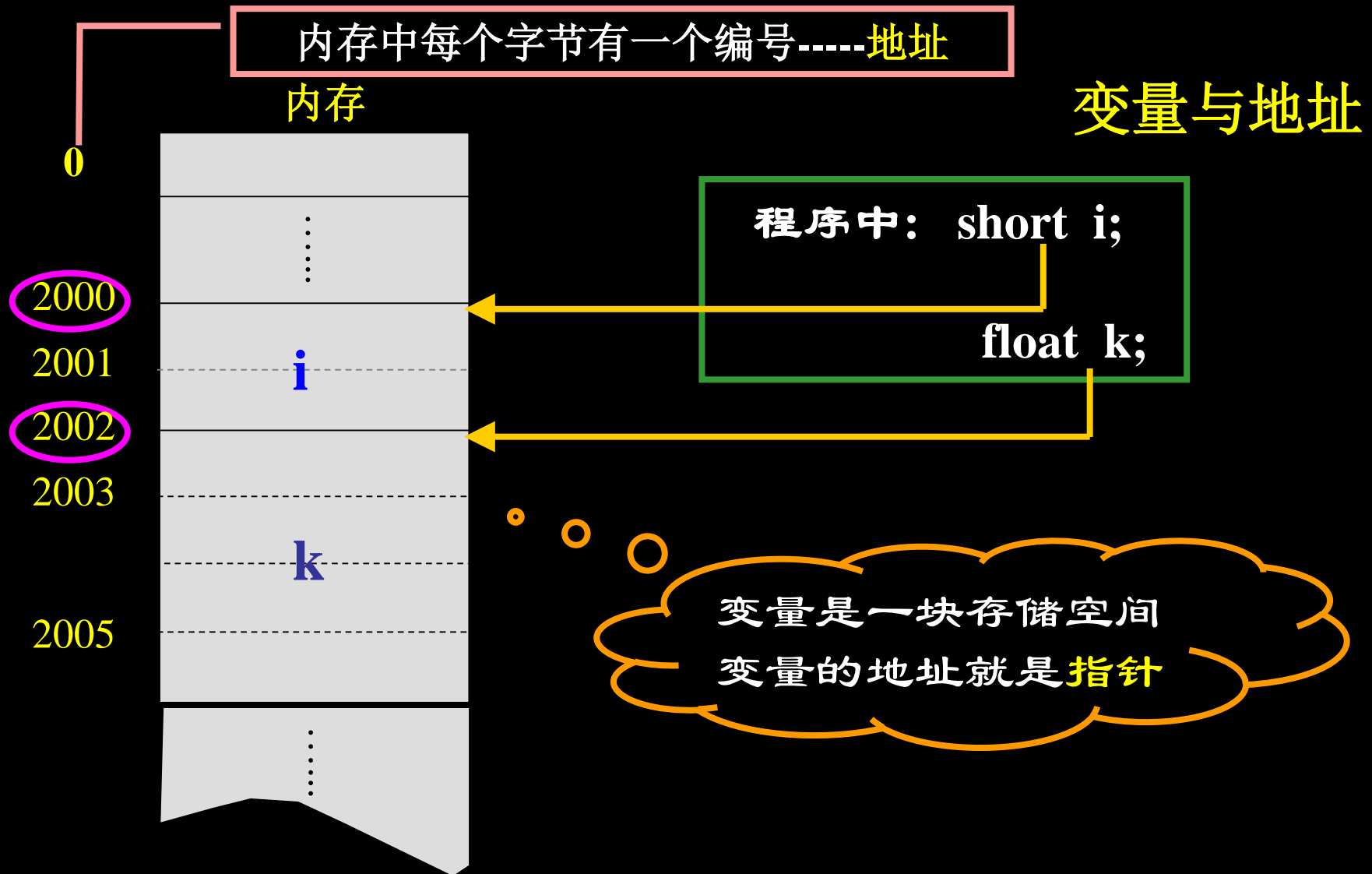
概念繁杂，使用时较灵活，不易掌握。

§ 10.1 地址和指针的概念

- 在计算机中，所有的数据都是存放在存储器中的。一般把存储器中的一个字节称为一个内存单元，不同的数据类型所占用的内存单元数不等，如整型量占2个单元，字符量占1个单元等。为了正确地访问这些内存单元，必须为每个内存单元编上号。根据一个内存单元的编号即可准确地找到该内存单元。内存单元的编号也叫做地址。既然根据内存单元的编号或地址就可以找到所需的内存单元，所以通常也把这个地址称为指针。

§ 10.1 地址和指针的概念

一、关于内存



- 内存单元的**指针**和内存单元的**内容**是**两个不同的概念**。对于一个内存单元来说，单元的地址即为指针，其中存放的数据才是该单元的内容。在C语言中，允许用一个变量来存放指针，这种变量称为指针变量。因此，一个指针变量的值就是某个内存单元的地址或称为某内存单元的指针。

二、关于变量的访问

1. 直接访问：根据变量名直接访问变量。

```
int i;
```

```
i=3;
```

```
printf("%d",i);
```

2. 间接访问：将变量的地址存放于另一变量

问题：

如何获得一个变量的地址？

如何保存一个变量的地址？

如何获得一个变量的地址？

& 取地址运算符。

```
int a;
```

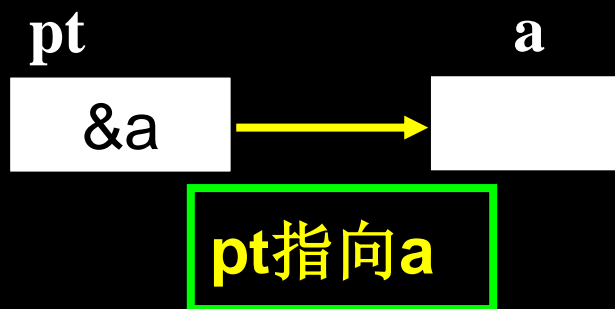
则&a获得变量的地址

如何保存一个变量的地址？

将变量的地址放在另一个（变量）中

三、若干术语

1. 指针：一个变量的地址，称为该变量的指针。
2. 指针变量：专门用来存放另一变量的地址（指针）的变量，称为指针变量。
3. 指向：指针变量指向变量。



§ 10.2 变量的指针和指向变量的指针变量

一、指针变量的定义

一般形式： 类型标识符 * 标识符；

例如： `float * pf;`
`int * p1,* p2;`

说明：

(1)定义时，标识符前面的“*”表示该变量为指针变量，上例中变量名为p1,p2。

(2)类型标识符指出p1、p2所指向的变量的类型。

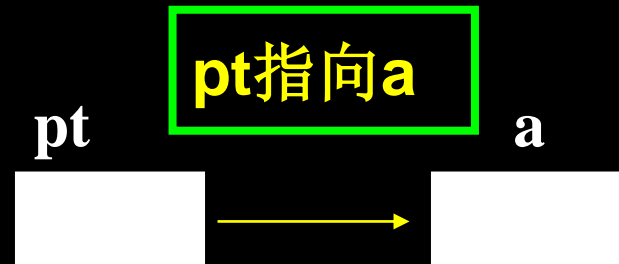
二、指针变量的引用

两个相关运算符:

1. & 取地址运算符。

例: `int a, *pt;`

若: `pt=&a`



2. * 指针运算符(间接访问运算符)

`*pt`为指针变量`pt`所指向的变量.

`*pt`等价于`a`

`*pt=10;` 相当于`a=10`

例:

main()

```
{ int a,b;
```

```
int *pointer_1,*pointer_2;
```

```
a=100;b=10;
```

```
pointer_1=&a;
```

```
pointer_2=&b;
```

```
printf(“%d,%d\n”,a,b);
```

```
printf(“%d,%d\n”,*pointer_1,*pointer_2);
```

```
}
```

pointer_1

&a



100

a

pointer_2

&b



10

b

例：输入两个整数**a**,**b**，按大小顺序输出。

main()

{

int a,b,*p1,*p2,*p;

scanf("%d%d",&a,&b);

p1=&a;p2=&b;

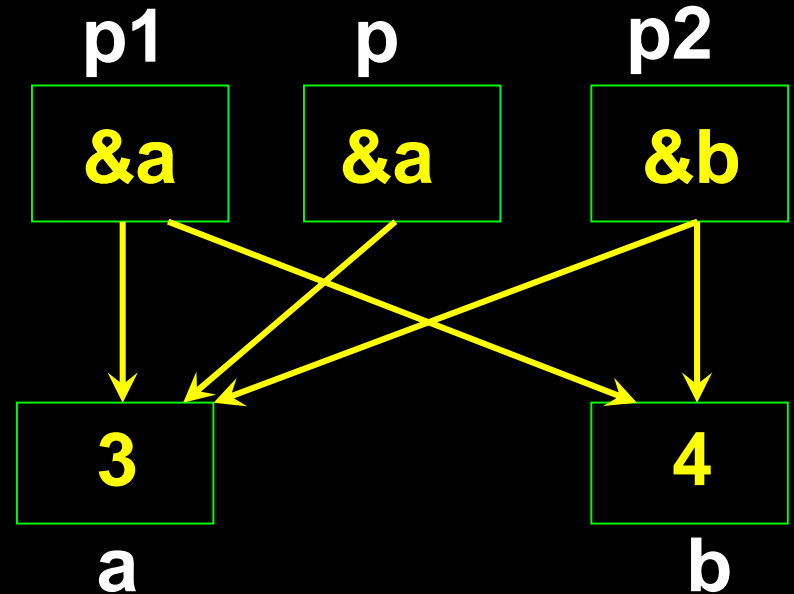
if(a<b)

{ p=p1;p1=p2;p2=p; }

printf("a=%d,b=%d",a,b);

printf("max=%d,min=%d",*p1,*p2);

}



注意:

1、指针变量必须先赋值, 再使用, 只能保存地址

例如: `int *p, *p1, a=10;`
`*p=20;`
`printf ("%d", *p1);`

2、一个指针变量只能指向同一类型的变量。

`int a,*p; float f;`
`p=&f;`

3、&和*运算符互为逆运算

若: `int a=10,*pt=&a;`

则: `&a`、`pt`、`&*pt`是等价的

则: `*&a`、`*pt`、`a` 是等价的

三、指针变量作为函数参数

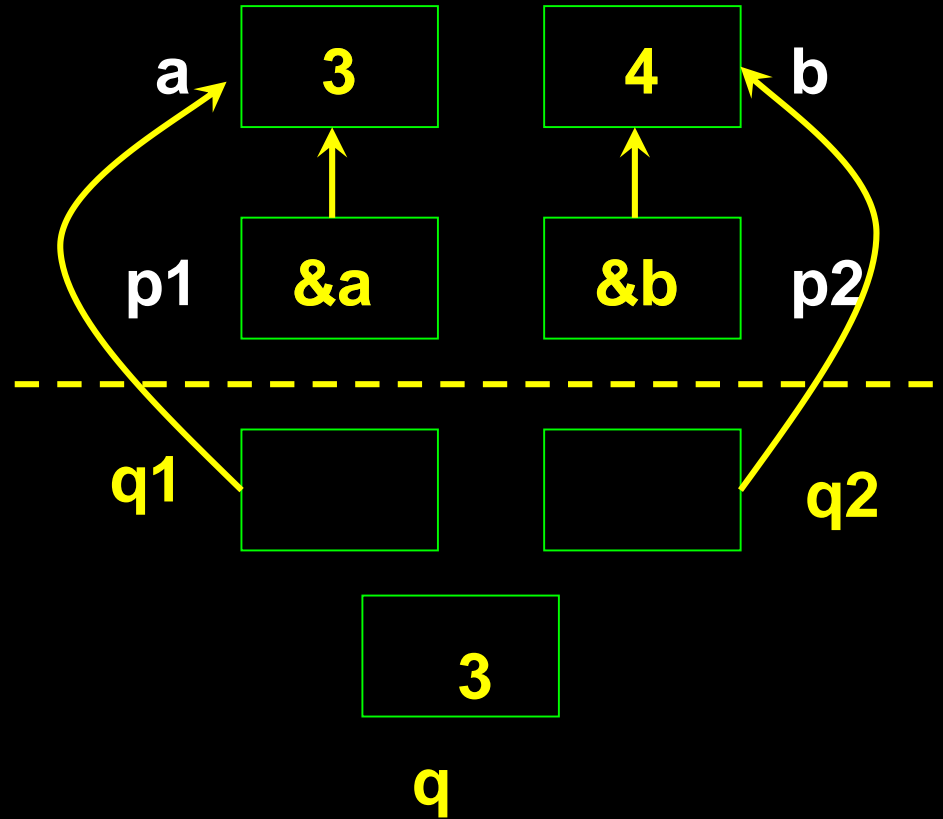
实参→形参：传去的是地址值，分占不同空间。

1、指针变量作形参，在函数内部可以改变函数外部变量的值

例：输入两个整数，从大到小输出。

```
void swap(int*,int*);
main()
{
    int a,b; int *p1,*p2;
    scanf("%d,%d",&a,&b);
    p1=&a;p2=&b;
    if(a<b) swap(p1,p2);
    printf("%d,%d",a,b);
}

void swap(int *q1,int *q2)
{
    int q;
    q=*q1; *q1=*q2; *q2=q;
}
```



2、指针变量作形参，可以用来保存函数的返回值 (输出型形参)

例：写一个函数，功能是求出一维整型数组中的最大值和最小值。


```
void maxmin(int a[], int n, int *pmax, int *pmin)
{
    int i;
    *pmax=*pmin=a[0];
    for(i=1;i<n;i++)
    {
        if(a[i]>*pmax) *pmax=a[i];
        if(a[i]<*pmin) *pmin=a[i];
    }
}

void main( )
{
    int a[10]={0,9,-5,4,23,87,34,2,65,-45};
    int max,min;
    maxmin(a,10, &max, &min);
    printf("max=%d,min=%d\n",max,min);
}
```

课上练习

看程序写结果:

```
void ast(int x,int y,int *cp,int *dp)
{ *cp=x+y; *dp=x-y; }
void main()
{
    int a,b,c,d;
    a=4;b=3;
    ast(a,b,&c,&d);
    printf(“%d,%d\n”,c,d);
}
```

答案: 7, 1

10.3 数组的指针和指向数组的指针变量

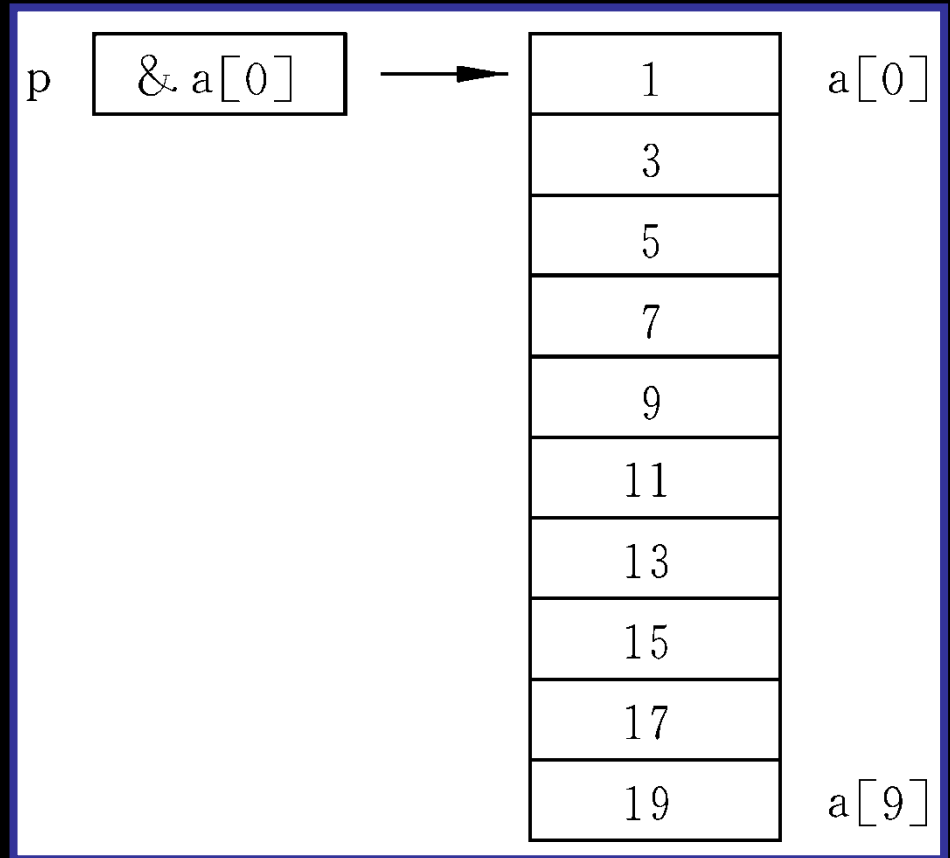
- 一个变量有地址，一个数组包含若干元素，每个数组元素都在内存中占用存储单元，它们都有相应的地址。指针变量既然可以指向变量，当然也可以指向数组元素（把某一元素的地址放到一个指针变量中）。
- 所谓数组元素的指针就是数组元素的地址

一、指向数组元素的指针

1.定义: `int a[10],*p;`

注意:如果数组为int型, 则指针变量的基类型亦应为int型。

2. 赋值: `p=&a[0];`
把 `a [0]`元素的地址
赋给指针变量 `p` 。
也就是使 `p` 指向 `a`
数组的第 0 号元素,
如图:



- 说明:

c语言规定数组名（不包括形参数组名），代表数组中首元素（下标为0的元素）的地址。

p=&a[0]; 则：等价于p=a;

- 注意:

- 数组名**a**不代表整个数组，上述“**p=a;**”的作用是把**a**数组的首元素的地址赋给指针变量**p**，而不是把数组**a**各元素的值赋给**p**;

- 在定义指针变量时可以对它赋予初值:

int *p=&a[0]; 或int *p=a;

等效于下面两行:

int *p;

p=&a[0]; 或 p=a; /*不是*p=&a[0]; */

习题集第七单元 一、8, 9, 10, 36

二、通过指针引用数组元素

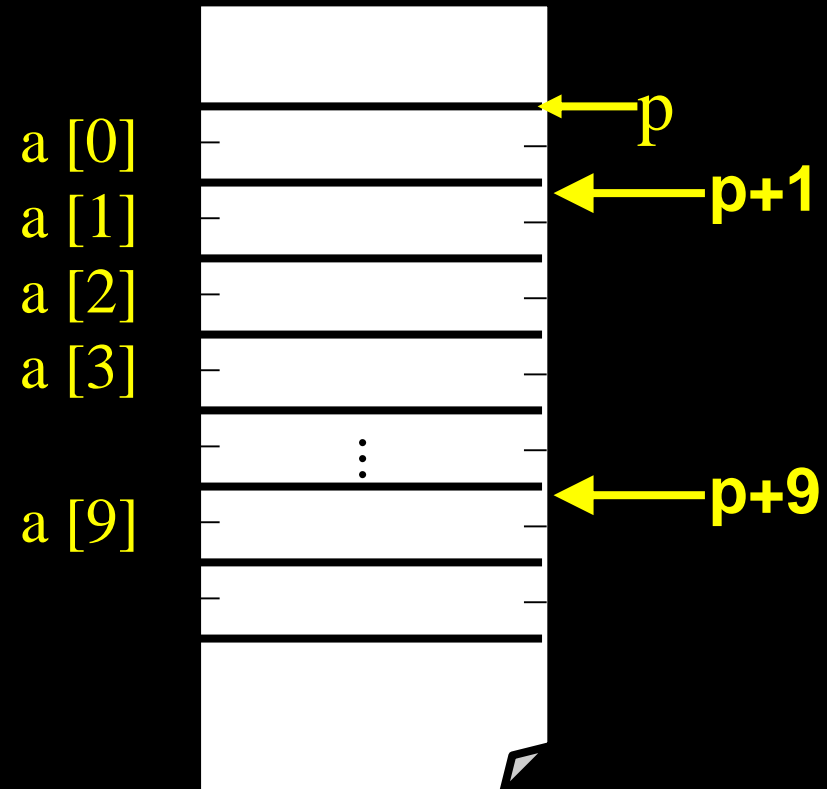
```
int a[10],*p;
```

若 $p=a$; 则 $*p=1$ 等价于 $a[0]=1$

$p+1$ 含义是什么?

C中规定: 若 $p=\&a[0]$,

则 $p+1$ 指向下一个元素, 不是简单的加1。



因此，如果

```
int a[10],*p;
```

```
p=a;
```

则有以下关系：

(1) $p+i \Leftrightarrow a+i \Leftrightarrow \&a[i]$ 。

(2) $*(p+i) \Leftrightarrow *(a+i) \Leftrightarrow a[i]$ 。

(3) 指向数组元素的指针变量也可带下标：

$p[i] \Leftrightarrow *(p+i)$ 。

即有 $*(p+i) \Leftrightarrow *(a+i) \Leftrightarrow a[i] \Leftrightarrow p[i]$

练习（习题集7单元二、3）

阅读程序，写结果

```
void main()
{
    int a[ ]={2,4,6,8,10};
    int y=1,x,*p;
    p=&a[1];
    for(x=0;x<3;x++)
        y+=*(p+x);
    printf("y=%d\n",y);
}
```

y=19

例：输出数组中全部元素。
(设a数组，整型，10个元素)

(1)下标法

```
void main ( )
```

```
{  int a[10];
```

```
    int i;
```

```
    for (i=0;i<10;i++)
```

```
        scanf("%d",&a[i]);
```

```
    printf("\n");
```

```
    for (i=0;i<10;i++)
```

```
        printf("%3d",a[i]);
```

```
}
```

(2)用数组名计算数组元素地址法

```
void main ( )
{   int a[10];
    int i;
    for (i=0;i<10;i++)
        scanf("%d",a+i);
    printf("\n");
    for (i=0;i<10;i++)
        printf("%3d",*(a+i));
}
```

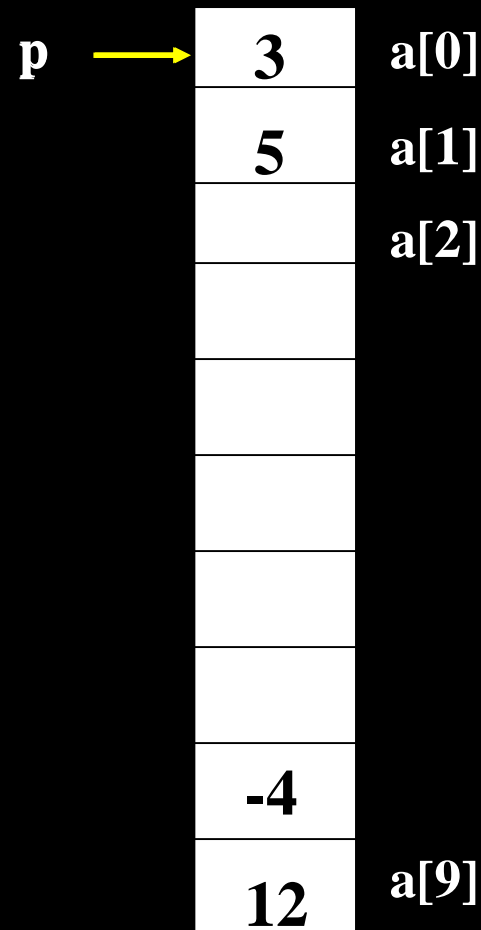
(3) 用指针变量法1:

```
main( )  
{ int *p,i,a[10];  
  p=&a[0]; // p=a;  
  for(i=0;i<10;i++)  
    scanf("%d", p+i);  
  printf("\n");  
  p=&a[0]; // p=a;  
  for (i=0;i<10;i++)  
    printf("%3d",*(p+i) );  
}
```

用指针变量法2

```
main( )
{  int  *p,i,a[10];
    p=a;
    for(i=0;i<10;i++)
        scanf("%d", p++);
    printf("\n");

    for (p=a;p<(a+10);p++)
        printf("%3d",*p );
}
```



在使用指针变量时应注意的几个问题：

1. 数组名是数组的首地址，因此 $p = \&a[0]$ 与 $p = a$ 等价。
2. 指针变量可以实现使本身的值改变，数组名不可以。

例： $p++$ 正确 $a++$ 不正确

3. 要注意指针变量的当前值，指针变量可以指到数组后的内存单元。即 指针变量的越界问题

4. 注意指针变量的运算，若先使 p 指向数组 a (即 $p = a$)，则
 - (1) $p++$ (或 $p += 1$) 指向下一个元素。
 - (2) $*p++$ 等价于 $*(p++)$ 先得到 p 指向的变量的值， p 再加1。
 - (3) $*(p++)$ 与 $*(++p)$ 作用不同，前者先取 $*p$ ，后使 p 加1；后者相反。
 - (4) $(*p)++$ 表示 p 所指向的元素值加1。
 - (5) 若 p 当前指向 a 数组第 i 个元素，则：
 - $*(p--)$ 相当于 $a[i--]$ ，先取 p 值作 $*$ 运算，再使 p 自减；
 - $*(++p)$ 相当于 $a[++i]$ ，先使 p 自加，再作 $*$ 运算；
 - $*(--p)$ 相当于 $a[--i]$ ，先使 p 自减，再作 $*$ 运算。

例:编程实现把整型数组**b**中的所有元素拷贝到数组**a**中。

```
void main()
{
    int a[10],b[10],i;
    int *pa,*pb;
    pa=a;pb=b;
    for(i=0;i<10;i++)
        scanf("%d",pb++);
    pb=b; /*必不可少，否则出现越界*/
    for(i=0;i<10;i++)
    { *pa=*pb;pa++;pb++;}
    pa=a; /*必不可少，否则出现越界*/
    for(i=0;i<10;i++)
        printf("%5d",*pa++);
}
```

习题集第七单元

一、1, 14, 15, 16,17,18,19,20



二、数组名作函数参数(P235)

内容回顾:

数组名做函数参数，当发生函数调用时实参传递给形参是什么？

```
int sum(int arr[],int n)
{
    int s=0,i;
    for(i=0;i<n;i++)
        s+=a[i];
    return s;
}
```

```
void main()
{
    int array[5]={1,2,3,4,5},i,s;
    s=sum(array,5);
    printf("sum=%d\n",s);
}
```

实参传递给形参是实参数组的首地址

实际上，形参数组名就是一个指针变量

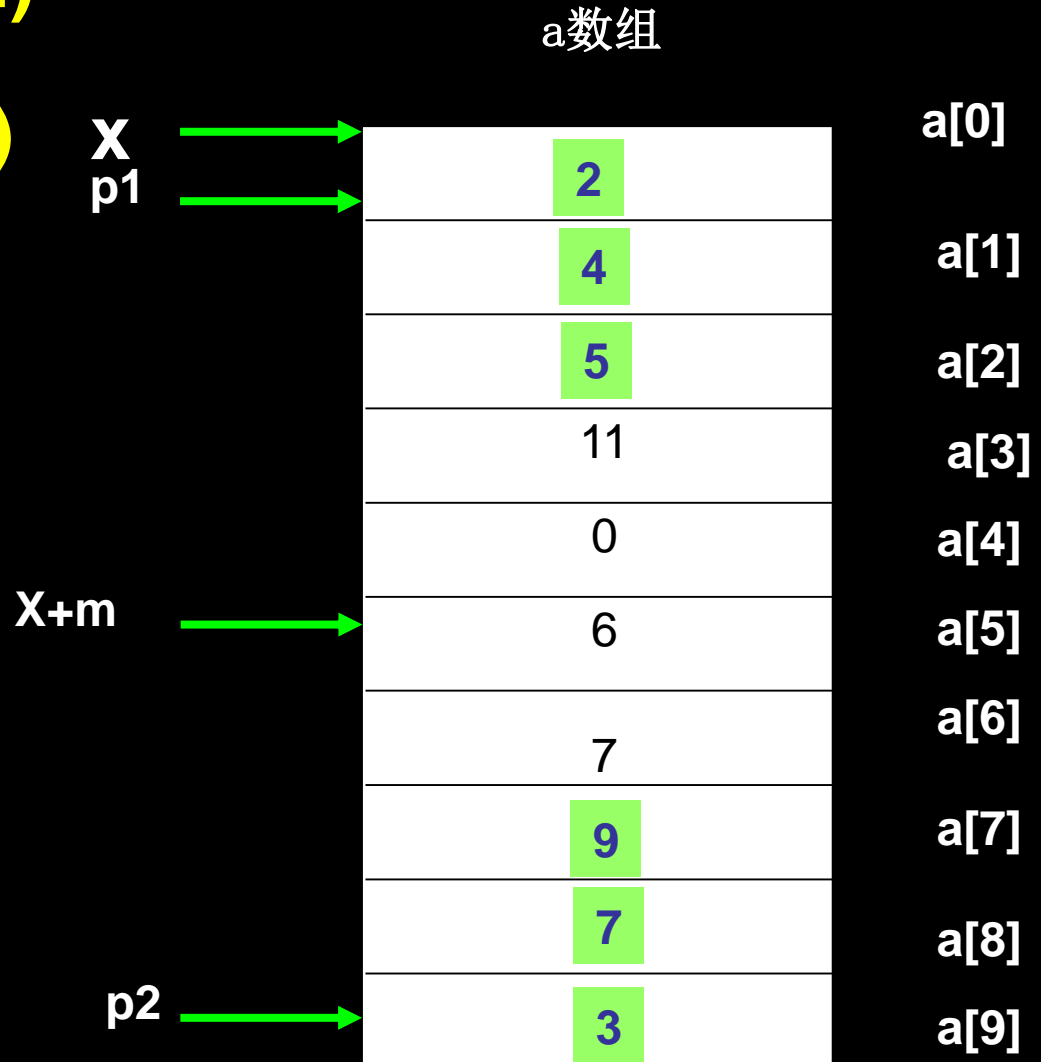
int sum(int arr[],int n)等价于

int sum(int * arr,int n)

例10.7：将数组a中n个整数按相反顺序存放。

void inv(int x[], int n)

void inv(int *x, int n)



```
void inv(int *x, int n)
{  int temp,*p1,*p2,m;
   m=n/2;
   p1=x;  p2=x+n-1;
   for ( ;p1<x+m;p1++,p2--)
   {   temp=*p1;
       *p1=*p2;
       *p2=temp;
   }
}
```

```
void main()
{
   int a[5]={1,2,3,4,5};
   int i;
   inv(a,5);
   for(i=0;i<5;i++)
       printf("%4d",a[i]);
}
```

如果有一个实参数组，想在函数中改变此数组，实参与形参的对应关系有以下几类情况

形参	实参
数组名	数组名
数组名	指针变量
指针变量	数组名
指针变量	指针变量

(1) 形参用数组名，实参用指针变量

```
void sort(int x[], int n)
```

```
{  int i,j,k,t;
    for (i=0;i<n-1;i++)
    {  k=i;
        for (j=i+1;j<n;j++)
            if (x[j]>x[k]) k=j;
        if (k!=i)
        {  t=x[i]; x[i]=x[k];
            x[k]=t; }
    }
}
```

```
main ( )
```

```
{  int *p,i,a[10];
    p=a;
    for (i=0;i<10;i++)
        scanf("%d",p++);
    p=a;
    sort(p,10);
    for (p=a,i=0;i<10;i++)
        printf("%d",*p++);
}
```

sort(a,10);也行

(2) 形参用指针变量, 实参用数组名

```
void sort(int *x, int n)
{ int i,j,k,t;
  for (i=0;i<n-1;i++)
  { k=i;
    for (j=i+1;j<n;j++)
      if (*(x+j)>*(x+k)) k=j;
    if (k!=i)
    { t=*(x+i);
      *(x+i)=*(x+k);
      *(x+k)=t;
    }
  }
}
```

```
main ( )
{ int i,a[10];
  for (i=0;i<10;i++)
    scanf("%d",&a[i]);
  sort(a,10);
  for (i=0;i<10;i++)
    printf("%3d",a[i]);
}
```

- 指针变量与一维数组的关系

`int *p` 与 `int q[10]`

- 数组名是**指针**（地址）**常量**

- `p=q`; `p+i` 是`q[i]`的地址

- 数组元素的表示方法:**下标法**和**指针法**,

即若`p=q`, 则 $p[i] \Leftrightarrow q[i] \Leftrightarrow *(p+i) \Leftrightarrow *(q+i)$

- **形参数组**实质上是指针变量,

即 $\text{int } q[] \Leftrightarrow \text{int } *q$

- 系统只给`p`分配能保存一个指针值的内存区(一般4字节) ; 而给`q`分配4*10字节的内存区

三、二维数组和指针(P242)

1、使用指针访问二维数组方法一：指向数组元素的指针变量。

书上246~247页

例：用指针变量输出二维数组元素的值。

```
main( )
```

```
{
```

```
    int a[2][3]={2,3,6,7,8,5};
```

```
    int *p;
```

```
    p=&a[0][0];
```

```
    for(i=0;i<6;i++)
```

```
        printf("%3d",*p++);
```

```
}
```

p →

2

3

6

7

8

5

a[0][0]

a[0][1]

a[0][2]

a[1][0]

a[1][1]

a[1][2]

例：用指针变量输出数组元素a[i][j]的值。

```
main()  
{  int a[2][3]={2,3,6,7,8,5};  
    int i,j,*p;  
    p=&a[0][0];  
    scanf("%d %d",&i,&j);  
    printf("%3d", *(p+i*3+j) );  
}
```

其中3是二维数组的
列数

如果有如下定义:

```
int a[4][5];
```

```
int * p; p=&a[0][0];
```

问题:

p+11是哪个数据元素的地址。

答案: &a[2][1]

总结: $p+i$ 是 $\&a[i/5][i\%5]$

其中5是二维数组的列数

习题集第七单元

二、4

例： 将一个二维数组a[4][5]进行按行全排序

```
main()
```

```
{  int a[4][5]={1,4,2,3,7,6,9,10,8,-1,  
                -2,-6,-4,4,2,7,6,9,11,14};
```

```
    int i,j;
```

```
    for(i=0;i<4;i++)
```

```
    {  for(j=0;j<5;j++)
```

```
        printf("%4d",a[i][j]);
```

```
        printf("\n");
```

```
    }
```

```
    sort(&a[0][0],4*5);
```

```
    for(i=0;i<4;i++)
```

```
    {  for(j=0;j<5;j++)
```

```
        printf("%4d",a[i][j]);
```

```
        printf("\n");
```

```
    }
```

```
}
```

sort(&a[0][0],4*5);

void sort(int *p,int n)

{ int i,j,t,k;

for(i=0;i<n-1;i++)

{ k=i;

for(j=i+1;j<=n-1;j++)

if(*(p+k)<*(p+j))

k=j;

if (k!=i)

{ t=*(p+i); *(p+i)=*(p+k);

***(p+k)=t; }**

}

p →

	a[0][0]
	a[0][1]
	<u>a[0][2]</u>
	a[0][3]
	a[0][4]
	a[1][0]
	a[1][1]
	a[1][2]
	a[1][3]
	a[1][4]
.	.
.	.
.	.
	a[3][2]
	a[3][3]
	a[3][4]

2、使用指针访问二维数组方法二： 指向由m个整数组成的一维数组的指针变量。

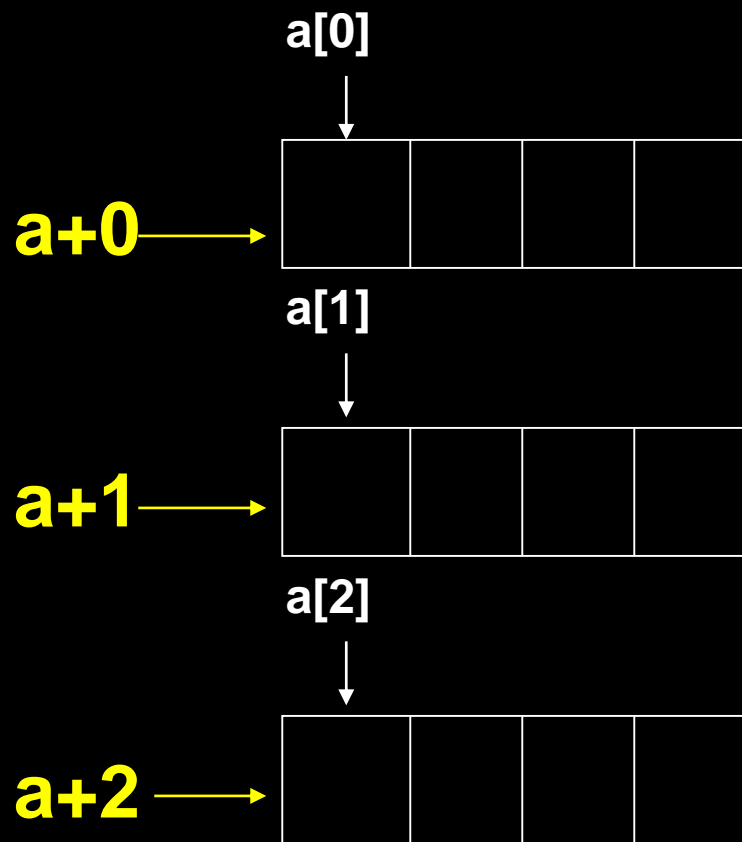
(1) 二维数组的地址 P242

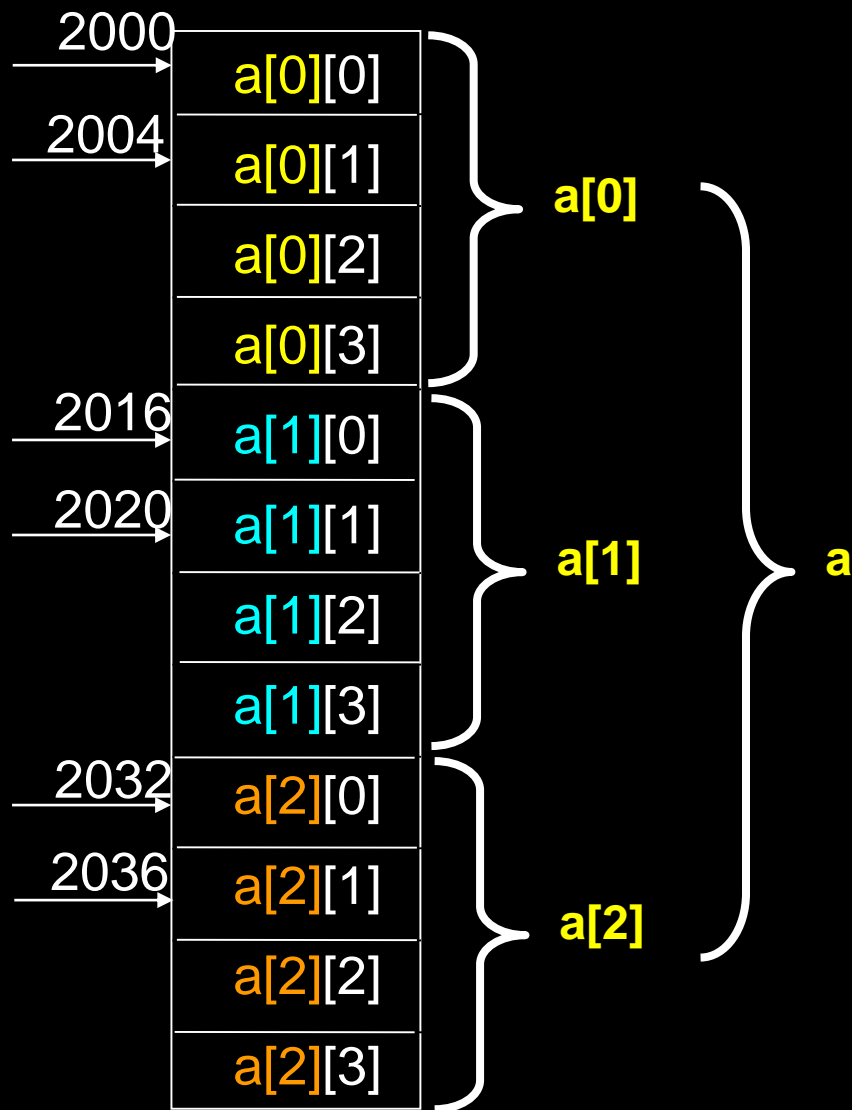
```
int  a[3][4]={ {6,9,0,16},  
               {8,-1,2,7},  
               {3,48,-5,12}};
```

a代表的是首行(即第0行)的首地址

a+1代表的是第1行的首地址

a+2代表的是第2行的首地址





数组a包含三个元素
`a[0]`,`a[1]`,`a[2]`

数组名a是首元素的地址
a就是`&a[0]`;

`a[0]`,`a[1]`,`a[2]`是一维数组名, C语言又规定了数组名代表数组首元素地址, 因此,
`a[0]`代表一维数组`a[0]`中第0列元素的地址, 即`&a[0][0]`
`a[1]`代表一维数组`a[1]`中第0列元素的地址, 即`&a[1][0]`
`a[2]`代表一维数组`a[2]`中第0列元素的地址, 即`&a[2][0]`

1) $a+i \Leftrightarrow \&a[i]$ 第i行的地址

$*(a+i) \Leftrightarrow a[i]$ 是 $a[i]$ 元素

2) $a[i]$ 元素本身又是一个一维数组, $a[i]$ 相当于一维数组名, 即 $a[i]$ 中保存一个地址, 是 $a[i][0]$ 的地址, 即 $\&a[i][0]$

$a[i] \Leftrightarrow \&a[i][0]$

3) $a[i]+j \Leftrightarrow \&a[i][j]$ $a[i][j]$ 的地址

即: $*(a+i)+j \Leftrightarrow \&a[i][j]$

4) $a[i][j]$ 元素可以表示为:

$*(a[i]+j) \Leftrightarrow *((a+i)+j) \Leftrightarrow a[i][j]$

(2) 指向由m个整数组成的二维数组的指针变量

填空:

`int a[3][4];`

`p=a;`

p是a[0]的地址
是二维数组a中第
0行的地址

答案:

`int * p;` ✗

正确答案:

`int (*p)[4];`

p称为指向一维数组的指针
是指向由4个整数组成的二维
数组的指针。

例：输出二维数组任一行任一列的元素的值。

```
main ()  
{  int a[3][4]={1,3,5,7,9,11,13,15,17,19,21,23};  
    int (*p)[4];  
    int i,j;  
    p=a;  
    scanf("i=%d,j=%d",&i,&j);  
    printf("a[%d,%d]=%d\n",i,j,*(*p+i)+j);  
}
```

(3) 多维数组的指针作函数参数。

例： 求二维数组a[4][5]每列最大值

a+0 →	2	12	3	27	-4
a+1 →	3	7	33	2	5
a+2 →	1	6	12	3	15
a+3 →	4	8	10	9	14
b →	4	12	33	27	15

```
void colum_max(int (*a)[5], int * b)
{
    int i,j;

    for(j=0;j<5;j++)
    {
        *(b+j)=*(*a+0)+j; /*a[0][j]*/
        for(i=1;i<4;i++)
            if (*b+j)<*(*a+i)+j)
                *(b+j)=*(*a+i)+j);
    }
}
```

```
main( )
{   int a[4][5],b[5];
    int i,j;
    for(i=0;i<4;i++)
        for(j=0;j<5;j++)
            scanf("%d",&a[i][j]);
    colum_max(a,b);
    for(i=0;i<4;i++)
    {   for(j=0;j<5;j++)
        printf("%4d",a[i][j]);
        printf("\n");
    }
    for(j=0;j<5;j++)
        printf("%4d",b[j]);
    printf("\n");
}
```

打印原二维数组

打印每列最大值

习题集第七单元

一、2, 23

二、4, 19, 24, 25, 27