**江西软件职业技术大学信息工程学院**

**教 案**



二〇一九年

**目 录**

[程序设计基础（C）教案 1](#_Toc7084)

[第一次课 初识C语言 1](#_Toc418)

[第二章 变量与数据类型 7](#_Toc18623)

[第三章 运算符与表达式 13](#_Toc95)

[第四章 顺序结构与选择结构 20](#_Toc2758)

[第五章 循环结构 30](#_Toc3884)

[第六章 函数 43](#_Toc12506)

[第七章 一维数组 51](#_Toc12831)

[第八章 二维数组 56](#_Toc25232)

[第九章 结构体 61](#_Toc16110)

# 程序设计基础（C）教案

## 第一次课 初识C语言

**教学目的**

使用VC2010创建C程序项目、了解C程序的执行过程、编写第一个C语言程序。掌握进制的转换，原码、反码、补码的转换。

**教学要求**

1. 熟练掌握C程序的执行过程
2. 掌握第一个C语言程序的编写
3. 熟练掌握不同进制的转换

**重点和难点**

1. C程序的执行过程
2. C语言程序的编写
3. 不同进制的转换

**教学支撑环境与教学资源：**

1. 笔记本电脑
2. VC2010
3. 投影仪

**教学方法**

讲授法、讨论法 练习法

**授课课时**

6课时

**教学过程**

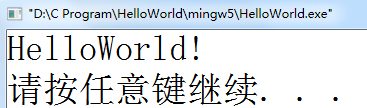
**------------------------------------AM-------------------------------------**

一、课程引入

讲述C语言的用途和发展前景、直接使用C语言编写的游戏项目进行演示。

二、第一个C程序

编写程序实现输出效果如下：



该程序的源文件如下：

|  |
| --- |
| #include <stdio.h>//包含头文件stdio.h，只有包含了该头文件，才能输出  int main(void)//主函数，C程序就是从这里开始运行的  {  printf("HelloWorld!\n");//输出语句，双引号里面可以是任意的内容  return 0;  } |

创建C程序项目小结：  
1>选择Visual C++，Win32控制台

2>添加源文件.c，C++源文件.cpp,头文件.h

3>项目文件.sln

三、main主函数

1.只能有一个主函数，必须要有一个主函数，C程序从主函数开始运行。

2.int main(void),int：返回值类型，main：函数名，void：参数

3.return 0，返回值是0。

四、注释

1.单行注释//

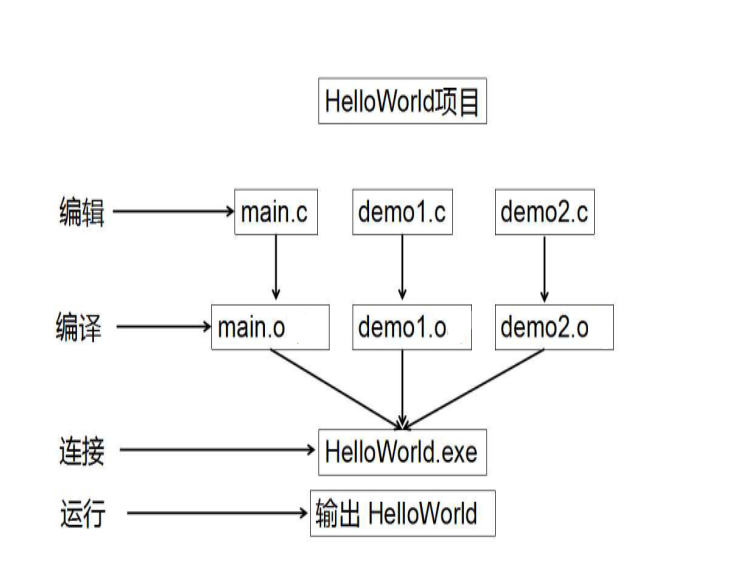
2.多行注释/\* \*/

注意：不能在多行注释中嵌套多行注释

注释一个函数时通常用多行注释，注释某个语句用单行注释

|  |
| --- |
| #include <stdio.h>//包含头文件stdio.h，只有包含了该头文件，才能输出  /\*  int:函数返回值类型，整型  main:函数名，主函数  void:函数参数，空的  \*/  int main(void)//主函数，C程序就是从这里开始运行的  {  printf("HelloWorld!\n");//输出语句，双引号里面可以是任意的内容  return 0;//返回0  } |

五、C程序执行的过程：



编辑---写代码的过程，生成.c文件

编译---检查语法错误的过程，生成.obj文件

连接---多个.obj文件合并成一个.exe文件的过程

执行---运行.exe文件的过程

六、C程序的结构

项目--->文件--->函数--->语句--->单词

项目：后缀是.sln

文件：源文件后缀是.c

函数：可以有多个函数，但一定要有主函数

语句：以分号;结尾

单词：不是单纯的英文单词，而是标识符，标识符又分成：关键字、预定义标识符、自定义标识符。C语言的标识符命名规范：由字母、数字、下划线构成且第一个字符不能是数字，且不能是系统占用的单词。

练习：在程序中分别找出哪些是关键字、预定义标识符、自定义标识符？

**------------------------------------PM-------------------------------------**

七、进制的转换

在计算机内存储和运算数据时，通常要涉及到的数据单位有以下3种：

位（bit）：计算机中的数据都是以二进制来表示的，二进制的代码只有“0” “1”两个数码，采用多个数码（0和1的组合）来表示一个数，其中的每一个数码称为一位，位是计算机中最小的数据单位。

字节（Byte）：在对二进制数据进行存储时，以8位二进制代码为一个单元存放在一起，称为一个字节，即1 Byte =8 bit。字节是计算机中信息组织和存储的基本单位，也是计算机体系结构的基本单位。在计算机中，通常用B（字节）、KB（千字节）、MB（兆字节）或GB（吉字节）为单位来表示存储器（如内存、硬盘、U盘等）的存储容量或文件的大小。

字长：人们将计算机一次能够并行处理的二进制代码的位数，称为字长。字长是衡量计算机性能的一个重要指标，字长越长，数据所包含的位数越多，计算机的数据处理速度越快。计算机的字长通常是字节的整倍数，如8位、16位、32位、64位和128位等。

1.内存容量

1TB--->1024GB

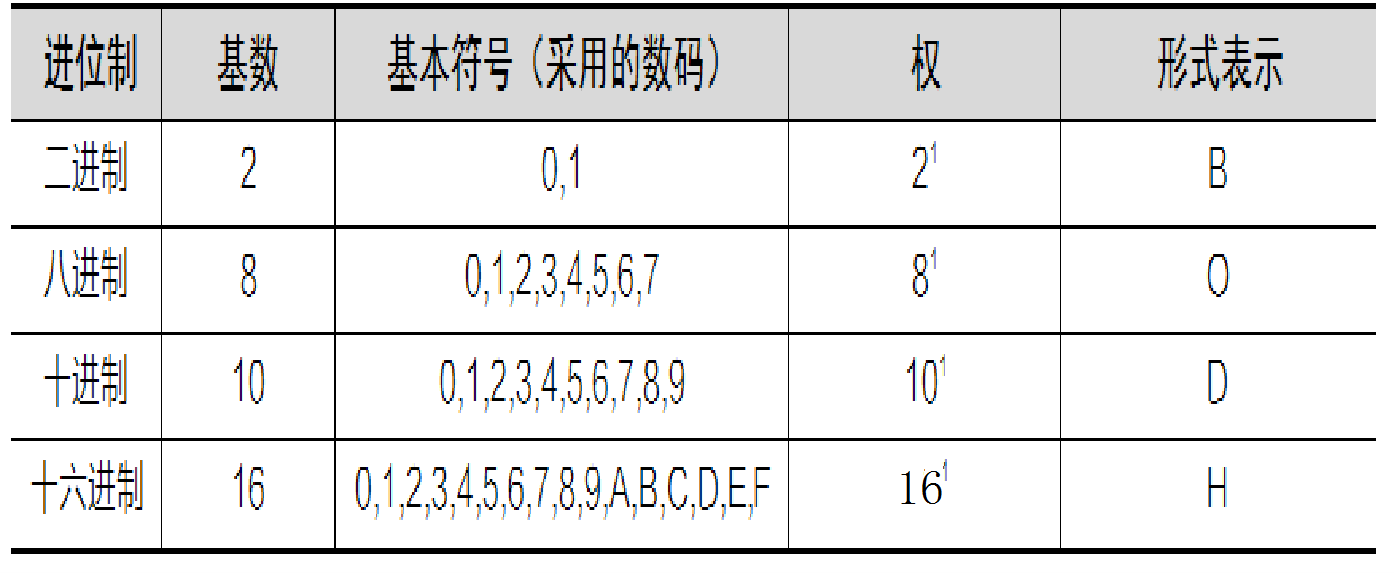
1GB--->1024MB

1MB--->1024KB

1KB--->1024B byte(字节)

1byte--->8bit(位)

数制是指用一组固定的符号和统一的规则来表示数值的方法。如下图所示为计算机中常用的几种进位计数制的表示。



2.二进制与十进制的转换

1>将二进制数转换成十进制

10110转换成十进制数:先将二进制数10110按位权展开，再对其乘积相加，转换过程如下所示。

（10110）2=（1×24+0×23+1×22+1×21+0×20）10

=（16+4+2）10

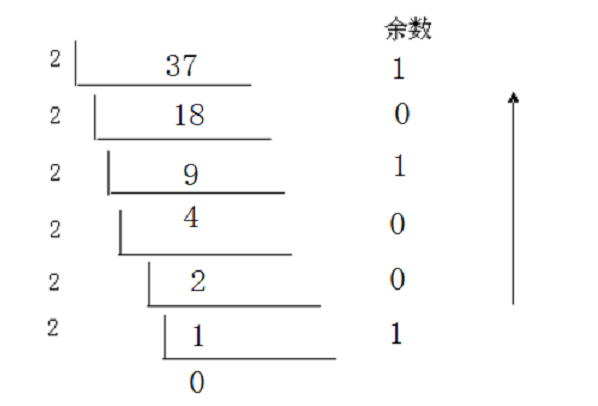
=（22）10

8421法（128 64 32 16 8 4 2 1）

1011 1101--->128+0+32+16+8+4+0+1=189

2>将十进制转换成二进制数

除以2逆向取余法：除以2，逆向取余数，直到商为0终止。



37--->100101

3.二进制与八进制的转换

1>将八进制数转换成二进制:每1个八进制位转换成3个二进制位，左边不足三个的则补0。

567--->101110111

234--->010011100

2>将二进制数转换成八进制：从右向左，每3个二进制位为一组，每一组转换成1个八进制数。

011 010 111 100 110--->32746

011 010 111 101 110---32756

011 101 110 101 011---35653

4.二进制与十六进制的转换

1>将十六进制数转换成二进制:每1个十六进制位转换成4个二进制位，左边不足4个的则补0。

0----0000

1----0001

2----0010

3----0011

4----0100

5----0101

6----0110

7----0111

8----1000

9----1001

A----1010 10

B----1011 11

C----1100 12

D----1101 13

E----1110 14

F----1111 15

1289ADF--->0001 0010 1000 1001 1010 1101 1111

2>将二进制数转换成十六进制:从右向左，每4个2进制位为一组，每一组转换成1个十六进制数。

1110 1101 1111 1001 1110--->EDF9E

课堂练习：

1. 请问十进制数89，二进制数100111，八进制数234，十六进制数1A2中哪个最大？

八、原码、反码、补码

数据在计算机里面都是以补码的形式存储。

原码 反码 补码

37:0010 0101--->0010 0101--->0010 0101

-37:1010 0101--->1101 1010--->1101 1011

正数的原码、反码、补码都是一样的！

负数的反码是在原码的基础上“符号位不变，数值位取反”

负数的补码是在反码的基础上“符号位不变，数值位加1”

3-2=1

3+（-2）= -2:1000 0010---1111 1101---1111 1110

0000 0011

+1111 1110

=0000 0001

-2:1000 0000 0000 0010

1111 1111 1111 1101

1111 1111 1111 1110

九、课后作业

1.编写程序并输出“这是我的第一个C程序!”

2.编写一个菜单界面

========成绩管理系统========

========选择菜单序号========

======== 1.添加成绩 ========

======== 2.删除成绩 ========

======== 3.修改成绩 ========

======== 4.查询成绩 ========

======== 5.成绩排序 ========

======== 6.求平均分 ========

======== 7.求最高分 ========

======== 8.求最低分 ========

======= 9.查询所有成绩 =======

======= 0.退出管理系统 =======

======请输入您选择的序号======

## 第二章 变量与数据类型

**教学目的**

掌握变量的定义及使用，掌握常量的基本概念和使用，掌握C语言中的基本数据类型，掌握各个数据类型占据的内存空间大小与取值范围。

**教学要求**

1. 掌握变量的定义及使用
2. 掌握常量的概念与使用
3. 熟练掌握C语言中的基本数据类型
4. 熟练掌握各个数据类型占据的内存空间大小与取值范围

**重点和难点**

1. 变量的定义与使用
2. C语言的数据类型
3. 各个类型占据的字节数与取值范围

**教学支撑环境与教学资源：**

1. 笔记本电脑
2. VC2010
3. 投影仪

**教学方法**

讲授法、练习法、讨论法

**授课课时**

6课时

**教学过程**

1. 课程引入

在生活中，我们用整数进行计数，比如今天上课的人数，用小数记录比较精确的数据，比如我们的成绩，这些都是我们最常用的数据类型。

二、变量

变量的概念：值可以发生改变的量

定义语法：类型名 变量名1,变量名2,变量名3;

类型名 变量名1 = 初值,变量名2 = 初值;

char a,b,c;

int e;

char a = 0,b = 0,c = 0;

char a = 0;

unsigned char e = 0;

int height = 170;

int weight = 0;

注意：

1.变量名以字母、数字、下划线构成且第一个字符不能是数字，要符合标识符的命名规范，且不能是系统占用的单词，如关键字int、预定义标识符printf等。

char 8Tom;//错误的，数字不能作为第一个字符

char Tom MJ;//错误的，有空格

char Tom\_MJ;//正确的

char MJ8;//正确的

int a\*b;//错误的，\*不符合标识符命名规范

int ab&c;//错误的，&不符合标识符命名规范

int \_ab;//正确的

char int;//错误，int是关键字，不能用来作为变量名

//变量名的字母分大小写，ab和AB不是同一个变量名

int ab;

int AB;

2.变量要先定义再使用

a = 45;//错误，a要先定义再使用

char a;

3.变量是占据内存的，占据的内存大小由变量的数据类型决定

如：char a;//a占据1byte

a = 10;

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 变量a占据的1byte | | | | | | | | |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

课堂练习2：

1.使用不同数据类型分别定义变量，并写出该变量占据的内存大小。

unsigned char a;//a占据1byte

int b;//b占据4byte

unsigned int c;//c占据4byte

short int d;//d占据2byte

long int e;//e占据4byte

long long int f;//f占据8byte

float g;//g占据4byte

double h;//h占据8byte

4.变量的本质就是一块内存空间，变量名就是这块内存的名字，通过变量名对内存进行写或者读。

变量的赋值(写操作)和变量的输出(读操作)：

int main(void)

{

char a,b;

a = 45;//把45赋值给a

//（把45写入a的内存块中）

b = a;//先读出a的值(45)，再把45赋值给b

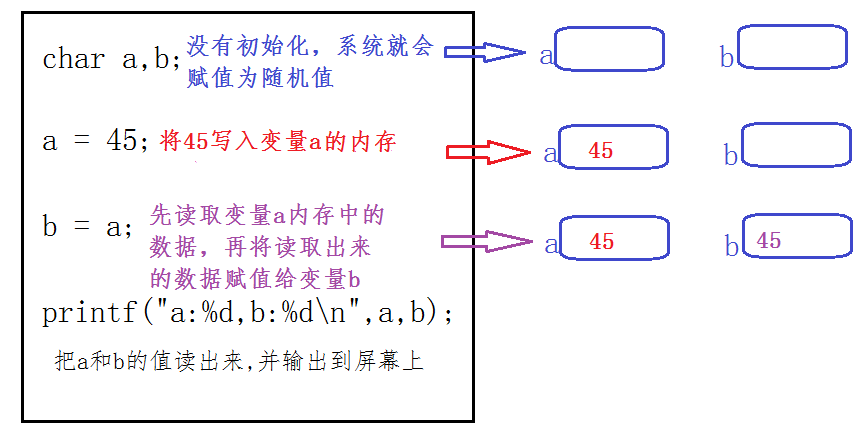
//（把45写入b的内存块中）

printf("a:%d,b:%d\n",a,b);//把a和b的值读出来

return 0;

}

以上程序的赋值过程如下：



5.要养成给变量赋初值为0的习惯

int a = 0;//在变量定义时，给变量赋值叫赋初值

a = 0;//不是赋初值，这叫赋值

6.定义变量时要根据需求来确定数据类型，防止“溢出”（超出取值范围）

#include <stdio.h>

int main(void)

{

//有符号的char类型数据范围：-128 -- 127

char y = 127;

char x = -128;

char a = x - 1;

char b = y + 1;

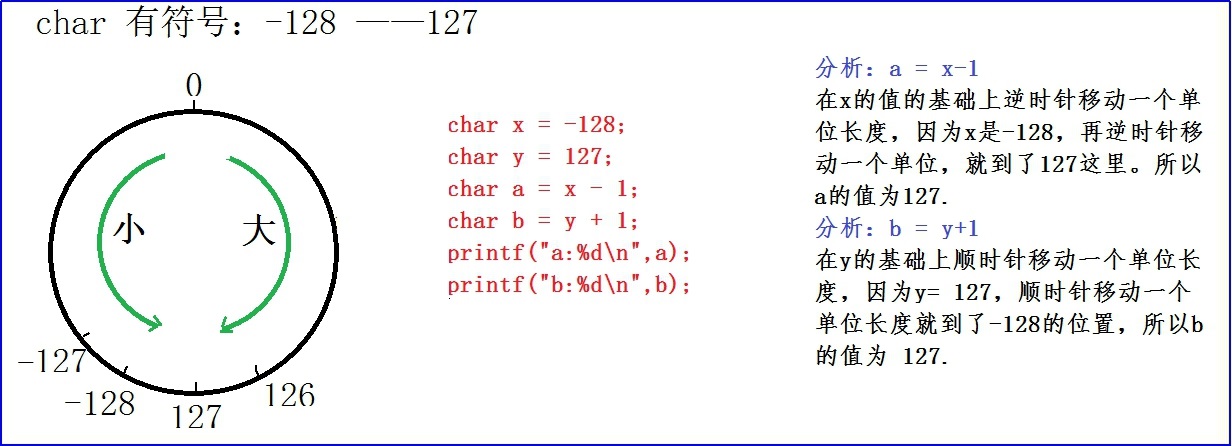
printf("a:%d\n",a);

printf("b:%d\n",b);

//a 和 b 的值是多少

return 0;

}



7.单词、关键字、标识符的区别

语句由单词构成

单词:即标识符:

关键字:在VC2010中蓝色的单词，如int

自定义标识符:自己定义的标识符，如char abc中的abc

预定义标识符:系统已经定义好的标识符，如printf

课堂练习3：指出以下程序中哪些是关键字、自定义标识符和预定义标识符。

int main(void)

{

unsigned char height = 0;

printf("");

return 0;

}

C语言的标识符命名规范：由字母、数字、下划线构成且第一个字符不能是数字，且不能是系统占用的单词。

课堂练习4：指出并修改以下程序中的错误。

int main(void)

{

int float = 0;//错误的，float是关键字

int printf = 0;//错误的，printf是预定义标识符

int weight = 0;//正确的

printf("");

return 0;

}

三、数据类型

C语言的数据类型分为：基本类型和构造类型两种（本次讲解基本数据类型）。本学期课程均以VC2010为编译器为标准讲解，其他编译器会有些差别。

1.基本类型（字符型、整型、浮点型）

1>字符型：特殊的整型，占据1个字节(1byte---8bit)

有符号字符型signed char/char:最高位（最左边）为符号位

取值范围:1000 0000(-128) ----0111 1111（127）

无符号字符型unsigned char:没有符号位，都是数值位

取值范围:0000 0000(0)---1111 1111(255)

2>整型：整数类型

有符号的整型 signed int/int，占据4byte（32位）

1000 0000 0000 0000 0000 0000 0000 0000---

0111 1111 1111 1111 1111 1111 1111 1111

无符号的整型 unsigned int，占据4byte

0000 0000 0000 0000 0000 0000 0000 0000---

1111 1111 1111 1111 1111 1111 1111 1111

short int:占据2byte

1000 0000 0000 0000(-32768)---0111 1111 1111 1111(32767)

unsigned short int:占据2byte

0000 0000 0000 0000(0)---1111 1111 1111 1111(65535)

long int:占据4byte

long long int:占据8byte

3>实型:小数，浮点型，没有无符号的浮点型数据

float:单精度浮点型，占据4byte

double:双精度浮点型，占据8byte

long double:长双精度浮点型，占据8byte或16byte(VC2010中是8byte)

总结：不同的编译器，数据类型所占字节数大小不同

可以使用sizeof进行检验

例如：printf(“%d\n”,sizeof(int));//输出int类型所占字节数的大小

下表是用VC2010的结果

|  |  |  |
| --- | --- | --- |
| **数据类型** | **字节数** | **取值范围** |
| 字符型 | | |
| char | 1 | -2^7——2^7-1 |
| unsigned char | 1 | 0——2^8-1 |
| 整型 | | |
| short | 2 | -2^15——2^15-1 |
| unsigned short | 2 | 0——2^8-1 |
| int | 4 | -2^31—— 2^31-1 |
| unsigned int | 4 | 0——2^32 -1 |
| long | 4 | -2^31—— 2^31-1 |
| unsigned long | 4 | 0——2^32-1 |
| long long | 8 | -2^63——2^63 -1 |
| unsigned long long | 8 | 0——2^64-1 |
| 浮点型 | | |
| float | 4 | -/+3.4e38（精确到6位小数） |
| double | 8 | -/+1.7e308（精确到15位小数） |
| long double | 8 | -/+1.19e4932（精确到15位小数） |

2.构造类型：

数组、指针、结构体、共用体、枚举类型

这些知识后面再讲解

课堂练习1：

1.写出char , int ,short int 的取值范围，使用二进制表示。

## 第三章 运算符与表达式

**教学目的**

理解运算符和表达式的基本概念，掌握各类运算符的优先级和结合性，掌握各类算术运算符和表达式的运用，掌握关系运算符的使用，掌握逻辑运算符的运用，掌握不同数据类型进行运算时的类型转换原则。

**教学要求**

1. 掌握算术运算符、赋值运算符
2. 掌握关系运算符、逻辑运算符、逻辑表达式
3. 掌握运算符的优先级、强制类型转换

**重点和难点**

1. 运算符的优先级
2. 算术运算符的使用
3. 关系运算符的使用、逻辑运算符的运用

**教学支撑环境与教学资源：**

1. 笔记本电脑
2. VC2010
3. 投影仪

**教学方法**

讲授法、练习法、讨论法

**授课课时**

6课时

**教学过程**

**------------------------------------AM------------------------------**

一、课程引入

数学中常用的算术运算符在C语言中又是如何使用的？数学中常用的关系运算符在C语言中又是如何使用的？逻辑判断的原则有哪些？

二、运算符与表达式

运算符：+ - \* / %

表达式：由操作数和运算符组成的式子，如 a-b,1+2

C语言中单一的一个变量或者单一的一个常量是最简单的表达式

提问：以下程序是输出变量的值还是表达式的值？

int main(void)

{

char a = 0;

char b = 0;

a = 10;

b = 15;

printf("%d",a);

printf("%d\n",a+b);

}

总结：永远输出表达式的值！！！

1.优先级与结合性

优先级:参照附录C P359 从1开始逐渐减弱

结合性:同一优先级时，从左到右或者从右到左的运算顺序。

例如表达式:1+2\*3-4/2，该表达式的计算过程是先计算2\*3 和 4/2，因为\* /的优先级比+ -高，而\* /是同优先级，结合性（即运算顺序）是从左往右，所以先算2\*3，然后再算4/2，之后再去计算1+6-2。

2.单目、双目和三目运算符

单目运算符：运算对象只有1个，如：++ -- !

双目运算符：运算符两侧各有一个运算对象，如： + - \* /

三目运算符：运算对象有三个，如: ? :

表达式1 ? 表达式2 : 表达式3

10>15 ？ 10 : 15

三.算术运算符和算术运算表达式:

1. + - \* / %

+ - \* /:只要有一个运算对象是浮点型，则表达式的值为double型。

课堂练习1：写出以下程序中的输出结果。

int main(void)

{

float a = 0;

float c = 0;

a = 7.5;

c = 7 / 2.0;//c = 3.500000000000000

printf("%f\n", c);//

c = 7/2;

printf("%f\n", c);//

c = a\*2;

printf("%f\n", c);//

return 0;

}

注意：%求余运算符的两个运算对象必须都是整数！

20 % 6.5//错误，不能是小数

课堂练习2：写出以下表达式的结果。

int a = 10,b = 3,c = -7;

a/b

a%b

b%a

c%a

a%c

2.自增++ 自减--

1> ++后缀:先读后写

int main(void)

{

int a = 0;

a = 20;

a++;//相当于a = a + 1; 21

printf("%d\n", a++);//先读后写,先把a 的值(21)读出来，再a = a + 1;

printf("%d\n", a);//22

}

printf("%d\n", a++);

相当于

printf("%d\n", a);

a = a + 1;

1. ++前缀:先写后读

int main(void)

{

int a = 0;

a = 20;

++a;//相当于a = a + 1; a = 21

printf("%d\n", ++a);//先写后读,先a = a + 1;再把a的值(22)读出来

printf("%d\n", a);//22

}

printf("%d\n", ++a);

相当于

a = a + 1;

printf("%d\n", a);

四、关系运算符和关系运算表达式:> < == >= <= !=

关系运算表达式的值为“真”或者“假”用“1”和“0”表示。

课堂练习3：写出以下表达式的值。

int a = 15;

int b = 10;

a > b//1

a < b//0

a < 15//0

a > 15//0

a >= 15//1

a == 15//1

a == b//0

a != b//1

**----------------------------------PM--------------------------------**

五、逻辑运算符和逻辑运算表达式

逻辑运算表达式的值为“真”或者“假”用“1”和“0”表示。

参与运算的对象只要不为0，则运算对象为真。

1.&&逻辑与运算符:参与运算的对象都为真(不为0)，结果才是真。

课堂练习4：写出以下表达式的值。

int a = 15;

int b = 10;

a > b && a < b//0

a > b && a != b//1

a && b//1

2.||逻辑或运算符:参与运算的对象只要有一个为真(不为0)，结果就是真，只有两个都是假的，结果才是假。

课堂练习5：写出以下表达式的值。

int a = 15;

int b = 10;

a > b || a < b//1

a > b || a != b//1

a < b || a == b//0

a || b//1

3.!逻辑非运算符:使表达式的值由真变假,由假变真

课堂练习6：写出以下表达式的值。

int a = 15;

int b = 10;

!(a > b)//0

!(a < b)//1

!a//0

六、赋值运算符和赋值运算表达式:

简单赋值运算符:=

变量 = 表达式;

a = 45;

复合的赋值运算符: += -= \*= /= %=

int a = 15;

a += 10;//a = a + 10; a = 25;

a -= 10;//a = a - 10; a = 15;

a \*= 2;//a = a\*2; a = 30;

a /= 2;//a = a/2; a = 15;

a %= 6;//a = a % 6; a = 3;

a++;//相当于a += 1;

a / = 10;//错误，不能有空格

七、sizeof求字节运算符

sizeof(表达式) 结果为表达式的数据类型占据的字节数

课堂练习8：写出以下表达式的值。

int a = 10;

sizeof(int)//4

sizeof(unsigned int)//4

sizeof(a)//4

sizeof(165)//4

sizeof(a + 10)//4

sizeof(float)//4

sizeof(double)//8

sizeof(long double)//8

八、强制类型转换运算符:(类型名)(表达式)

课堂练习7：写出以程序的输出结果。

int main(void)

{

float a = 0;

int b = 0;

int c = 0;

float d = 0;

a = 7.5;

b = 2;

d = 1.5;

c = a \* b;//c = 15.000 000 000 000 000;

printf("%d\n", c);//15

c = (int)a \* b;//c = 7\*2;

printf("%d\n", c);//14

c = (int)(a \* b);//c = 15;

printf("%d\n", c);//15

c = a\*b\*d;//c = 22.5; c = 22

printf("%d\n", c);//22

c = (int) a\*b\*d;

printf("%d\n", c);//21

c = (int)a\*b\*(int)d;

printf("%d\n", c);//14

}

九、不同类型的数据混合运算时为了保证运算的精度，计算机会自动转向精度高的数据类型进行转换 char 一定会转换为int

float 一定会转换为double

char b = 0;

float c = 0;

char d = 'M';

b = 'A'+ 32;//b = 65+32;

c = 7 / 2.0f;//c = 3.500 000 000 000 00;

//7.000 000 000 000 00 /2.000 000 000 000 00

十、扩展知识----逗号运算符和逗号运算表达式:,

表达式1,表达式2,表达式3,...

逗号表达式的最终值是最后一个表达式的值，但是所有的表达式都会运行。

课堂练习7：写出以下表达式的值。

int main(void)

{

int a = 0;

int b = 0;

int c = 0;

a = 5;

b = 15;

c = 10;

printf("%d\n", (++a, ++b, c++));//10

printf("%d\n",a);//6

printf("%d\n", b);//16

}

十一、课后作业

1.写出以下表达式的值。

int a = 20,b = 11;

float c = 3.5;

a>b||!a

b+8!=a&&b<=a

c\*a

(int)c\*a

(a+1,b++,a+b)

2.编程实现：求出C语言成绩、PS成绩与英语成绩加起来后的平均分。

## 第四章 顺序结构与选择结构

**教学目的**

熟练掌握顺序结构中复合语句的用法，熟练掌握C语言选择结构中的if语句、if else语句、switch语句。

**教学要求**

1. 熟练掌握顺序结构中复合语句的用法
2. 熟练掌握if语句、if else语句、switch语句

**重点和难点**

1. 复合语句的用法
2. if语句、if else语句、switch语句的应用

**教学支撑环境与教学资源：**

1. 笔记本电脑
2. VC2010
3. 投影仪

**教学方法**

讲授法、练习法、讨论法

**授课课时**

6课时

**教学过程**

**---------------------------------AM---------------------------------**

一、课程引入

计算1+2+3+...+10有多少种方法？

二、算法的基本结构

算法：解决问题的方法(结构)

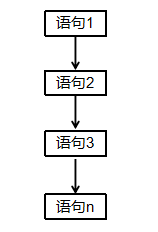
计算从1开始加到10的结果

算法一：1+2+3+4+5+6+7+8+9+10=55

算法二：(1+10)\*10/2=55

算法的三种基本结构：顺序结构、选择结构（分支结构）和循环结构。

三、顺序结构:按语句书写的先后顺序依次执行的结构。



1.复合语句：由大括号{}中的0个或多个声明和语句列表共同构成。

课堂练习1：写出以下程序的输出结果。

int main(void)

{

int a = 0;

int b = 0;

a = 5;

b = 4;

{

int a = 6;//变量声明

printf("%d\n",a);//这个a的值是6，和前面的a不是同一个

a++;//自增表达式语句

printf("%d\n",a);//函数调用语句

++b;

}

printf("%d\n", a);//这个a的值是5

printf("%d\n",b);

return 0;

}

注意：

1>复合语句不以分号;作为结束符，即最后的}后面不加分号。

2>在复合语句中可以定义变量，但仅在复合语句中有效，即作用域是{}括起来的部分。

课堂练习2：指出下列程序的错误。

int main(void)

{

int b = 0;

b = 4;

{

int a = 6;

printf("%d\n",a);

a++;

printf("%d\n",a);

++b;

};//错误，不能加;

printf("%d\n", a);//错误，a没有定义

printf("%d\n",b);

return 0;

}

四、选择结构（分支结构）:根据条件判断来选择执行哪一条语句。

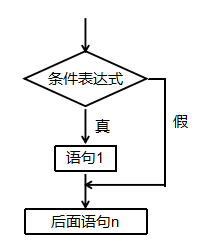
选择结构的语句有：if条件语句和switch多路分支语句。

1.if语句

if(条件表达式)

语句1;

执行过程：如果条件表达式的值为真，则执行语句1，否则不执行语句1。



课堂练习3:输入a和b的值，判断并输出最大值

int main(void)

{

int a = 0;

int b = 0;

int max = 0;

printf("请依次输入a和b的值:");

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

max = a;

if(max < b)

max = b;

printf("max:%d\n",max);

return 0;

}

注意：if条件语句中的语句1不是只有一条语句的意思，可以是多条语句用{}括起来的复合语句。

int main(void)

{

int a = 0;

int b = 0;

int max = 0;

printf("请依次输入a和b的值:");

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

max = a;

if(max < b)

{

max = b;

printf("a < b \n");

}

printf("max:%d\n",max);

return 0;

}

2.if...else语句

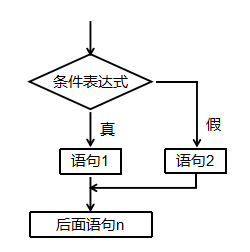
if(条件表达式)

语句1;

else

语句2;

执行过程:如果条件表达式的值为真，则执行语句1，否则执行语句2。



课堂练习4:输入a和b的值，判断并输出最大值

int main(void)

{

int a = 0;

int b = 0;

int max = 0;

printf("请依次输入a和b的值:");

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

if(a < b)

max = b;

else

max = a;

printf("max:%d\n",max);

return 0;

}

3.if...else if语句

if(条件表达式1)

语句1;

else if(条件表达式2)

语句2;

else if(条件表达式3)

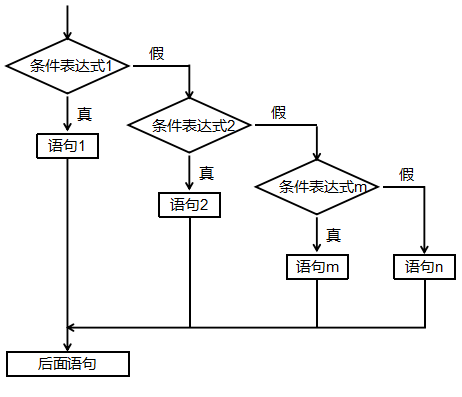
语句3;

....

else

语句n;

执行过程：依次判断条件表达式的值，当某个值为真时，则执行相应的语句，然后跳出整个if语句之外，继续执行后面的程序。如果所有的表达式都为假，则执行语句n，然后继续执行后面的程序。



课堂练习5:输入成绩分数，并判断该成绩属于哪个等级

int main(void)

{

int score = 0;

printf("请输入成绩:");

scanf("%d",&score);

if (score >= 0 && score <= 100)

{

if(score >= 90)

printf("A+\n");

else if(score >= 80)

printf("A\n");

else if(score >= 70)

printf("B+\n");

else if(score >= 60)

printf("B\n");

else

printf("不合格\n");

}

else

printf("请输入合法的成绩！");

return 0;

}

4.（扩展知识）if语句嵌套：当if语句中的执行语句又是if语句时，则形成了if语句嵌套。

if(条件表达式)

if语句;

或者

if(条件表达式)

if语句;

else

if语句;

课堂练习6:修改下面的程序使程序：输入a和b的值，判断最大值是否大于100，是则输出最大值。

int main(void)

{

int a = 0;

int b = 0;

printf("请依次输入a和b的值:");

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

if(a < b)

if(b > 100)

printf("max:%d\n",b);

else

if(a > 100)

printf("max:%d\n",a);

return 0;

}

注意：C语言规定，else总是与它前面最近的为配对的if配对。

所以修改后应为：

int main(void)

{

int a = 0;

int b = 0;

printf("请依次输入a和b的值:");

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

if(a < b)

{

if(b > 100)

printf("max:%d\n",b);

}

else

{

if(a > 100)

printf("max:%d\n",a);

}

return 0;

}

课堂练习7::输入a、b、c的值，判断并输出最大值

int main(void)

{

int a = 0;

int b = 0;

int c = 0;

int max = 0;

printf("请依次输入a、b、c的值:");

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

if (a > b)

{

if (a > c)

max = a;

else

max = c;

}

else//b > a

{

if (b > c)

max = b;

else

max = c;

}

printf("max:%d\n", max);

return 0;

}

**------------------------------------PM------------------------------**

5.switch多路分支语句

switch(表达式)//1

{

case 常量表达式1:语句1(集合);

case 常量表达式2:语句2(集合);

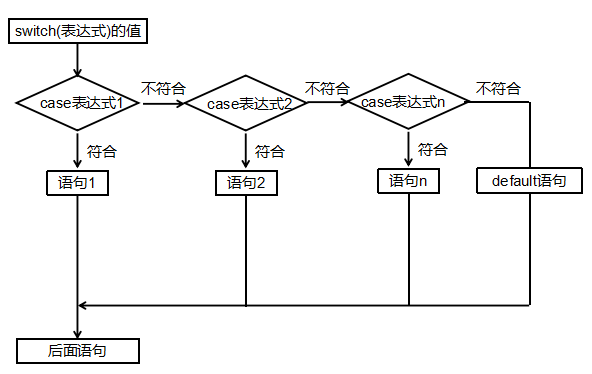
...

case 常量表达式n:语句n(集合);

default:语句n+1;

}

执行过程：首先计算表达式的值，与常量表达式i进行比较，如果与其中一个常量表达式i的值相等，就执行其后的语句直到遇到break语句才结束switch语句，如果case后无break语句，则继续执行随后所有的case后的语句。如果没有找到与表达式的值相匹配的常量表达式，则执行default后的语句n+1。



课堂练习8:输入数字1-7，并输出显示对应的星期

int main(void)

{

int week = 0;

printf("今天星期几：");

scanf("%d", &week);//1---'1'

switch (week)//week是表达式

{

case 1:printf("今天是星期一\n"); break;

case 2:printf("今天是星期二\n"); break;

case 3:printf("今天是星期三\n"); break;

case 4:printf("今天是星期四\n"); break;

case 5:printf("今天是星期五\n"); break;

case 6:printf("今天是星期六\n"); break;

case 7:printf("今天是星期天\n"); break;

default:printf("输入数据有误!\n");

}

return 0;

}

注意：

1>switch后面括号中的表达式可以是整型、字符型和枚举型。

2>在case后的各常量表达式的值不能相同

3>在case后，允许有多个语句，可以不用{}括起来，而整个switch结构一定要有一对{}

4>各case和default语句的先后顺序可以改变，不影响程序执行结果。

课堂练习9:demo：指出以下程序的错误

int main(void)

{

float week = 0;

printf("今天星期几：");

scanf("%f", &week);

switch (week);//week不能是浮点型，

//不能有;,缺少{

case1:printf("今天是星期一\n"); break;//case 后面要空格

case 2:printf("今天是星期二\n"); break;

case 3:printf("今天是星期三\n"); break;

case 4:printf("今天是星期四\n"); break;

case 5:printf("今天是星期五\n"); break;

case 6:printf("今天是星期六\n"); break;

case 7:printf("今天是星期天\n"); break;

default:printf("输入数据有误!\n");

//缺少}

return 0;

}

6.条件运算符和条件运算表达式: (?:)

表达式1 ? 表达式2 :表达式3

求值过程：如果表达式1的值为真，则以表达式2的值作为条件表达式的值，否则以表达式3的值作为条件表达式的值.

课堂练习10:输入a、b的值，判断并输出最大值，使用条件表达式。

int main(void)

{

int a = 0;

int b = 0;

int max = 0;

printf("请依次输入a和b的值:");

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

max = a>b?a:b;

printf("max:%d\n",max);

return 0;

}

五、课后作业

1.编程实现：从键盘输入C语言的成绩后输出对应的成绩等级，要求使用switch选择语句实现。

## 第五章 循环结构

**教学目的**

掌握循环结构中的while语句、do while语句、for语句的用法，掌握中断语句中break和continue语句的用法。

**教学要求**

1. 熟练掌握while语句、do while语句、for语句
2. 熟练掌握break语句、continue语句

**重点和难点**

1. while语句、do while语句、for语句的使用
2. break语句、continue语句的使用

**教学支撑环境与教学资源：**

1. 笔记本电脑
2. VC2010
3. 投影仪

**教学方法**

讲授法、练习法、讨论法

**授课课时**

6课时

**教学过程**

**---------------------------------AM---------------------------------**

一、课程引入

计算1+2+3+...+n有几种方法，需要多少步骤？

二、循环结构语句

什么时候用到循环结构:处理重复的问题的时候。

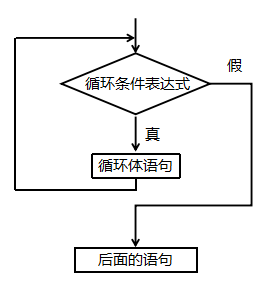
循环结构的三大语句：while语句、do while语句、for语句。

1.while语句

while(循环条件表达式)

循环体语句;

执行过程：只有循环条件表达式的值为真就 执行循环体语句，先判断后执行。



课堂练习1:

1.输出1---n之间的数

int main(void)

{

int i = 1;

int n = 0;

printf("请输入n的值:");

scanf("%d",&n);//10 15

while(i <= n)

{

printf("%d\n",i);

i++;

}

return 0;

}

2.计算m+...+n的值

int main(void)

{

int i = 1;

int n = 0;

int m = 0;

int sum = 0;

printf("请依次输入m和n的值:");

scanf("%d%d",&m,&n);//m+...n 10 15

while (m <= n)

{

sum = sum + m;

m++;

}

printf("sum=%d\n", sum);

return 0;

}

2.do...while语句

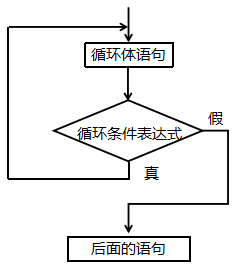
do

{

循环体语句;

}while(循环条件表达式);

执行过程：先执行循环体语句，再检查循环条件表达式的值是否为真，如果为真则继续 执行循环体语句，否则结束循环。



课堂练习2:

1.计算m+...+n的值，使用do while语句。

int main(void)

{

int i = 1;

int n = 0;

int m = 0;

int sum = 0;

printf("请依次输入m和n的值:");

scanf("%d%d",&m,&n);//m+...n 10 15

do

{

sum = sum + m;

m++;

}while (m <= n);

printf("sum=%d\n", sum);

return 0;

}

2.输出1---n之间的数，使用do while语句。

int main(void)

{

int i = 1;

int n = 0;

printf("请输入n的值:");

scanf("%d",&n);//10 15

do

{

printf("%d\n",i);

i++;

}while(i <= n);

return 0;

}

while语句和do...while语句的区别：while语句是先判断后执行，do...while语句是先 执行，至少会执行一次。

课堂练习3：写出以下程序的运行结果。

int main(void)

{

int i = 1;

int n = 11;

int m = 12;

int sum = 0;

while (m <= n)

{

sum = sum + m;

m++;

}

printf("sum=%d\n", sum);//sum的值为0

return 0;

}

int main(void)

{

int i = 1;

int n = 11;

int m = 12;

int sum = 0;

do

{//先执行一次循环体语句，再去判断条件表达式的值

sum = sum + m;

m++;

}while (m <= n);

printf("sum=%d\n", sum);//sum的值为12

return 0;

}

三、while与do while循环语句的练习

1.输出n---1之间的数

int main(void)

{

//1.定义变量

//2.输入n的值

//3.循环的语句(循环要做的事情):输出当前的数据

//4.循环退出的条件:i--;i<1或者i<=0

//5.循环的条件:i>=1或者i>0

int i = 0;

int n = 0;

printf("请输入n的值:");

scanf("%d",&n);

//i = n;

//while(i>=1)

//{

// printf("%d\t",i--);

//}

while(n>=1)

{

printf("%d\t",n--);

}

return 0;

}

课堂练习4:使用do while语句实现输出n---1之间的数。

2.计算1\*2\*3\*....\*n的值（n的阶乘n!）

int main(void)

{

//1.定义变量

//2.输入n的值

//3.循环的语句(循环要做的事情):把当前的值都\*起来

//4.循环退出的条件:i++;i>n

//5.循环的条件:i<=n

int i = 1;

int n = 0;

int sum = 1;

printf("请输入n的值:");

scanf("%d",&n);

while(i<=n)

{

sum = sum \* i;

i++;

}

printf("n!的值是:%d\n",sum);

return 0;

}

课堂练习5:使用do while语句实现计算1\*2\*3\*....\*n的值（n的阶乘n!）。

3.计算10- 1/2 - 1/3 -....1/n的值

int main(void)

{

//1.定义变量

//2.输入n的值

//3.循环的语句(循环要做的事情):把当前的值都-起来

//4.循环退出的条件:i++;i>n

//5.循环的条件:i<=n

int i = 2;

int n = 0;

float sum = 10;

printf("请输入n的值:");

scanf("%d",&n);

while(i<=n)

{

sum = sum - 1.0/i;//sum -= 1.0/i;

i++;

}

printf("1- 1/2 - 1/3 -....1/n的值是:%f\n",sum);

return 0;

}

课堂练习6:使用do while语句实现计算10- 1/2 - 1/3 -....1/n的值。

**----------------------------------PM--------------------------------**

四、for语句

for(表达式1;表达式2;表达式3)

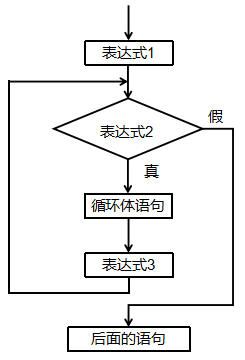
循环体语句;

说明：

表达式1:设置初始条件，只执行一次，为0个或多个变量设置初值。

表达式2：是循环条件表达式，用来判定是否继续循环。在每次执行循环体之前要先执行表达式2，然后再决定是否继续执行循环。

表达式3：作为循环的调整，比如是循环体变量增值，它是执行循环体语句之后再执行。



while语句与for语句的替换：计算1+2+3+...+n的值

使用while语句实现:

i = 1;

while(i <= n)

{

sum = sum + i;

i++;

}

使用for语句实现:

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

{

sum = sum + i;

}

课堂练习7:

1.输出1---n之间的数，使用for语句。

int main(void)

{

int i = 1;

int n = 0;

printf("请输入n的值:");

scanf("%d",&n);//10 15

/\*while(i <= n)

{

printf("%d\n",i);

i++;

}\*/

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

{

printf("%d\n",i);

}

return 0;

}

2.计算m+...+n的值，使用for语句。

int main(void)

{

int n = 0;

int m = 0;

int sum = 0;

printf("请依次输入m和n的值:");

scanf("%d%d",&m,&n);//m+...n 10 15

/\*while (m <= n)

{

sum = sum + m;

m++;

}\*/

for(;m <= n;m++)

{

sum = sum + m;

}

printf("sum=%d\n", sum);

return 0;

}

注意：

1>for语句的三个表达式不是必须的。

2>当条件表达式(表达式2)的值为假(为0)时for循环语句就结束。

3>可以在循环体内执行break,continue,goto语句。

4>表达式2是空的，表明表达式2的值一直是真，即死循环。

for(;;)//死循环语句

{

}

相当于

while(1)

{

}

小结：

for(i=m;i<n;i++)//循环次数:n-m次

for(i=m;i<=n;i++)//循环次数:n-m+1次

五、中断语句

break语句：跳出本层循环，执行循环后的语句。

continue语句：跳出本次循环，执行下一次循环。

goto语句：跳出到指定的标号位。

1.break语句

demo:计算m+...+n的值，当累加的值大于100时退出循环

int main(void)

{

int n = 0;

int m = 0;

int sum = 0;

printf("请依次输入m和n的值:");

scanf("%d%d",&m,&n);//m+...n 10 50

for(;m <= n;m++)

{

sum = sum + m;

if(sum > 100)break;

}

printf("sum=%d\n", sum);

return 0;

}

课堂练习8:输出1---n之间的数 ，当输出的数大于100时结束循环

int main(void)

{

int i = 1;

int n = 0;

printf("请输入n的值:");

scanf("%d",&n);//10 15

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

{

printf("%d\n",i);

if(i > 100)break;

}

return 0;

}

2.continue语句

只用于结束本次循环，即直接去执行表达式3.

demo:

int main(void)

{

int i = 1;

int n = 0;

int sum = 0;

printf("请输入n的值:");

scanf("%d", &n);//150

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

{

sum = sum + i;

printf("sum:%d\n",sum);

if (i > 100)continue;

printf("i=%d\n", i);

}

return 0;

}

3.goto语句（了解）

demo:

int main(void)

{

int i = 1;

int j = 0;

int k = 0;

for (j = 0; j < 2; j++)//2

{

for (i = 0; i < 15; i++)//15

{

if (i == 10)goto again;

printf("i=%d\t", i);

}

}

again:printf("穿越啦\n");

return 0;

}

六、for循环语句的练习

1.输出n---1之间的数

int main(void)

{

//1.定义变量

//2.输入n的值

//3.循环的语句(循环要做的事情):输出当前的数据

//4.循环退出的条件:i--;i<1或者i<=0

//5.循环的条件:i>=1或者i>0

int i = 0;

int n = 0;

printf("请输入n的值:");

scanf("%d",&n);

/\*i = n;

while(i>=1)

{

printf("%d\t",i--);

}\*/

for(i=n;i>=1;i--)//for(i=n;i;i--)

{

printf("%d\t",i--);

}

return 0;

}

2.计算1\*2\*3\*....\*n的值（n的阶乘n!）

int main(void)

{

//1.定义变量

//2.输入n的值

//3.循环的语句(循环要做的事情):把当前的值都\*起来

//4.循环退出的条件:i++;i>n

//5.循环的条件:i<=n

int i = 1;

int n = 0;

int sum = 1;

printf("请输入n的值:");

scanf("%d",&n);

/\*while(i<=n)

{

sum = sum \* i;

i++;

}\*/

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

{

sum = sum \* i;

}

printf("n!的值是:%d\n",sum);

return 0;

}

3.计算10- 1/2 - 1/3 -....1/n的值

int main(void)

{

//1.定义变量

//2.输入n的值

//3.循环的语句(循环要做的事情):把当前的值都-起来

//4.循环退出的条件:i++;i>n

//5.循环的条件:i<=n

int i = 2;

int n = 0;

float sum = 10;

printf("请输入n的值:");

scanf("%d",&n);

/\*while(i<=n)

{

sum = sum - 1.0/i;//sum -= 1.0/i;

i++;

}\*/

for(i=2;i<=n;i++)

{

sum = sum - 1.0/i;//sum -= 1.0/i;

}

printf("1- 1/2 - 1/3 -....1/n的值是:%f\n",sum);

return 0;

}

4.计算1+2+3+....+n的值，当累加的和大于100时退出循环

int main(void)

{

//1.定义变量

//2.输入n的值

//3.循环的语句(循环要做的事情):把当前的值都加起来

//4.循环退出的条件:i++;i>n

//5.循环的条件:i<=n

int i = 0;

int n = 0;

int sum = 0;

printf("请输入n的值:");

scanf("%d",&n);

/\*while(i<=n)

{

sum = sum + i;

if(sum>100)break;

i++;

}\*/

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

{

sum = sum + i;

if(sum>100)break;

}

printf("1+2+3+...+n的值是:%d\n",sum);

return 0;

}

七、课后作业

1.分别使用while、do while、for语句实现：100以内的奇数累加和。

2.分别使用while、do while、for语句实现：循环3次的程序。

3.分别使用while、do while、for语句实现：计算10- 1/(2+2) - 1/(3+3) -....1/(n+n)的值。

## 第六章 函数

**教学目的**

掌握函数的定义、函数的声明、函数的调用、函数的设计原则，能够灵活的去定义和调用函数。

**教学要求**

1. 掌握函数的定义、声明、调用掌握常量的概念与使用
2. 熟练掌握函数的设计原则
3. 熟练掌握函数形参的异常判断、函数返回值的使用

**重点和难点**

1. 函数定义和使用
2. 形参的异常判断、函数返回值的使用

**教学支撑环境与教学资源：**

1. 笔记本电脑
2. VC2010
3. 投影仪

**教学方法**

讲授法、练习法、讨论法

**授课课时**

6课时

**教学过程**

1. 课程引入

什么是模块化？模块化有哪些特点？我们C语言中是否可以实现模块化编程？

函数是完成一个个特定任务的语句集合，它能完成你所想要的某种特定任务，当你要用时，只需要调用它即可，在后续的修改或是维护过程中，只需要针对这一个进行修改即可。

作用1：减少程序的代码量

作用2：代码复用

作用3：使程序具有良好的结构

二、函数的定义

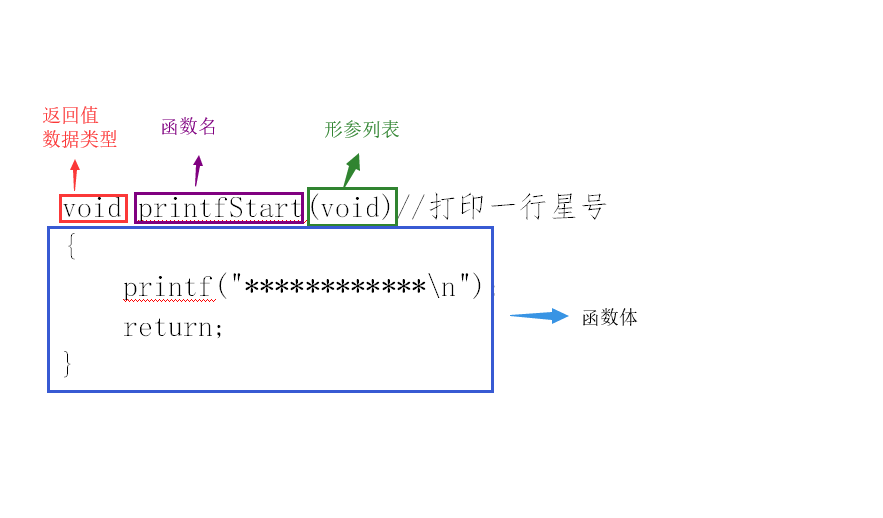
函数的定义：包括两个部分，分别是“函数头”和“函数体”。

返回值数据类型 函数名(数据类型 形参1，数据类型 形参2,....)//函数头

{

//函数体（可以有多条语句）

}



**[注意1]:**函数的返回值和参数可以是任意类型，包括空类型！！当函数无返回值 或无形参列表时，建议用void指明

**[注意2]:**函数名是一种标识符，必须符合标识符的命名规则！单词首字母不大 写，从第二个单词起首字母大写。

**[注意3]**:函数的参数可以没有，也可以有多个参数（本章最后讲解）

【思考】如果源文件只有如上说是的代码可以运行吗？

同学们自己尝试一下！

**[注意4]**：一个项目中必须要有一个主函数，没有主函数，程序找不到执行的入口，代码无法运行。

**[注意5]**：函数头要有头注释（功能，参数，返回值），函数定义不可以嵌套。

【思考】现在在上面代码的基础上添加好主函数，试试可以运行吗？

**[注意6]**：函数如果不被调用，就不会执行。

三、函数的调用

语法 : 函数名（实际参数1，实参2，...）

#include <stdio.h>

void printStar(void)

{

printf("\*\*\*\*");

return ;

}

int main(void)

{

printStar();//调用函数printStar

return 0;

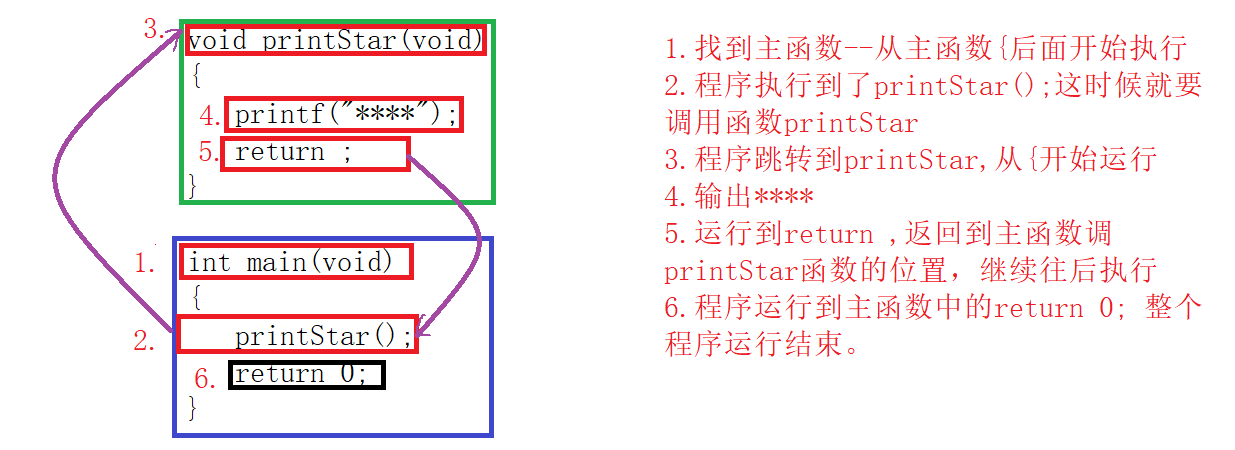
}

[注意1]：如果是A函数调用B函数，则A函数叫主调函数，B叫被调用函数

在这个代码示例中，main函数就可以称为主调函数，prinfStar函数 就可以称为被调函数。

[注意2]：如果函数返回值类型是void,那么函数中的return可以省略

[注意3]：以上程序调用函数（无返回值无参数）的执行过程如下:



【思考】如果printStar()函数放在主函数下面可以吗？

1. **函数的声明**

函数声明的好处：

(1)函数列表更加一目了然，方便使用。

(2)便于多人合作共同开发。

(3)可以用于商业，保护知识产权（只将调用方式告诉他人，但是函数中具体的实现算法隐藏）。

**声明的方式：只保留函数头且后面加分号,通常放在包含头文件的下面。**

demo:

**void printStar(void);//声明**

int main(void){ printStar(); //调用}

void printStar(void){ printf("\*\*\*\*\*\*\*\*"); }//定义

1. 函数的练习

我们先来看看如下几个练习，分析问题：

1. 定义一个函数，输出1--10之间的数据

分析：这个函数中所需要的数据已经唯一确定，不要其他函数传递数据进行辅助计算

故：该函数的定义不需要形式参数，不需要返回值，函数中需要实现的就是输出数据（数据变化的），建议使用循环。

定义形式如下（Demo）:

|  |
| --- |
| **void printData(void)**  **{**  **int i ;**  **for(i = 1; i <= 10 ;i++)**  **printf("%d\t",i);**  **return ;**  **}** |

无返回值无形参函数的调用不再赘述。

*练习1：定义一个函数，输出1，3，5，7，9,调用这个函数，看看最后的效果。*

*练习2：定义一个函数，输出2，4，6，8，10，调用这个函数，看看最后的效果。*

1. 定义一个函数，输出1--n之间的数据

分析：该函数中，n不能唯一确定，所以是其他函数传递过来的数据，因此需要形式参数。根据题目意思，形式参数的数据类型为int,此题目中没有计算结果，所以不需要返回值

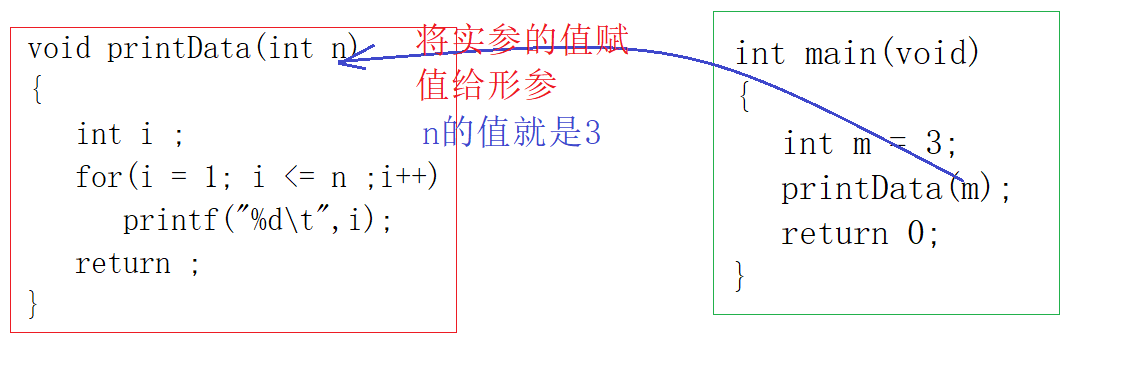
故：该函数的定义需要一个int类型的形式参数（参数名和变量名的命名规则一致），不需要返回值

参考代码如下（Demo）:

|  |
| --- |
| #include <stdio.h>  void printData(int n);//函数的声明  int main(void)  {  int m = 3;//m的值也可以从键盘输入  printData(m);//调用函数printData  return 0;  }  /\*函数名： printData  \*函数功能：输出1--n之间的数据（包括1 和 n）  \*函数参数：n (确定输出的范围)  \*函数返回值：无  \*/  void printData(int n)**//在执行这个函数的时候，n的值从哪来的**  {  int i ;  for(i = 1; i <= n ;i++)  printf("%d\t",i);  return ;  } |

**带参数的函数的执行过程（重点看实际参数和形式参数之间的传递）**

**在调用有参数的函数前，调用语句必须有实际参数（可以为变量，也可以为常量），在调用函数的瞬间，将实参的值赋值给形参。**



*练习1：定义一个函数，输出1--n之间偶数*

*练习2：定义一个函数，输出1--n之间奇数*

1. 定义一个函数，计算1\*2\*3\* ...\* n，并将结果返回给主调函数

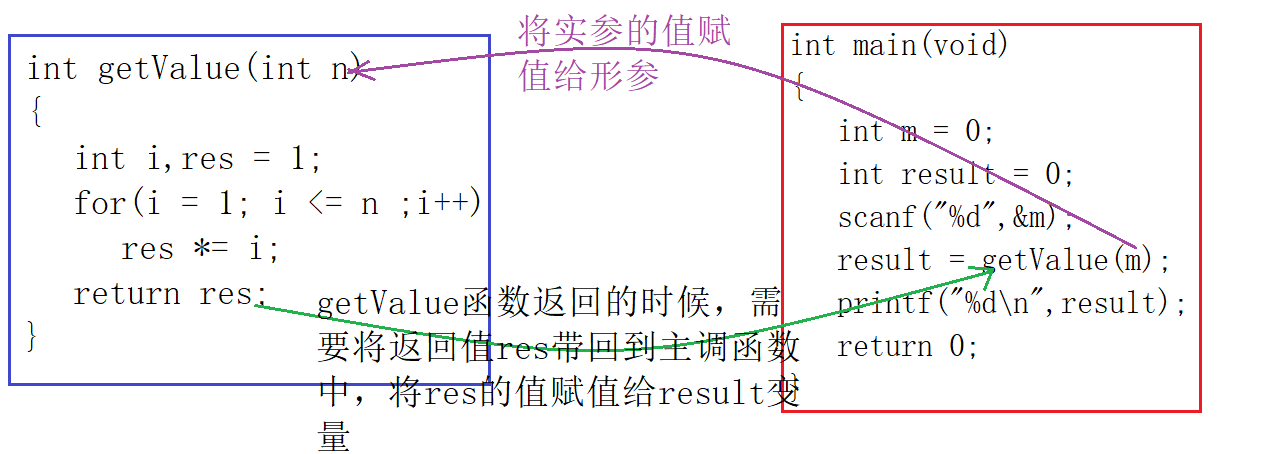
分析：该函数需要计算，需要将计算结果返回到主调函数，所以需要返回值，返回值类型int,也需要一个int类型的形参，函数要实现的功能是累乘

故：改函数的定义需要一个int类型的返回，需要一个int类型的返回值

参考代码如下（Demo）:

|  |
| --- |
| #include <stdio.h>  int getValue(int n);//函数的声明  int main(void)  {  int m = 0;  int result = 0;  scanf("%d",&m);  result = getValue(m);//调用函数getValue  printf("%d\n",result);  return 0;  }  /\*函数名： getValue  \*函数功能：计算n的阶乘  \*函数参数：n (确定输出的范围)  \*函数返回值：有  \*/  int getValue(int n)  {  int i,res = 1;  for(i = 1; i <= n ;i++)  res \*= i;  return res;  } |

**带参数的函数的执行过程（重点看返回值的返回过程）**



*练习：1.定义一个函数，计算1+2+3+ ...+ n，并将结果返回给主调函数*

*2.定义一个函数，计算1\*3\*7...\* n，并将结果返回给主调函数*

*3.*定义一个函数实现：

2x (x<1)

y={ 3x-2 (1=<x<5)

4x-3 (x>=10)

根据x求y的值。主函数实现：输入y的值，调用该函数后，输出对应的结果。

1. 定义一个函数，找出两个整数数据最大值，这两个数据由主函数从键盘获取，调用函数后并将结果返回给主调函数输出。

分析：该函数中，需要比较大小的两个数据由主函数从键盘获取，所以该函数需要两个int类型的形参，并且需要一个int类型的返回值

故：函数的定义需要两个形参，需要有返回值

参考代码如下（Demo）:

|  |
| --- |
| #include <stdio.h>  int findMax(int a,int b);  int main(void)  {  int a ,b ;  scanf("%d%d",&a,&b);  printf("最大值：%d\n", findMax(a,b));  return 0;  }  int findMax(int m,int n)  {  if(m > n)  return m;  else  return n;  } |

*练习：*

1. *定义一个函数，判断a是否可以被b整除，如果可以，返回1，否则返回0，m在主函数中，对函数返回值进行判断，如果返回值为1，输出可以整除，否则输出不可以整除。*
2. *定义一个函数实现：求两个整数的和,结果返回给主函数输出*
3. 定义一个函数，实现成绩等级划分，输出成绩等级。要求成绩在主函数中输入。（检查参数是否合法）

等级划分标准：

100 -- 90 优秀

89 -- 80 良好

79 -- 70 中等

69 -- 60 及格

60以下 不及格

分析：该函数需要一个形参（int），不需要返回值

参考代码如下（Demo）:

|  |
| --- |
| #include <stdio.h>  void fun(int s);  int main(void)  {  int score = 0;  scanf("%d",&score);  fun(score);  return 0;  }  void fun(int s)  {  if(s < 0 || s >100)//参数合法检测  {  printf("成绩不在可取范围内\n");  return ;  }  switch(s/10)  {  case 10:  case 9: printf("优秀\n");break;  case 8: printf("良好\n");break;  case 7: printf("中等\n");break;  case 6: printf("及格\n");break;  default: printf("不及格\n");break;  }  } |

六、函数的设计原则

**1．一个函数实现一个功能**

**2．返回值的原则：如果计算有结果需要加返回值**

**3．形参的设计及原则：如果计算的过程需要有数值进行辅助， 则需要加参数。**

**4．如果函数有形式参数则必须进行参数检查，检查参数是否有 异常。**

通常用分支结构（选择结构）进行参数检查，如果有错误则返回.

**5．函数体的设计思路**

第一步：定义变量

第二步：逻辑关系运用，选择结构还是循环结构还是综合运用。

七、课后作业

1.定义一个函数实现：100以内的奇数累加和。

2.定义一个函数实现：计算1的立方+2的立方+...+n的立方的和。

3.计算10 - 1/2 - 1/3 - ...- 1/n的值

4.编程实现一个计算器的功能：

======计算器======

=======1.加=======

=======2.减=======

=======3.乘=======

=======4.除=======

=======5.求余=====

=======0.退出=====

请输入0-5进行选择:

要求：能够循环的从键盘输入对应的数值后实现相应的功能，对应的功能要求分别定义成函数。

## 第七章 一维数组

**教学目的**

掌握一维数组的定义、存储、引用和初始化的基本语法，掌握数组作为函数参数的使用方法。

**教学要求**

1. 熟练掌握一维数组的定义、存储、引用和初始化的基本语法
2. 熟练掌握数组的应用（输入和输出）
3. 熟练掌握一维数组作为函数参数

**重点和难点**

1. 数组的应用（输入和输出）
2. 一维数组作为函数参数

**教学支撑环境与教学资源：**

1. 笔记本电脑
2. VC2010
3. 投影仪

**教学方法**

讲授法、练习法、讨论法

**授课课时**

6课时

**教学过程**

**---------------------------------AM---------------------------------**

一、课程引入

同时存储全班同学的C语言成绩需要重复定义多个类型相同的变量，代码重复率较高，可以采用定义数组的方式解决。

二、一维数组的定义

类型名 数组名1[EXP],数组名2[EXP],...;

demo

int c\_score[1000];//相当于定义了1000个int型变量

int mysql\_score[1000], ps\_score[1000];

注意：

1>数组的长度：EXP表示数组元素的个数即"数组的长度"

2>EXP为整型常量表达式，不能包含变量。

整型常量表达式：只能是字面值组合而成的表达式

int main(void){

int v = 100;

int c\_score[10 \* v];

return 0;

}

3>数组的长度和类型最好用宏定义的形式

#define ARR\_SIZE 1000

int main(void){

int c\_score[ARR\_SIZE];

return 0;

}

思考：这些数组元素的值是多少？

三、一维数组的初始化

数据类型 数组名[数组元素个数] = {第一个元素值，第二个元素的值,....};

demo

#define ARR\_SIZE 3

int main(void){

int c\_score[ARR\_SIZE] = {1, 2, 3};//全部赋值

int ps\_score[ARR\_SIZE] = {1};//部分赋值,未赋值的元素未0

int mysql\_score[] = {1,2 ,3 ,4 ,5};//省略长度赋值，长度5（根据初值的个数确定）

return 0;

}

注意：

1>初值的个数不能大于数组的长度 ，否则会产生运行时错误

int a[3] = { 1,2,3,4,5 };//错误，超出了数组长度

2>实际开发中经常使用部分赋值的方式进行

demo

#define ARR\_SIZE 5

int c\_score[ARR\_SIZE] = {0};

1. 一维数组的存储

数组可以用来存储多个数据,数组占用多个连续的内存空间,这些内存空间用来存储数据

demo

#define ARR\_SIZE 5

int a[ARR\_SIZE] = {2, 5, 10, 1, 3};//画出数组a的内存图

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a[0] | a[1] | a[2] | a[3] | a[4] |
| 2 | 5 | 10 | 1 | 3 |

注意：

1>保存数组所需字节单元的个数与数组的类型和长度有关

总字节数 = sizeof(数组类型)\*数组长度

int a[5];//数组a占据的字节数为4\*5=20字节

long long int b[2];//8\*2=16

2>数组长度的计算可以采用sizeof的方式完成

int main(void){

int c\_score[1000];

printf("长度=%d", sizeof(c\_score) / sizeof(int));

printf("长度=%d", sizeof(c\_score) / sizeof(c\_score[0]));

return 0;

}

**----------------------------------PM--------------------------------**

五．一维数组的引用

引用普通变量时，直接使用变量名；而引用数组元素时，使用数组名[下标]。

注意：如有数组int arr[N];则数组的下标为：0 ~ N-1，即最大的下标为数组长度-1。

#define ARR\_SIZE 5

int main(void)

{

int arr[ARR\_SIZE] = {0};

arr[2] = 10;//正确

a[5] = 3;//错误

return 0;

}

demo:

编写程序从键盘上接收10个整数存到一个一维数组中，之后在屏幕上逐个输出这10个数。

#include <stdio.h>

#define ARR\_SIZE 10

int main(void)

{

int a[ARR\_SIZE] = {0};

int i = 0;

for( i = 0; i < ARR\_SIZE ; i++)

{

printf("请输入第%d 个数:", i+1);

scanf(“%d”, &a[i]);

}

for(i = 0; i < ARR\_SIZE ; i++)

{

printf("第%d 个数是:%d\n", i + 1, a[i]);

}

return 0;

}

六、一维数组作为函数参数

1>数组作为函数参数可以理解为形参数组和实参数组共享一块内存空间，

形参数组的长度可以省略,编译器不会检查。

2>实际参数写数组名称。

#define ARR\_SIZE 10

void foo(int a[ ], int len){

a[1] = 10;

a[2] = 20;//形参数组改变，实参数组的值也会一起改变

}

int main(void){

int arr[ARR\_SIZE] = {0};

foo(arr,ARR\_SIZE);

printf("%d,%d\n", arr[1], arr[2]);

return 0;

}

课堂练习1：

1.编写函数实现求数组的平均值，其中数组作为函数的参数

#define ARR\_SIZE 10

float avg(int a[],int len);//返回平均值,len为数组中存放的元素个数

int main(void)

{

int arr[ARR\_SIZE];

int i = 0;

float aver = 0;

for(i=0;i<ARR\_SIZE;i++)

{

printf("请输入第%d个整数:",i+1);

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

}

aver = avg(arr,ARR\_SIZE);

printf("平均值是:%f\n",aver);

return 0;

}

float avg(int a[],int len)

{

int i = 0;

float average = 0;

for(i=0;i<len;i++)

{

average += a[i];//把数组中的所有数据加起来

}

average = average/len;//平均值=总值/总个数;

return average;

}

七、课后作业

1.定义2个函数，一维数组作为函数的参数，

函数1：把该一维数组的数组元素顺序输出

函数2：把该一维数组的数组元素逆序输出

要求：使用while循环语句或者for循环语句实现

2.定义一个函数，一维数组作为函数的参数， 把该一维数组的数组元素数据进行首尾对调，假设定义有int a[10];则把a[0]的数据与a[9]的数据进行互换，把a[1]的数据与a[8]的数据进行互换... 最后把对调后的数组顺序输出。

## 第八章 二维数组

**教学目的**

掌握二维数组的定义、内存存储图、二维数组的每个元素引用的方法和初化的基本语法，以及二维数组的遍历，二维数组元素下标的规律。

**教学要求**

1. 熟练掌握二维数组的定义、存储、引用和初始化的基本语法
2. 熟练掌握二维数组的数组元素的遍历及引用规则
3. 掌握二维数组元素的下标规律

**重点和难点**

1. 二维数组的定义、存储、引用和初始化
2. 二维数组的遍历

**教学支撑环境与教学资源：**

1. 笔记本电脑
2. VC2010
3. 投影仪

**教学方法**

讲授法、练习法、讨论法

**授课课时**

6课时

**教学过程**

1. 课程引入

同时存储全班同学的C语言成绩、Mysql成绩、PS成绩等需要定义多个类型相同的一维数组，代码重复率较高，可以采用定义二维数组的方式解决。

二、二维数组的定义

类型名 数组名1[EXP1][EXP2],数组名2[EXP1][EXP2],...;

int a[2][3];//画二维矩阵图

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 第0列 | 第1列 | 第2列 |
| 一维数组a[0] | 第0行 | a[0][0] | a[0][1] | a[0][2] |
| 一维数组a[1] | 第1行 | a[1][0] | a[1][1] | a[1][2] |

对于这个数组：我们可以将a[0][0],a[0][1],a[0][2]看成第一个一维数组，而将a[1][0],a[1][1],a[1][2]，看成第二个一维数组，则数组a就是这两个一维数组，数组名分别是a[0]和a[1]。

EXP1:行下标

EXP2:列下标

二维数组表示多个类型相同的数据元素的集合，等于同时定义多个变量，这些变量的名字：数组名[行下标][列下标]；

一个二维数组还可以理解为多个类型和长度相同的一维数组的集合。

int a[4][5];//相当于定义了4\*5=20个变量，相当于定义了4个一维数组

三、二维数组的存储

二维数组占用多个连续的内存空间（画图演示）。

int a[2][3];//在内存中的结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |
| a[0][0] | a[0][1] | a[0][2] | a[1][0] | a[1][1] | a[1][2] |

总字节数 = sizeof(数组类型)\*行数\*列数

int a[2][3];//数组a占据的字节数：4\*2\*3=24字节

计算元素a[i][j]在数组中位置的公式(从0开始计数)：i\*m+j（m为数组的列数)，如：a[1][2]是在数组a中的第1\*3+2=5位

四、二维数组的引用

方法同一维数组：区别就是它使用的双下标。

注意：下标从0开始，控制下标越界问题。

int a[2][5];

a[0][0]=10;

a[1][5]=20;//错误，数组越界了

五、二维数组的初始化

1.按行全部赋初值

int a[2][3]={{10,20,30},{40,50,60}};

|  |  |  |  |
| --- | --- | --- | --- |
| 行  列 | 0 | 1 | 2 |
| 0 | 10 | 20 | 30 |
| 1 | 40 | 50 | 60 |

2.按存储顺序全部赋初值

int a[2][3]={100,200,300,400,500,600};

|  |  |  |  |
| --- | --- | --- | --- |
| 行  列 | 0 | 1 | 2 |
| 0 | 100 | 200 | 300 |
| 1 | 400 | 500 | 600 |

3.部分赋初值（按存储顺序）

int a[2][3]={10,20,40}; //a[0][0]=10,a[0][1]=20,a[0][2]=40,

|  |  |  |  |
| --- | --- | --- | --- |
| 行  列 | 0 | 1 | 2 |
| 0 | 10 | 20 | 40 |
| 1 | 0 | 0 | 0 |

4.部分赋初值（按行部分赋值）

int x[2][3]={{10,20},{40}};//a[0][0]=10,a[0][1]=20,a[0][2]=0,a[1][0]=40

|  |  |  |  |
| --- | --- | --- | --- |
| 行  列 | 0 | 1 | 2 |
| 0 | 10 | 20 | 0 |
| 1 | 40 | 0 | 0 |

注意：使用部分赋初值时，**未赋值的元素均为零**，如果数组中零的个数很多，则用此法。在VC2010中可以同时有按行赋值和按存储顺序赋值。

int a[2][3]={{1,2},4,5,6};//正确，第0行为{1,2,0}，第1行为{4,5,6}。

5.省略行数赋初值：定义时可省略行数！

int a[ ][3]={{1,2,3},{4,5,6},{7,0,0}};

int b[ ][4]={{0,0,3},{0},{0,10}};

注意：行初值括号里必须有值。

int x[ ][3]={10,20,30,40,50,60,70}; //注意：不能省略列数。

|  |  |  |  |
| --- | --- | --- | --- |
| 行  列 | 0 | 1 | 2 |
| 0 | 10 | 20 | 30 |
| 1 | 40 | 50 | 60 |
| 2 | 70 | 0 | 0 |

int a[4][];//错误

六、二维数组的遍历方法

注意两层循环的范围和起始值。

demo:输出a[3][4]所有的数组元素的值（按行输出）

#define M 3

#define N 4

int main(void)

{

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

int,i,j;

for(i = 0; i < M; i++)

{

for(j = 0; j < N; j++)

{

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

}

printf("\n");

}

}

思考： 可以将内层循环的中的j = 0省略不写吗？如果省略会出现什么问题？自己修改代码得出总结！

课堂练习：

1.输出a[3][4]所有的数组元素的值（按列输出）

七、最值问题

demo:对3\*4列的矩阵求其最大值，以及其所在的行号和列号

#define M 3

#define N 4

int main(void)

{

int i,j,max;

int row = 0, colum = 0;

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

max=a[0][0];//假设第0个元素为最大值

for(i=0; i<M; i++)

{

for(j=0; j<N; j++)

{

if(a[i][j] > max)

{

max=a[i][j];

row=i;

colum=j;

}

}

}

printf(“max=%d,row=%d,colum=%d\n”, max, row, colum);

return 0;

}

课堂练习：

1.对3\*4的二维数组求其最小值，以及其所在的行号和列号

2.对3\*4的二维数组求每行的最大值，以及其所在的行号和列号

3.对3\*4的二维数组求每列的最小值，以及其所在的行号和列号

八、二维数组作为函数参数

1.二维数组名作为参数传递

demo1:定义一个输出二维数组所有元素的函数

#define M 3

#define N 4

void output(int a[][N], int row, int column);

int main(void)

{

int b[M][N] = { { 1, 4, 7, 9 }, { 2, 5, 8, 10 }, { 3, 6, 9, 12 } };

output(b,M,N);

}

void output(int a[][N],int row,int column)

{

int i = 0,j = 0;

for (i = 0; i < row; i++)

{

for (j = 0; j < column; j++)

{

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

}

printf("\n");

}

}

output函数的函数头中的二维数组可以有以下两种形式：

1. output(int a[][N])
2. output(int a[M][N])

注意：

数组名传递给函数的是一个地址值，在函数中引用的将是实参的数组元素，编译器只为形参开辟一个存放地址的存储单元。

课堂练习：

1.定义函数实现：对3\*4的二维数组求其最大值

2.定义函数实现：对3\*4的二维数组求其平均值

九、课后作业

1.编写函数，求3\*3二维数组中所有的奇数之和。

2.编写程序，求一个4\*5的二维数组每行元素的平均值，该二维数组的数据从键盘上输入。

## 第九章 结构体

**教学目的**

掌握结构体的定义、存储、引用和初始化的基本语法，学会引用结构体中的每一个成员变量。掌握结构体数组的定义、存储、引用和初始化的基本语法。

**教学要求**

1. 熟练掌握结构体的定义、存储、引用和初始化的基本语法
2. 学会引用结构体中的每一个成员变量
3. 熟练掌握使用typedef定义结构体
4. 熟练掌握结构体数组的定义、存储、引用和初始化的基本语法

**重点和难点**

1. 结构体的应用
2. 结构体成员变量的引用
3. 结构体数组的应用

**教学支撑环境与教学资源：**

1. 笔记本电脑
2. VC2010
3. 投影仪

**教学方法**

讲授法、练习法、讨论法

**授课课时**

6课时

**教学过程**

**---------------------------------AM---------------------------------**

一、课程引入

同时存储全班同学的C语言成绩、PS成绩等需要定义多个类型不同的变量，可以采用定义结构体的方式解决。

二、结构体的定义

C语言中，可以自己构造出自己想要的数据类型，这种类型叫做复合数据类型。

复合数据类型------结构体--是由一系列具有相同类型或不同类型的数据构成的数据集合

结构体：不同的数据类型组成的组合型的数据结构

1.结构体的定义

struct 结构体名

{

成员列表

};

struct student

{

int num;

int age;

float score;

double avg;

};

说明：

1>定义了一个结构体类型，结构体的关键字---struct，结构体名为student；

2>这个结构体包含了多个不同的基本数据类型。

注意：

1>结构体的类型可以多种，名字由关键字struct和结构体名组合而成

2>访问成员的规则：

（1）结构体普通变量通过"."来访问成员，例如：pe.id

（2）结构体指针变量通过"->"来访问成员, 例如：p->id（大致提一下）

三、定义结构体变量

1.先声明，再定义

struct 结构体名

{

数据类型 成员变量名;

数据类型 成员变量名;

数据类型 成员变量名;

...;

};

语法： struct 结构体名 变量名;

例如:

struct student//student为结构体名

{

int num;

int c\_score;

int ps\_score;

float avg;

};//定义结构体

int main(void)

{

struct student Tom;//Tom为结构体变量

Tom.num = 10;

printf("序号:%d\n",Tom.num);

return 0;

}

2.在声明类型的同时，定义变量

struct 结构体名

{

数据类型 成员变量名;

数据类型 成员变量名;

数据类型 成员变量名;

...;

}变量名列表;

例如：

struct people

{

int age;

int id;

}p1; //p1 是一个结构体变量

这里的p1, 相当于struct people p1;

p1.age = 20;

p1.id = 15;

3.省略结构体名，直接定义结构体类型的变量

struct

{

数据类型 成员变量名;

...;

}变量名列表;

例如：

struct

{

int age;;

int id

}p1;

特点：只能定义一次，接下来不能再定义结构体变量了。

课堂练习1：

1.定义结构体，包含学生的信息(序号、C语言成绩、PS成绩、平均分)，定义结构体变量并赋初值，输出学生的信息。

struct student//student为结构体名

{

int num;

int c\_score;

int ps\_score;

float avg;

};//定义结构体

int main(void)

{

struct student Tom = {15,90,95,92.5};

printf("我的序号是:%d\n",Tom.num);

printf("C语言成绩是:%d\n",Tom.c\_score);

printf("PS成绩是:%d\n",Tom.ps\_score);

printf("平均分是:%f\n",Tom.avg);

return 0;

}

2.定义结构体，包含学生的信息(序号、3门课的成绩、平均分)，定义结构体变量并赋初值，输出学生的信息。

struct student//student为结构体名

{

int num;

int score[3];

float avg;

};//定义结构体

int main(void)

{

struct student Tom = {15,{90,95,80},83.3};

int i = 0;

printf("我的序号是:%d\n",Tom.num);

printf("成绩分别是:%d,%d,%d\n",Tom.score[0],Tom.score[1],Tom.score[2]);

printf("平均分是:%f\n",Tom.avg);

return 0;

}

四、typedef关键字

功能：它不是用来创造类型，而是给已有的类型取一个别名。

定义变量

int a;

int a[10];

struct student stu;

规则：将名字去掉，剩下的就是类型

int

int [10]

struct student

typedef 已有的类型 别名;//注意：别名后面的分号;不能省略

例如： typedef int TYPE;======>TYPE与int 等价,如TYPE a;相当于int a;

typedef struct student

{

int num;

}STU1,STU2;//STU1和STU2都是别名

STU1 <===>struct student

demo:

typedef struct student

{

int num;

int age;

}STU;//STU是struct student的别名

int main(void)

{

struct student Jim;//定义别名后，原先的结构体名称也可以用

STU Tom;//相当于struct student Tom;

Tom.age = 18;

Jim.age = 20;

return 0;

}

课堂练习2：

1.定义结构体（使用typedef），包含学生的信息(序号、三门课的成绩、平均分)，定义结构体变量并赋初值，输出学生的信息。

typedef struct student//student为结构体名

{

int num;

int score[3];

float avg;

}STU;//STU是结构体别名

int main(void)

{

//相当于struct student Tom

STU Tom = {15,{90,95,80},83.3};

int i = 0;

printf("我的序号是:%d\n",Tom.num);

for(i=0;i<3;i++)

{

printf("第%d门的成绩:%d\n",i+1,Tom.score[i]);

}

printf("平均分是:%f\n",Tom.avg);

return 0;

}

2.定义结构体（使用typedef），包含学生的信息(序号、3门课的成绩、平均分)，从键盘输入学生的信息，计算平均分，并出这个学生的信息。

#define SIZE 3

typedef struct student//student为结构体名

{

int num;

int score[SIZE];

float avg;

}STU;//STU是结构体别名

int main(void)

{

//相当于struct student Tom

STU Tom;

int i = 0;

Tom.avg = 0;

printf("请输入序号:");

scanf("%d",&Tom.num);

for(i=0;i<SIZE;i++)

{

printf("请输入第%d门的成绩:",i+1);

scanf("%d",&Tom.score[i]);

Tom.avg += Tom.score[i];//求累加和

}

Tom.avg = Tom.avg / SIZE;//求平均分

printf("我的序号是:%d\n",Tom.num);

for(i=0;i<3;i++)

{

printf("第%d门的成绩:%d\n",i+1,Tom.score[i]);

}

printf("平均分是:%f\n",Tom.avg);

return 0;

}

**---------------------------------PM---------------------------------**

五、结构体数组

一个学生的信息我们知道怎么去存储了----使用结构体

那么全班50名学生的信息，我们怎么去存呢？ -----使用结构体数组

1.定义结构体数组的一般形式：

1>声明结构体的时候就定义结构体数组：

struct 结构体名

{

成员列表

} 数组名[数组长度];

2>先声明一个结构体类型（如struct student ), 然后再用此类型定义结构体数组;

结构体类型 数组名[数组长度]；

demo：

struct student

{

int num;

int c\_score;

int ps\_score;

float avg;

};

struct student class3[5];//class3 是结构体数组名

2.结构体数组的初始化：

结构体类型 数组名[数组长度] = { 初值表列 };

struct student class2[3] = {{10,90,95,92.5},{11,80,90,85.0},{12,100,90,95.5}};

课堂练习3：

1.有n个学生的信息（包括序号、成绩）,找出成绩的最高的学生，并输出该学生的信息

#include<stdio.h>

#define N 5

typedef struct student

{

int num;

int score;

}STU;

int main(void)

{

STU class3[N] = {{10,90},{14,70},{8,95}};

int i = 0;

int max = 0;//存放成绩最高所在的结构体数组元素下标

for(i=1;i<N;i++)

{

if(class3[i].score > class3[max].score)

max = i;

}

printf("成绩最高的学生:\n");

printf("序号:%d\n",class3[max].num);

printf("成绩:%d\n",class3[max].score);

return 0;

}

六、课后作业

1.定义函数：把10名学生的学号、成绩以及平均分放在一个结构体中，学生的学号、成绩由键盘输入，然后计算出平均分。