

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



课程内容

程序设计基础 (C语言)



第5周 程序设计概述

计算机语言及C语言

最简单的C语言程序

运行C程序的步骤和方法

程序设计的任务

算法概念及简单算法

算法表示

流程图

伪代码

第6周 C程序设计基础

常量、变量及数据类型

常量变量常量变量转义字符

标识符关键字数据类型

运算符和表达式及类型转换

C程序结构和C语句

数据的输入输出

第7周 选择结构

选择结构和条件判断

运算符和表达式

关系运算符和关系表达式

逻辑运算符和逻辑表达式

条件运算符和条件表达式

选择结构嵌套

switch 语句

第8周 循环结构

while语句和do...while语句

for语句

循环嵌套, break, continue

循环练习

第9周 数组 (一)

一维数组定义、引用、初始化

一维数组练习

二维数组定义引用初始化

多维数组及练习

第10周 数组 (二) 及函数 (一)

字符数组初始化及输入输出

字符串处理函数

函数定义与调用

函数返回值、声明、原型、嵌套调用

第11周 函数 (二)

递归调用

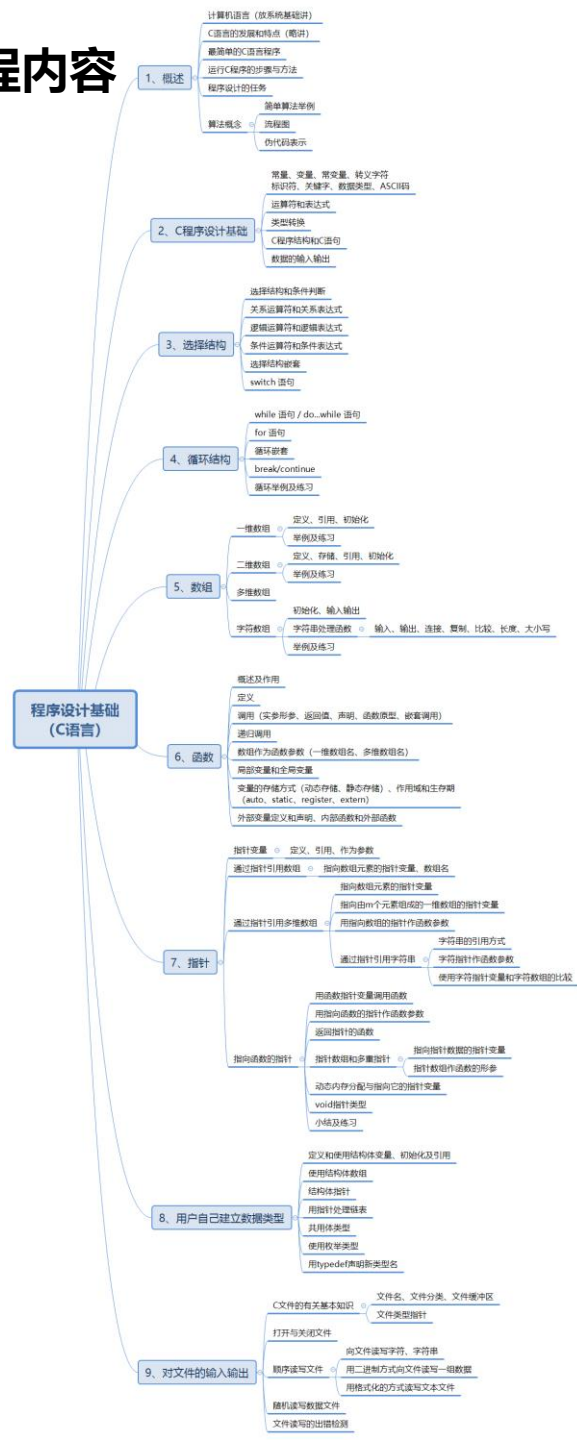
数组作为函数参数

变量作用域和生存期

变量存储方式、内部函数与外部函数



课程内容



第12周 指针 (一)

指针变量

通过指针引用数组

通过指针引用多维数组 (I)

指向数组元素的指针变量

指向一维数组的指针变量

第13周 指针 (二)

通过指针引用多维数组 (II)

用指向数组的指针作为函数参数

通过指针引用字符串

字符串的引用方式

字符指针作函数参数

使用字符指针变量和字符串数组的比较

第14周 指针 (三)

指向函数的指针 (I)

用函数指针变量调用函数

用指向函数的指针作函数参数

返回指向指针的函数

指向函数的指针 (II)

指针数组和多重指针

指向指针数据的指针变量

指针数组作函数的形参

动态内存分配与指向它的指针变量

第15周 指针 (四) 及结构体 (一)

指向函数的指针 (IV)

void指针类型

小结及练习

结构体

定义和使用结构体变量

使用结构体数组、结构体指针

第16周 结构体 (二) 及文件 (一)

用指针处理链表、共用体类型

枚举类型、使用typedef声明新类型

C文件的基本知识

打开与关闭文件

第17周 文件 (二)

顺序读写文件

随机读写数据文件

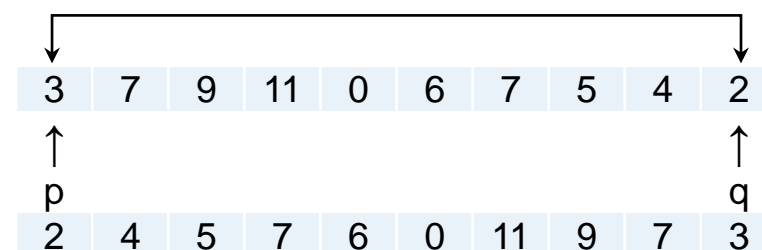
总复习练习

总复习练习

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 }
```

***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};
```

		a[0]	a[0]+1	a[0]+2	a[0]+3
a	a[0]	2000	2004	2008	2012
a+1	a[1]	1	3	5	7
		2016	2020	2024	2028
a+2	a[2]	9	11	13	15
		2032	2036	2040	2044
		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), *(*(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]);
```



Microsoft Visual Studio 调试控制台

005AF750

005AF750

005AF750

```

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

多维数组元素的地址

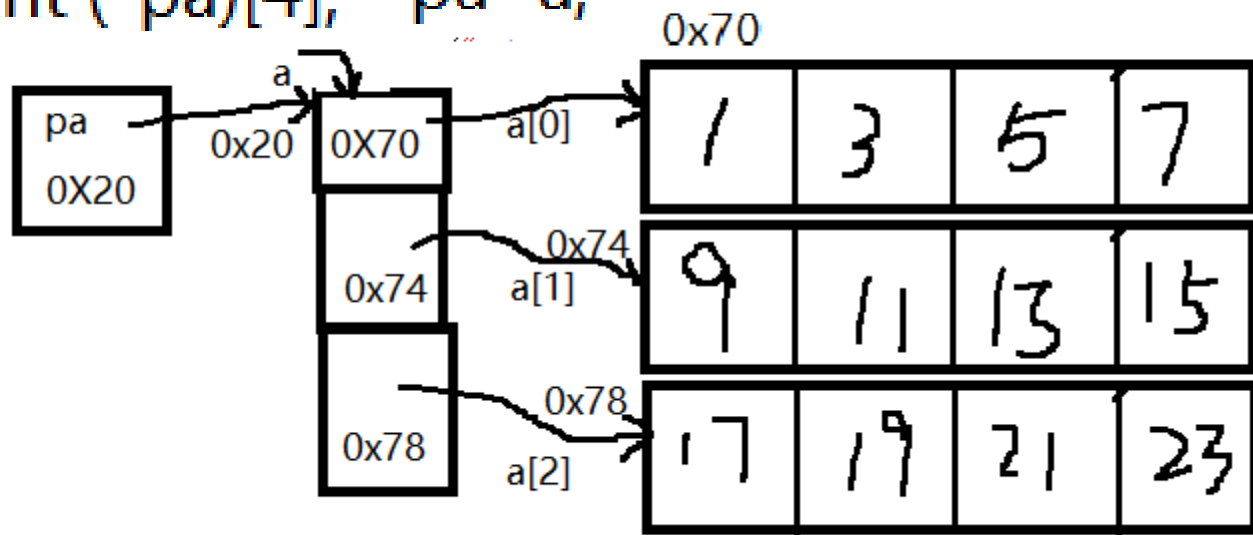
a是二维数组名，它是二维数组的首行起始地址，——一维数组的起始地址

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

```
int (*pa)[4]; pa=a;
```



```
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      int* (pa)[4]; //定义一个指针，指向有4个元素的一维数组
7      pa[0]=a[0]; // *pa= a[0];
8
9      printf("%p\n", a[0]);
10     printf("%p\n", pa[0]);
11     printf("%p\n", *pa+0);
12     printf("%d\n", *(*pa + 0));
13     printf("%d\n", a[0][0]);
14
15     return 0;
16 }
```

Microsoft Visual Studio 调试控

00B9F7E4
00B9F7E4
00B9F7E4

1
1

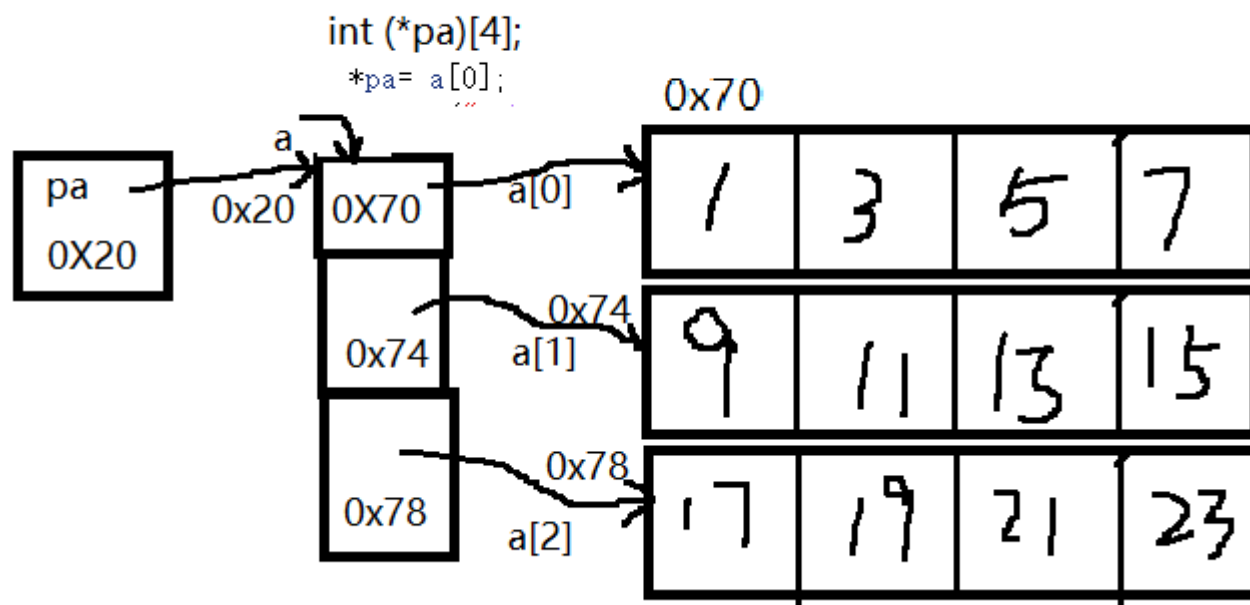
C:\Users\HP\Desktop\C

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));
```





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第二行的元素

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第二行的元素

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include<stdio.h>
3  void printRow(int(*p)[4], int row);
4  int main()
5  {
6      int a[3][4] = { {1, 3, 5, 7}, {9, 11, 13, 15}, {17, 19, 21, 23} };
7      printRow(a, 2);
8      return 0;
9  }
10 void printRow(int(*pa)[4], int row)
11 {
12     int i;
13     for (i = 0; i < 4; i++)
14     {
15         printf("%d\t", (*(pa + row) + i));
16     }
17     printf("\n");
18 }
```

Microsoft Visual Studio 调试控制台

17 19 21 23

【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。

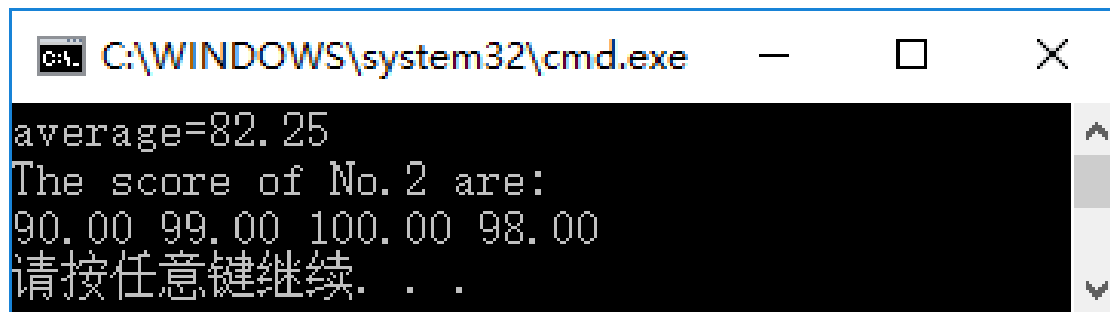
```
#include <stdio.h>
int main()
{
    void average(float *p,int n);
    void search(float (*p)[4],int n);
    float
    score[3][4]={65,67,70,60},{80,87,90,81},{90,99,100,98}};
    average(*score,12);           //求12个分数的平均分
    search(score,2);              //求序号为2的学生的成绩
    return 0;
}

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。

```
#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. . .
```