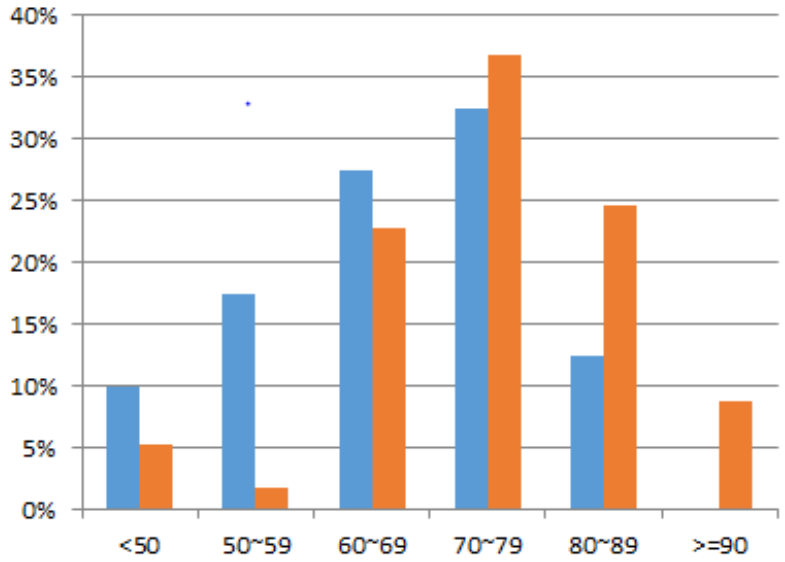
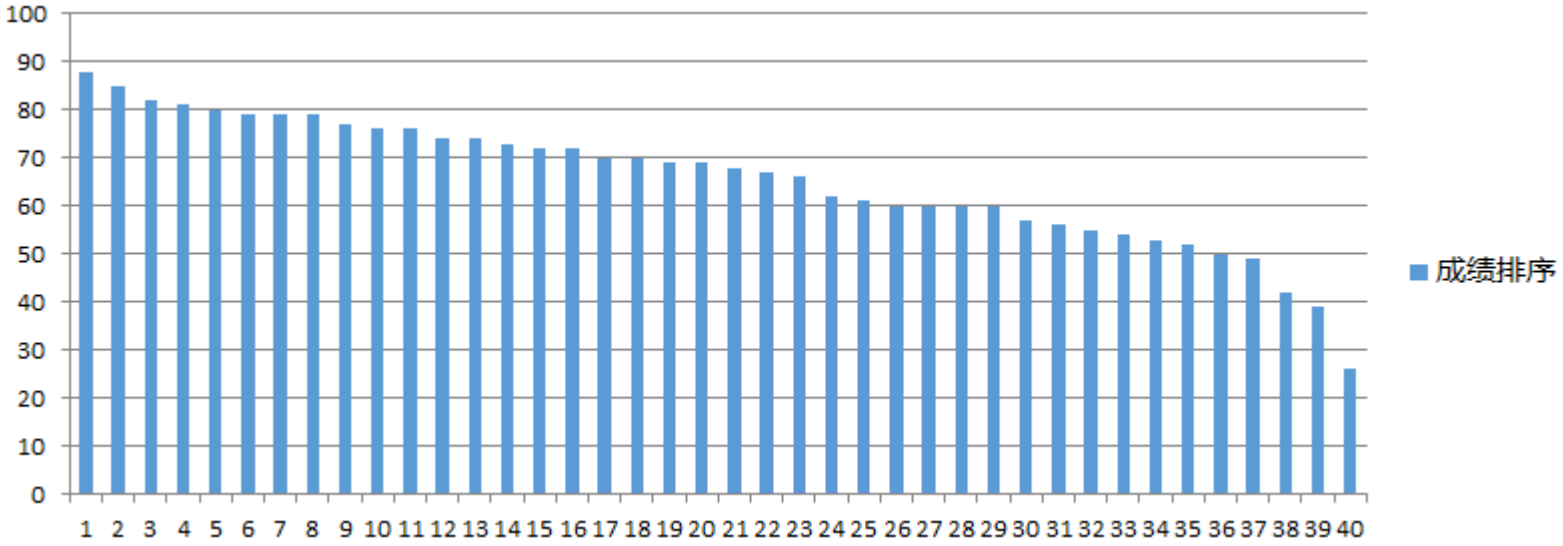
第8章 指针（2）

S.IST@XMU

半期考成绩分布



	人数
<50	4
50~59	7
60~69	11
70~79	13
80~89	5
>=90	0
平均分	65.6



复习回顾

- 上次课的内容：
  - 内存管理
  - 指针的基本概念
  - 如何定义指针变量
  - 如何引用指针变量
  - 指针运算符与应用
  - 指针变量作函数参数

单身人士就是一枚可怜的野指针
感情没有目标地址

至于已婚人士
情人就像指针
用的时候一定要注意
不然会带来巨大的灾难

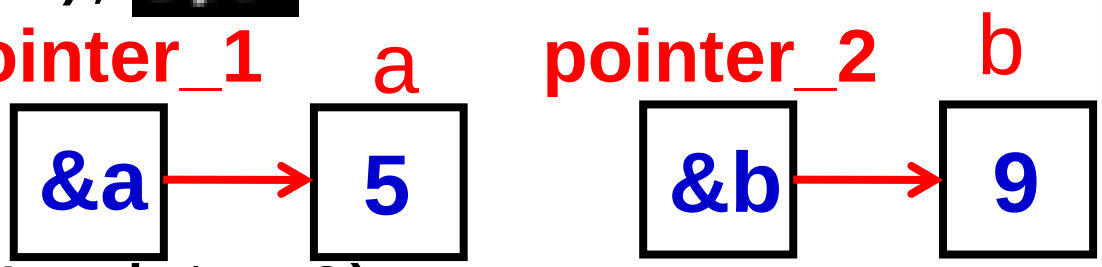
指针变量作函数参数的例子

- 对输入的两个整数按大小顺序输出。现用函数处理，而且用指针类型的数据作函数参数
- **解题思路：**定义一个函数swap，将指向两个整型变量的指针变量作为实参传递给swap函数的形参指针变量，在函数中通过指针实现交换两个变量的值。

指针变量作函数参数的例子

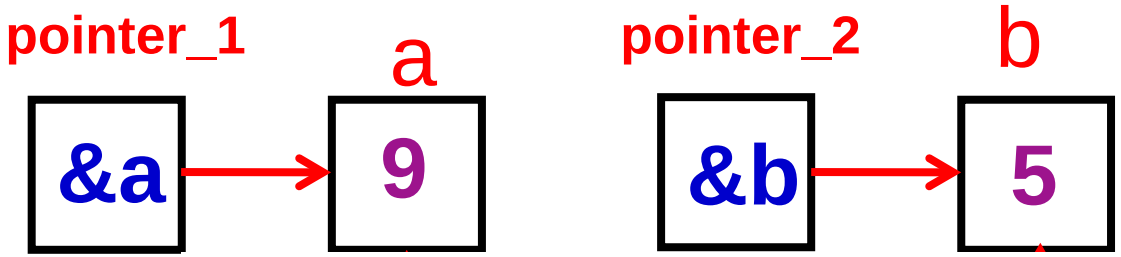
```
1. #include <stdio.h>
2. int main()
3. {
4.     void swap(int *p1,int *p2);
5.     int a,b;  int *pointer_1,*pointer_2;
6.     printf("please enter a and b:");
7.     scanf("%d,%d",&a,&b);
8.     pointer_1 = &a;
9.     pointer_2 = &b;
10.    if (a<b)
11.        swap(pointer_1,pointer_2);
12.    printf("max=%d,min=%d\n",a,b);
13.    return 0;
14.}
```

5,9



指针变量作函数参数的例子

```
void swap(int *p1,int *p2)
{//调用方式：swap(pointer_1,pointer_2);
    int temp;
    temp = *p1;
    *p1 = *p2;
    *p2 = temp;
}
```




```
please enter a and b:5,9
max=9,min=5
```

swap的失败例子之一

```
void swap(int *p1,int *p2)
{//调用方式：swap(pointer_1,pointer_2);
    int temp;
    temp = *p1;
    *p1 = *p2;
    *p2 = temp;
}
```

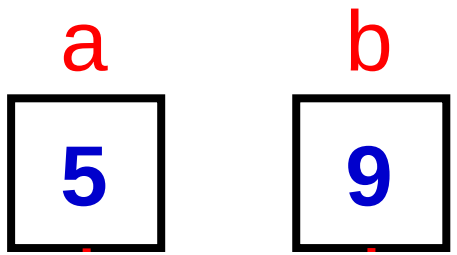
错！！！
无确定的指向

```
void swap(int *p1,int *p2)
{
    int *temp;
    *temp = *p1;
    *p1 = *p2;
    *p2 = *temp;
}
```

swap的失败例子之二

```
#include <stdio.h>
int main()
{   ……
    if (a<b)
        swap(a,b);
    printf("max=%d,min=%d\n",a,b);
    return 0;
}
void swap(int x,int y)
{
    int temp;

    temp = x;
    x = y;
    y = temp;
}
```



错！！！
无法交换a,b

swap的失败例子之三

```
#include <stdio.h>
int main()
{   ……
    scanf("%d,%d",&a,&b);
    pointer_1=&a;  pointer_2=&b;
    if (a<b)
        swap(pointer_1, pointer_2);
    printf("max=%d,min=%d\n",a,b);
    return 0;
}
void swap(int *p1,int *p2)
{
    int *p;
    p = p1;
    p1 = p2;
    p2 = p;
}
```

```
5,9
```

```
max=5,min=9
```

错！！！
只交换形参指向

swap例子启示录

- 函数的调用可以（而且只可以）得到一个返回值（即函数值），而**使用指针变量作参数，可以得到多个变化了的值**。如果不用指针变量是难以做到这一点的。
- 要从函数调用获得多个返回值，可以利用**指针参数传递**法。

指针参数传递法

- 如果想通过函数调用得到 n 个要改变的值：
  - ① 在主调函数中设 n 个变量，用 n 个指针变量指向它们
  - ② 设计一个函数，有n个指针形参。
  - ③ 在主调函数中调用这个函数，在调用时将这n个指针变量作实参，或直接将它们的地址传给该函数的形参
  - ④ 在执行该函数的过程中，通过形参指针变量，改变它们所指向的 n 个变量的值
  - ⑤主调函数中就可以使用这些改变了值的变量

指针参数传递实例

```
1. #include <stdio.h>
2. void fun(int *p, int *q)
3. {
4.     printf("*p=%d, *q=%d\n", *p, *q);
5.     (*p)++;
6.     (*q)--;
7.     printf("*p=%d, *q=%d\n", *p, *q);
8. }
9. int main()
10.{
11.    int x=0, y=0;
12.    printf("x=%d, y=%d\n", x, y);
13.    fun(&x, &y);
14.    printf("x=%d, y=%d\n", x, y);
15.    return 0;
16.}
```

指针作参数

改变指针指向的变量的值

```
x=0, y=0
*p=0, *q=0
*p=1, *q=-1
x=1, y=-1
```

指针与数组

- 指针与数组是“天然的好朋友”。
  - 每一个变量都有地址，一个数组包含若干元素，每个数组元素也都有相应的地址
  - 指针变量可以指向数组元素（即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

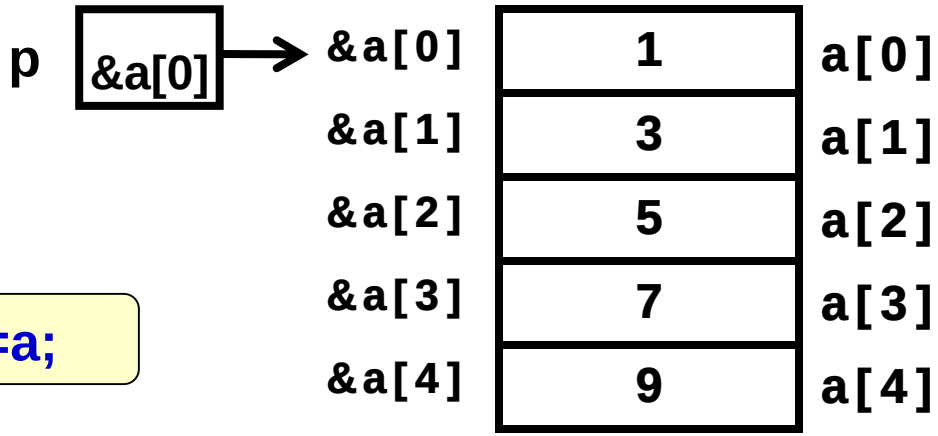
定义指向数组元素的指针

- 可以用一个指针变量指向一个数组元素

```
int a[5]={1,3,5,7,9};
int *p;
p = &a[0];
```

等价于int *p=a;
或int *p=&a[0];

等价于p=a;



注意：数组名a不代表整个数组，只代表数组首元素的地址。“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”。

一段代码揭示指针与数组的关系

```
1. #include <stdio.h>
2. int main()
3. {
4.     int a[10]={0,1,2,3,4,5,6,7,8,9};
5.     int *p;
6.     p = &a[0];

7.     printf("a[0]=%X;\n", a[0]);
8.     printf("&a[0]=%X;\n", &a[0]);
9.     printf("&a=%X;\n", &a);
10.    printf("a=%X;\n", a);
11.    printf("p=%X;\n", p);
12.    printf("*p=%X;\n", *p);

13.    return 0;
14.}
```

```
a[0]=0;
&a[0]=22FF14;
&a=22FF14;
a=22FF14;
p=22FF14;
*p=0;
```

通过指针方便地访问数组元素

- **基本事实**：数组元素在内存中是顺序地、连续地存储的，即逻辑上相邻的数组元素在物理地址上也是相邻的
- 于是，假设指针p指向数组a的第一个元素，我们就可以通过将指针p向前或者向后移动来访问数组a中的任意元素

通过指针访问数组其他元素实例

```
1. #include <stdio.h>
2. int main()
3. {
4.     int a[10]={0,1,2,3,4,5,6,7,8,9};
5.     int *p;
6.     p = &a[0];

7.     printf("a[5] = %d;\n", a[5]);
8.
9.     *(p+5) = 50;
10.    printf("a[5] = %d;\n", a[5]);

11.    printf("*(p+5) = %d;\n", *(p+5));

12.    return 0;
13.}
```

```
a[5] = 5;
a[5] = 50;
*(p+5) = 50;
```

引用数组元素时指针的运算

- 在指针指向数组元素时，允许以下运算：
  - 加一个整数(用+或+=)，如 p+1
  - 减一个整数(用-或-=)，如 p-1
  - 自加运算，如 p++，++p
  - 自减运算，如 p--，--p
  - 两个指针相减，如 p1-p2 (只有 p1和 p2都指向同一数组中的元素时才有意义)

(1) 如果指针变量p已指向数组中的一个元素，则p+1指向同一数组中的下一个元素，p-1指向同一数组中的上一个元素。

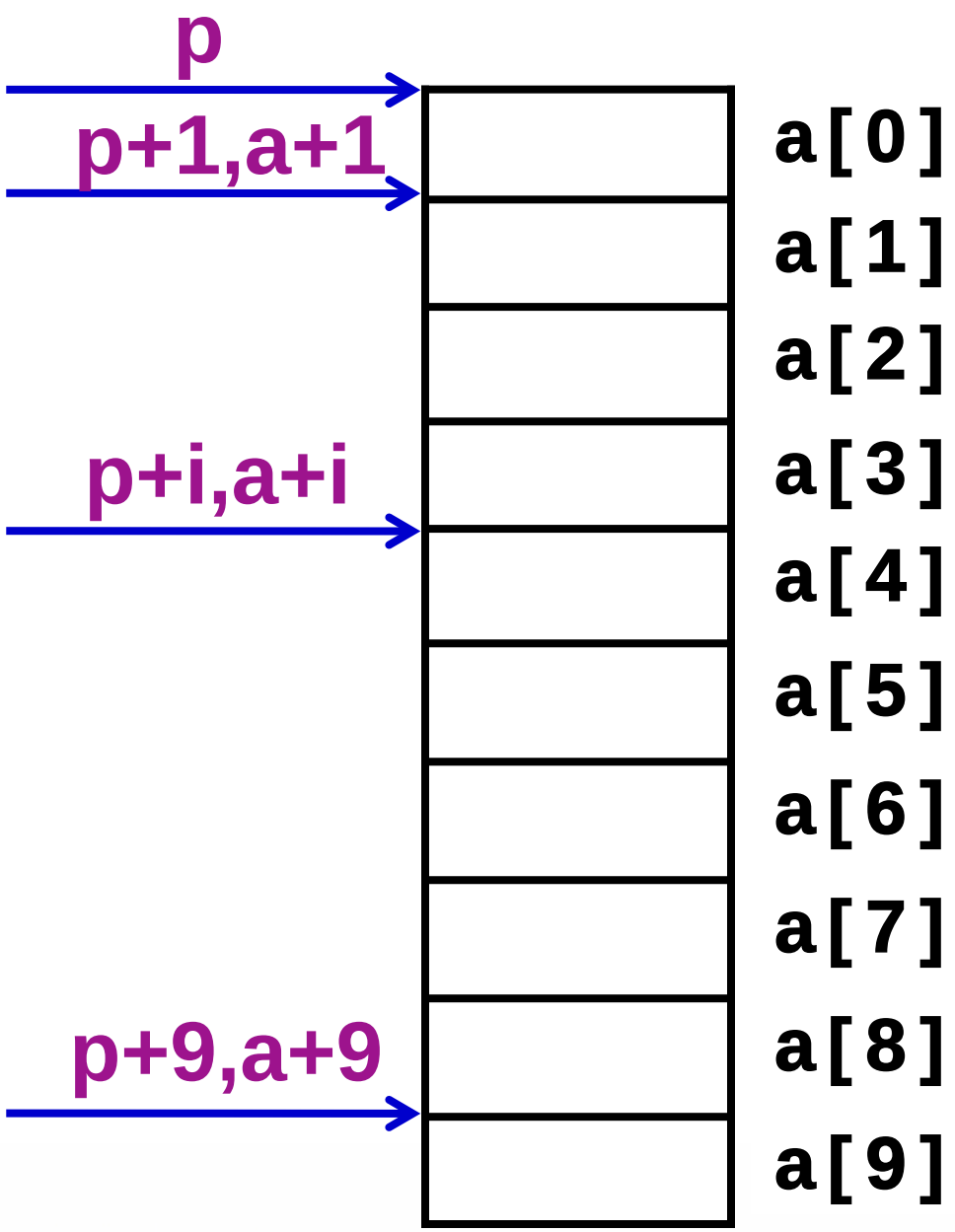
对于代码：`float a[10],*p=a;`

假设a[0]的地址为2000，则

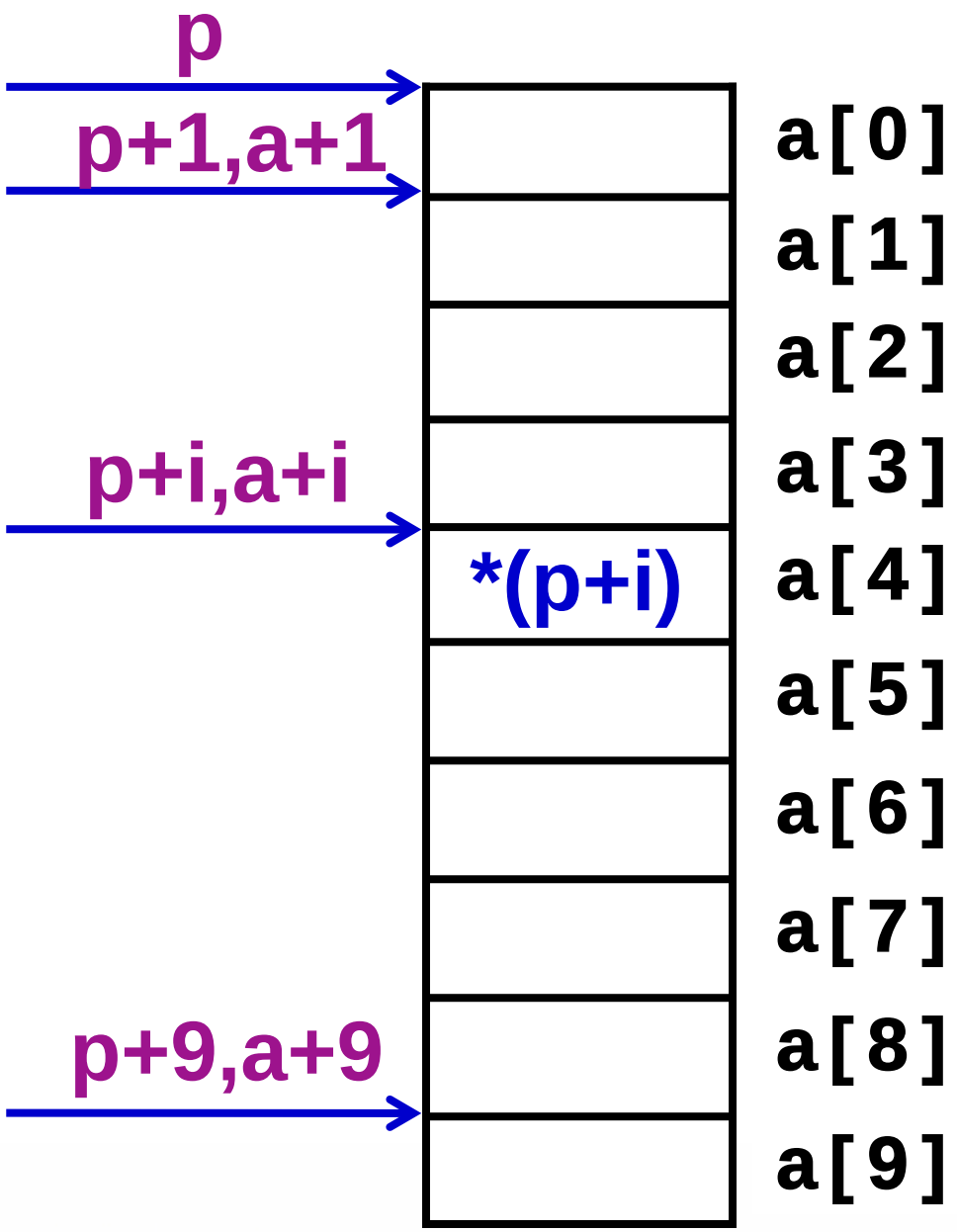
- p 的值为 2000
- p+1 的值为 2004
- P-1 的值为 1996

越界

(2) 如果 p 的初值为 &a[0]，则 p+i和a+i就是数组元素a[i]的地址，或者说，它们指向 a 数组序号为 i 的元素



(3) *(p+i)或*(a+i)是p+i或a+i所指向的数组元素，即a[i]。

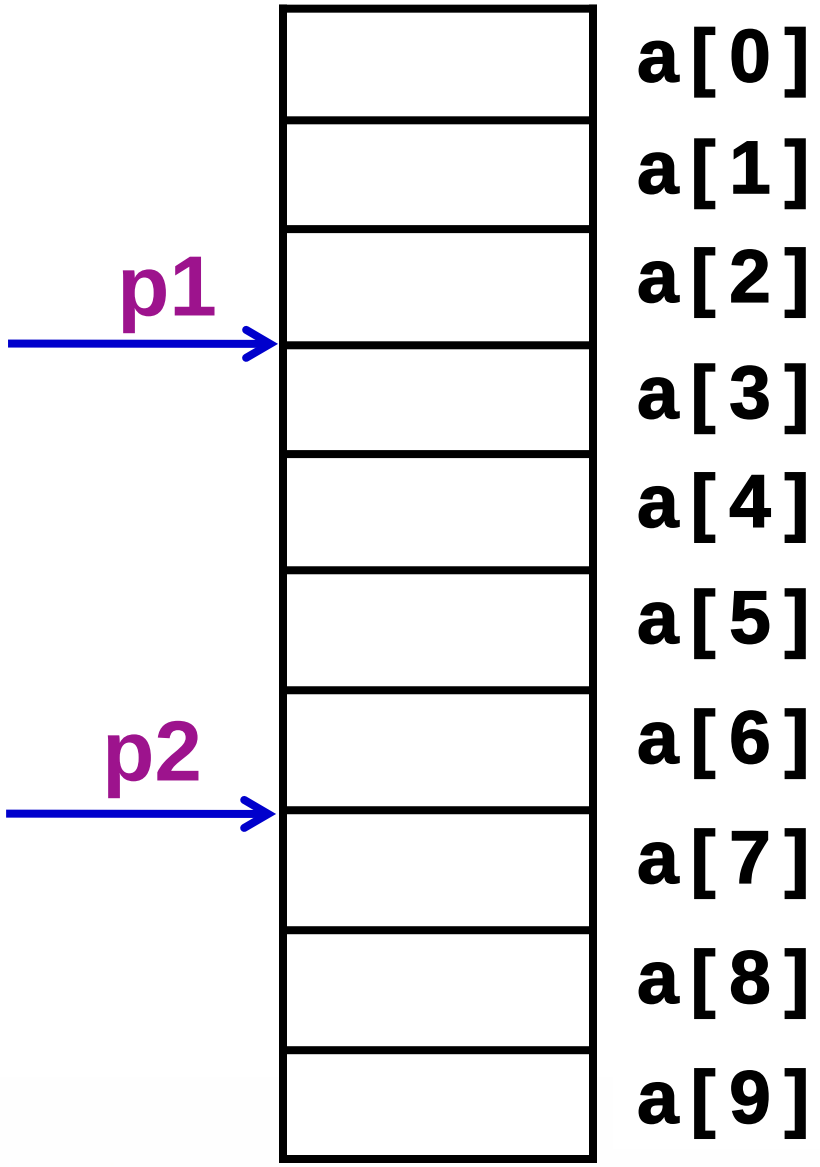


（4）如果指针p1和p2都指向同一数组

p2-p1的值是4

注意：不能计算p1+p2

因为没有任何意义！



作业 2017/12/06

- **按下列要求编写程序，提交手写源代码**

1. **主函数中定义字符数组s[1000]，并调用下列函数**

   (1) 写一个函数getInput(char s[])，接受一段字符串输入，长度不超过1000

   (2) 写一个函数inverse(char s[])，把(1)中输入的字符串的内容颠倒，如“abc”->”cba“

2. **（选做）不允许调用库函数，不允许定义全局变量或在函数体内定义局部变量，能否编写函数一个my_strlen函数实现strlen的功能？函数原型：`int my_strlen(char* strDest)`**

思考题 1-1

- 下列代码的运行结果是什么？

```
#include <stdio.h>
int main()
{
    int a[5]={1,2,3,4,5};

    int *ptr = (int*)(&a+1);
    printf("%d,%d", *(a+1), ptr[-1]);

    return 0;
}
```

思考题 1-2

- 下列代码的运行结果是什么？

```
#include <stdio.h>
#include <string.h>

int main()
{
    char a[1000];
    int i;
    for (i=0; i<1000; i++)
    {
        a[i] = -1-i;
    }
    printf("%d", strlen(a));
    return 0;
}
```