

Welcome

# 数据科学与大数据技术专业 程序设计基础(C语言)

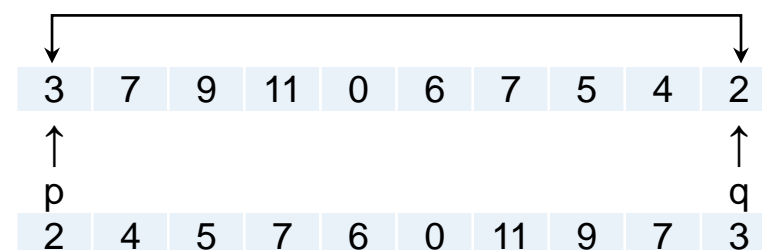
上海体育学院经济管理学院

Wu Ying

# 指针例

【例8.8】将数组a中n个整数按相反顺序存放。

```
34 void reverse(int array[], int n)
35 {
36     int* h, * t;
37     int temp;
38     h = array;
39     t = array + n-1;
40     while (h < t)
41     {
42         temp = *h;
43         *h = *t;
44         *t = temp;
45         h++;
46         t--;
47     }
48 }
```



# 指针例

【例8.10】用指针方法对10个整数按由大到小顺序排序。（选择排序法）

```
16 void selectsort(int* arr, int n)
17 {
18     int t, i;
19     int* min,*p;
20     for (i = 0; i < n ; i++)
21     {
22         min = arr + i;
23         for (p = arr+i; p < arr + n; p++)
24         {
25             if (*p < *min)
26             {
27                 t = *p;
28                 *p = *min;
29                 *min = t;
30             }
31         }
32     }
33 }
```

# 指针例

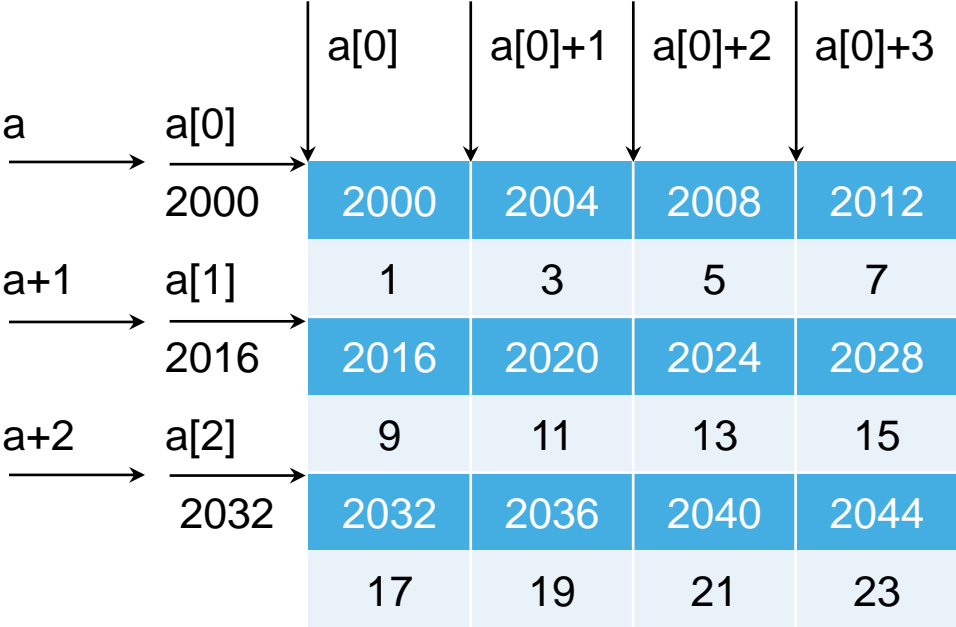
【例8.10】用指针方法对10个整数按由大到小顺序排序。（选择排序法）

```
16 void selectsort(int* arr, int n)
17 {
18     int t, i;
19     int* min,*p;
20     for (i = 0; i < n ; i++)
21     {
22         min = arr + i;
23         for (p = arr+i; p < arr + n; p++)
24         {
25             if (*p < *min)
26             {
27                 t = *p;
28                 *p = *min;
29                 *min = t;
30             }
31         }
32     }
33 }
```

# **\*通过指针引用多维数组**

# 多维数组元素的地址

```
int a[3][4]={1,3,5,7},{9,11,13,15},{17,19,21,23}};
```

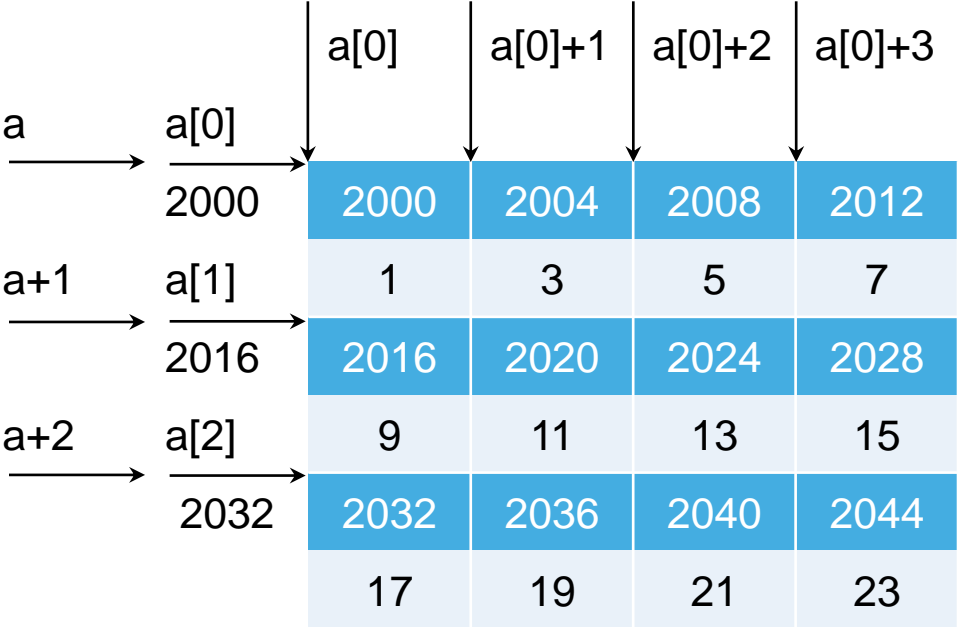


| 表示形式                           | 含义 | 地址 |
|--------------------------------|----|----|
| a                              |    |    |
| a[0], *(a+0), *a               |    |    |
| a+1, &a[1]                     |    |    |
| a[1], *(a+1)                   |    |    |
| a[1]+2, *(a+1)+2, &a[1][2]     |    |    |
| *(a[1]+2), *(*a+1)+2), a[1][2] |    |    |



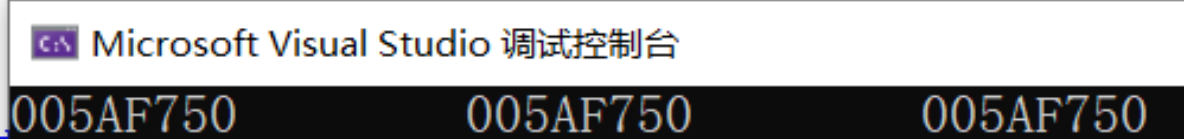
# 多维数组元素的地址

```
int a[3][4]={ {1,3,5,7},{9,11,13,15},{17,19,21,23}};
```



| 表示形式                               | 含义                       | 地址       |
|------------------------------------|--------------------------|----------|
| a                                  | 二维数组名，指向一维数组a[0]，即0行起始地址 | 2000     |
| a[0], *(a+0), *a                   | 0行0列元素地址                 | 2000     |
| a+1, &a[1]                         | 指向第1行起始地址                | 2016     |
| a[1], *(a+1)                       | 1行0列元素a[1][0]的地址         | 2016     |
| a[1]+2, *(a+1)+2, &a[1][2]         | 1行2列元素a[1][2]的地址         | 2024     |
| *(a[1]+2),<br>*(*(a+1)+2), a[1][2] | 1行2列元素a[1][2]的值          | 是元素值，为13 |

```
int a[3][4] = { {1, 3, 5, 7}, {9, 11, 13, 15}, {17, 19, 21, 23} };
printf("%p\t", *(a+1));
printf("%p\t", a[1]);
printf("%p\t", &a[1][0]);
```



```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include<stdio.h>
3  int main()
4  {
5      int a[3][4] = { {1, 3, 5, 7}, {9, 11, 13, 15}, {17, 19, 21, 23} };
6      printf("%p\n", a);
7      printf("%p\n", a + 1);
8      printf("%p\n", a[1]);
9      printf("%p\n", a[1] + 1);
10     printf("%p\n", &a[1][0]);
11     printf("%p\n", &a[1][1]);
12     printf("%d\n", *(a[1]+2));
13     printf("%d\n", a[1][2]);
14     return 0;
```



```

1  #define _CRT_SECURE_NO_WARNINGS
2  #include<stdio.h>
3  int main()
4  {
5      int a[3][4] = { {1, 3, 5, 7}, {9, 11, 13, 15}, {17, 19, 21, 23} };
6      printf("%p\n", a);
7      printf("%p\n", a + 1);
8      printf("%p\n", a[1]);
9      printf("%p\n", a[1] + 1);
10     printf("%p\n", &a[1][0]);
11     printf("%p\n", &a[1][1]);
12     printf("%d\n", *(a[1]+2));
13     printf("%d\n", a[1][2]);
14     return 0;
15 }

```

Microsoft Visual Studio 调试控制台

```

009DFB3C
009DFB4C
009DFB4C
009DFB50
009DFB4C
009DFB50
13
13

```

# 多维数组元素的地址

---

a是二维数组名，它是二维数组的首行起始地址，——一维数组的起始地址

如果用一个指针变量 pt 来指向此一维数组，应当这样定义：

```
int (*pt)[4];
```

//表示pt指向由4个整型元素组成的一维数组，此时指针变量pt的基类型是由4个整型元素组成的一维数组

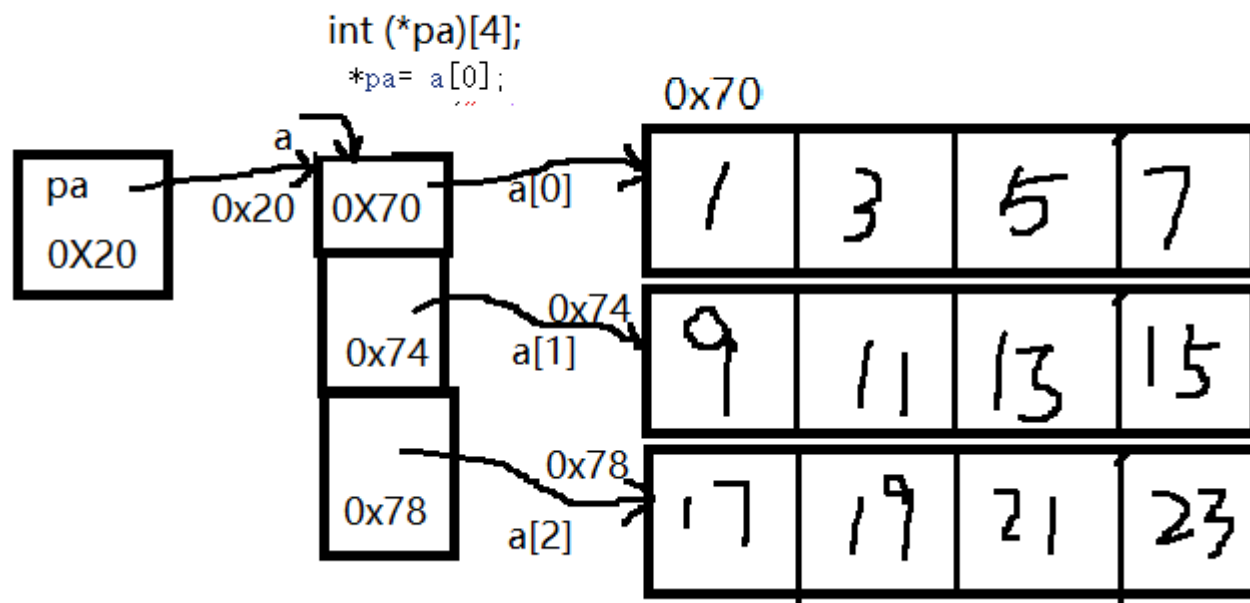
---

# 指向二维数组的指针

```
int (*p)[4];
```

p的类型是int (\*)[4]型，指向一维整型数组的指针变量，一维数组有4个元素

```
printf("%d\n", *(*(pa+2)+3));
```





# 用二维数组指针输入二维数组元素

---



# 用二维数组指针输入二维数组元素

```
srand((unsigned)time(NULL));
```

```
for (i = 0; i < 3; i++)  
{  
    for (j = 0; j < 4; j++)  
    {  
        *(*pa + j) = rand() % 50;  
    }  
    pa++;  
}
```



# 指向二维数组的指针

【例8.12】有一个 $3 \times 4$ 的二维数组，用指向二维数组的指针显示其行元素

# 指向二维数组的指针

【例8.12】有一个3×4的二维数组，用指向二维数组的指针显示其行元素

```
47  for (i = 0; i < 3; i++)
48  {
49      printrow(a+i);
50      printf("\n");
51  }
```

```
6  void printrow(int(*pa)[4])
7  {
8      for (int i = 0; i < 4; i++)
9          printf("%d\t", *(*pa + i));
10 }
```

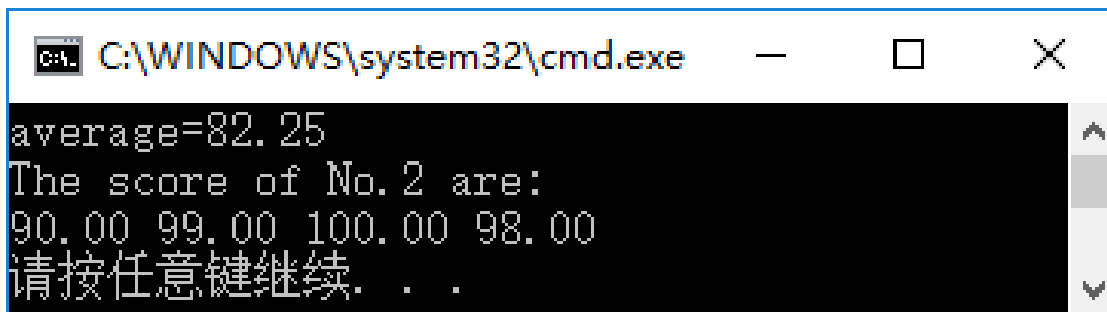
【例8.14】有一个班，3个学生，各学4门课，计算总平均分数以及第n个学生的成绩。

```
#include <stdio.h>
int main()
{
    void average(float *p,int n);
    void search(float (*p)[4],int n);
    float
    score[3][4]={65,67,70,60},{80,87,90,81},{90,99,100,98}};
    average(*score,12);           //求12个分数的平均分
    search(score,2);              //求序号为2的学生的成绩
    return 0;
}

void average(float *p,int n)      //定义求平均成绩的函数
{
    float *p_end;
    float sum=0,aver;
    p_end=p+n-1;
```

```
    //n的值为12时，p_end的值是p+11，指向最后一个元素
    for(;p<=p_end;p++)
        sum=sum+(*p);
    aver=sum/n;
    printf("average=%5.2f\n",aver);
}
```

```
void search(float (*p)[4],int n)
//p是指向具有4个元素的一维数组的指针
{
    int i;
    printf("The score of No.%d are:\n",n);
    for(i=0;i<4;i++)
        printf("%5.2f ",*(*(p+n)+i));
    printf("\n");
}
```



```
C:\WINDOWS\system32\cmd.exe
average=82.25
The score of No.2 are:
90.00 99.00 100.00 98.00
请按任意键继续. . .
```



【例8.15】在例8.14的基础上，查找有一门以上课程不及格的学生，输出他们的全部课程的成绩。

```
6 void printrow(int(*pa)[4])
7 {
8     for (int i = 0; i < 4; i++)
9         printf("%d\t", *(*pa + i));
10 }
11 void search(int(*pa)[4], int n)
12 {
13     int flag;
14     int i, j;
15     for (i = 0; i < n; i++)
16     {
17         flag = 0;
18         for (j = 0; j < 4; j++)
19         {
20             if (*(*pa + i) + j) < 60)
21                 flag = 1;
22         }
23         if (flag == 1)
24         {
25             printf("No. %d 学生存在不及格\n", i);
26             for (j = 0; j < 4; j++)
27             {
28                 printf("%d\t", *(*pa + i) + j));
29             }
30             printf("\n");
31         }
32     }
33 }
```

```
34 int main()
35 {
36     int a[3][4]; // = { 65, 81, 91, 90, 81, 91, 90, 85, 91, 90, 85, 57 };
37     int i, j;
38     srand((unsigned)time(NULL));
39     for (i = 0; i < 3; i++)
40     {
41         for (j = 0; j < 4; j++)
42         {
43             a[i][j] = rand() % 51 + 50; // 50-100
44         }
45     }
46     for (i = 0; i < 3; i++)
47     {
48         printrow(a+i);
49         printf("\n");
50     }
51     search(a, 3);
52
53     return 0;
54 }
55
56 }
```

**【例8.15】** 在例8.14的基础上，查找有一门以上课程不及格的学生，输出他们的全部课程的成绩。

【例8.15】在例8.14的基础上，查找有一门以上课程不及格的学生，输出他们的全部课程的成绩。

```
#include <stdio.h>
int main()
{   void search(float (*p)[4],int n); //函数声明
    float
score[3][4]={65,57,70,60},{58,87,90,81},{90,99,100,98}};
    //定义二维数组函数score
    search(score,3);           //调用search函数
    return 0;
}

void search(float (*p)[4],int n)
//形参p是指向包含4个float型元素的一维数组的指针变量
{   int i,j,flag;
    for(j=0;j<n;j++)
```

```
    {   flag=0;
        for(i=0;i<4;i++)
            if(*(*(p+j)+i)<60) flag=1;
        if(flag==1)
        {   printf("No. %d fails, his scores are:\n",j+1);

            for(i=0;i<4;i++)
                printf("%5.1f ",*(*(p+j)+i));

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

C:\WINDOWS\system32\cmd.exe

```
No.1 fails, his scores are:
 65.0  57.0  70.0  60.0
No.2 fails, his scores are:
 58.0  87.0  90.0  81.0
请按任意键继续. . .
```