蜂考系统课

《C 语言》

课时一　程序设计与 C 语言

考点	重要程度	分值	常见题型
1. 计算机语言与编译	★★★★	0～3	选择、填空、判断
2. 数位及其表示、数制之间的转换	★★	0～2	
3. C 语言程序的结构	★★★★★	0～4	
4. 编译预处理	★★★★★	0～4	
5. 算法	★★	0～2	
6. 三种基本结构	★★★	0～3	

1. 计算机语言与编译

机器语言：一串仅由 0 和 1 序列表示的语言。计算机只能识别和接受 0 和 1 组成的指令。

符号语言（汇编语言）：用一些英文字母和数字表示一个指令。

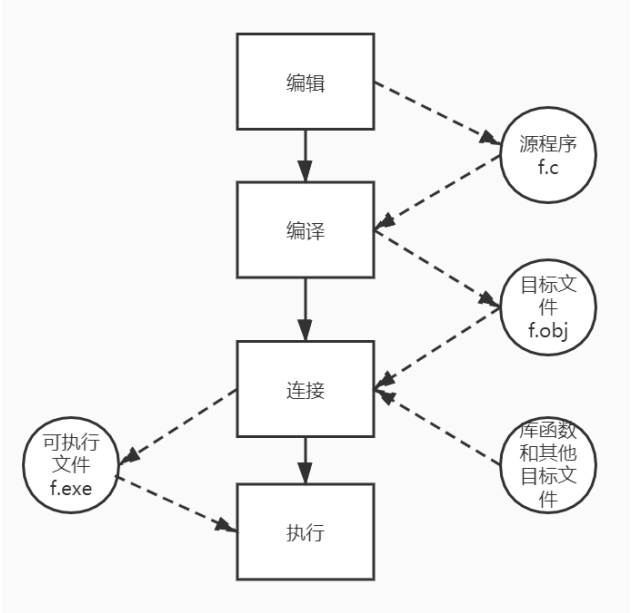
符号语言（汇编语言）——>机器语言

低级语言：完全依赖于具体机器特性的，是面向机器的语言。例：机器语言，汇编语言等。

高级语言：语言功能很强，且不依赖于具体的机器，用它写出的程序对任何型号的计算机都适用（或只须作很少的修改），同时它与具体的机器距离较远。

C 语言是面向过程的语言。

高级语言——>机器语言



题 1.下列关于 C 程序的运行流程描述，（　　）是正确的。

A.编辑目标程序、编译目标程序、连接源程序、运行可执行程序

B.编译源程序、编辑源程序、连接目标程序、运行可执行程序

C.编辑源程序、编译源程序、连接目标程序、运行可执行程序

D.编辑目标程序、编译源程序、连接目标程序、运行可执行文件

答案：C

题 2.C 语言源程序文件的后缀是_____，经过编译后生成文件的后缀是_____，经过连接后生成的文件后缀是_____。

答案：　.c　.obj　.exe

2.数位及其表示、数制之间的转换

	十进制	0 1 2 3 4 5 6 7 8 9
表示：0b	二进制	0 1
表示：0	八进制	0 1 2 3 4 5 6 7
表示：0x	十六进制	0 1 2 3 4 5 6 7 8 9 A B C D E F

题 1.095 表示一个八进制整数。（判断题）

答案：错误

3.C 语言程序的结构

（1）函数是 C 程序的基本单位。

（2）一个 C 语言程序是由一个或多个函数组成的，其中必须包含一个 main 函数，而且有且仅有一个 main 函数，main 函数=主函数。

（3）程序总是从 main 函数开始执行，而不论 main 函数在整个程序中的位置，同时也是在 main 函数的最后一条语句 结束执行。

（4）在每个数据声明和语句的最后必须有一个分号，分号表示语句的结束。

（5）C 语言本身不提供输入输出语句。输入和输出的操作是由库函数 scanf 和 printf 等函数来完成的。

（6）英文输入法输入。

```
#include<stdio.h>
int main(void)
{

    return 0;
}
```

题 1.C 语言程序总是从 main 函数开始执行，那么相对于其他函数，main 函数在程序中的位置为（　　）。

A.必须在程序开头

B.必须在其他函数之前

C.必须在其他函数之后

D.任何位置

答案：D

题 2.以下叙述不正确的是（　　）。

A.一个 C 源程序可由一个或多个函数组成。

B.一个 C 源程序必须包含一个 main 函数。

C.C 程序的基本组成单位是语句。

D.在 C 程序中，注释说明通常位于一条语句的后面。

答案：C。 C 程序的基本组成单位是函数。

题 3.下列对 C 语言源程序执行过程中描述正确的是（　　）。

A.从 main 函数开始执行，在 main 函数中结束执行。

B.从程序的第一个函数开始执行，到最后一个函数结束执行。

C.从 main 函数开始执行，到源程序最后一个函数结束执行。

D.从第一个函数开始执行，到 main 函数结束执行。

答案：A

4.编译预处理

（1）编译预处理以#开头。

（2）编译预处理包括文件包含，宏定义和条件编译。

（3）文件包含：#include

文件包含是可以嵌套的。

#include<stdio.h>

（4）宏定义：　#define 名称 内容

题 1.程序中的命名行：#define PI 3.14 在（　　）被处理。

A.编辑时　　B.程序运行时

C.编译前　　D.编译时

答案：C。 宏定义作为编译预处理命令的一种，在预编译阶段被处理。

题 2.有如下的宏定义，则下列语句的结果是（　　）。

```
#define MAX 20
#define NUM MAX+10
X=MAX*NUM
```

A.600　　B.500　　C.410　　D.50

答案：C。X=20*20+10

题 3.文件包含是不能嵌套的，即在一个被包含文件中不能包含另一个被包含文件。（判断题）

答案：错误

5.算法

（1）程序=算法+数据结构

算法：对操作的描述，即要求计算机进行操作的步骤。

数据结构：在程序中要指定用到哪些数据以及这些数据的类型和数据的组织形式。

（2）算法的特性

①有穷性。

②确定性。

③有零个或多个输入。

④有一个或多个输出。

⑤有效性。

（3）描述或者表述算法的方法：自然语言，流程图，伪代码。

题 1.著名计算机科学家 NikiklausWirth 提出一个公式：程序=（　　）。

A.数据+运算符　　B.数据结构+算法

C.结构+函数　　D.运算符+运算数

答案：B

题 2.下列（　　）不是通常算法的表示方式。

A.自然语言　　B.递推法

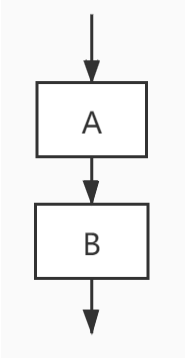
C.伪代码　　D.流程图

答案：B

6.三种基本结构

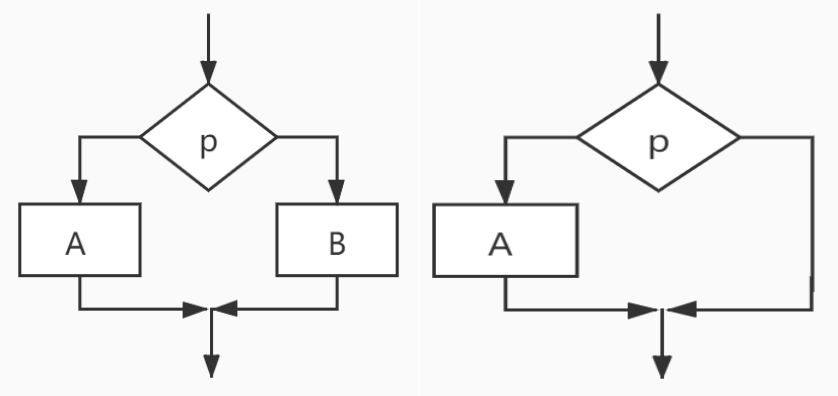
（1）顺序结构

在执行完 A 框所指定的操作后，接着执行 B 框所指定的操作。



（2）选择结构（分支结构）

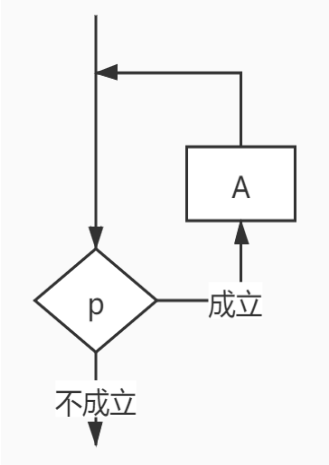
根据给定的条件 p 是否成立而选择执行 A 框或 B 框。无论条件 p 是否成立，只能执行 A 框或 B 框之一。A 或 B 两个框中可以有一个是空的，即不执行任何操作。



（3）循环结构（重复操作）

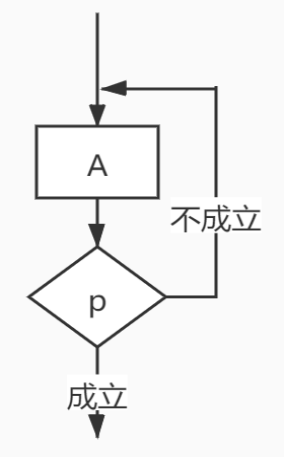
①当型（while 型）循环结构

当给定的条件 p 成立时，执行 A 框操作，执行完 A 后，再判断 p 是否成立，如果仍然成立，再执行 A 框，如此反复执行 A 框，直到某一次 p 条件不成立为止，此时不执行 A 框，而脱离循环结构。



②直到型（until 型）循环结构

先执行 A 框，然后判断给定的 p 条件是否成立，如果 p 条件不成立，则再执行 A，然后再对 p 条件做判断，如此反复执行 A，直到给定的 p 条件成立为止，此时不再执行 A，脱离循环结构。



以上 3 种基本结构，有以下共同特点：

① 只有一个入口。

② 只有一个出口。一个判断框有两个出口，而一个选择结构只有一个出口。

③ 结构内的每一部分都有机会被执行到。

④ 结构内不存在“死循环”（无终止的循环）。

题 1.结构化程序所要求的的基本结构不包括（　　）。

A.顺序结构　　B.GOTO 结构

C.选择（分支）结构　　D.重复（循环）结构

答案：B

题 2.下列关于 C 语言的说法错误的是（　　）。

A.C 程序的工作过程是编辑、编译、连接、运行。

B.C 语言不区分大小写。

C.C 程序的三种基本结构是顺序结构、选择结构、循环结构。

D.C 程序总是从 main 函数开始执行。

答案：B

课时一 练习题

1.用 C 语言编写的代码程序（　　）。

A.可立即执行

B.是一个源程序

C.经过编译后即可执行

D.经过编译解释才可执行

2.C 语言源程序经过编译之后，生成一个.obj 文件。（判断题）

3.058 表示一个八进制整数。（判断题）

4.C 语言程序是由（　　）构成的。

A.语句　　B.程序行

C.函数　　D.字符

5.一个可执行 C 语言程序的结束执行点是 main 函数的最后一条语句。（判断题）

6.下列叙述中正确的是（　　）。

A.C 语言程序将从源程序中第一个函数开始执行。

B.可以在程序中由用户任意指定一个函数作为主函数，程序将从此函数开始执行。

C.C 语言规定必须用 main 作为主函数名，程序将从此函数开始执行，在主函数结束执行。

D.main 函数可以作为用户标识符，用以命名任意一个函数作为主函数。

7.若程序中有定义：#define N 100，则以下叙述中正确的是（　　）。

A.定义行时，定义了标识符 N 的值为整数 100。

B.在编译系统对 C 源程序进行预处理时，用 100 替换标识符 N。

C.在对源程序进行编译时，用 100 替换标识符 N。

D.在运行时，用 100 替换标识符 N。

8.C 语言提供的编译预处理命令包括_____、文件包含、_____。

9.设有宏定义命名如下：#define RES 30-5，则表达式 RES*5+30 的值为（　　）。

A.60　　B.35　　C.45　　D.25

10.算法是解决问题的方法和步骤。（判断题）

11.若 C 程序中出现死循环则违背了算法的（　　）。

A.有穷性　　B.确定性

C.有效性　　D.有一个或多个输出

12.C 程序设计的三种基本结构是（　　）。

A.嵌套、递归、循环

B.顺序、选择、循环

C.递归、选择、循环

D.循环、顺序、转移

课时二 顺序结构程序设计（1）

考点	重要程度	分值	题型
1.常量和变量	★★★★★	0～6	选择题、编程题
2.标识符	必考	2～4	选择题、填空题
3.数据类型	★★★★★	0～6	选择题
4.运算符和表达式	必考	4～6	选择题、填空题

1.常量和变量

1.1 常量：在程序运行过程中，其值不能被改变的量。

常量
- 整型常量
- 实型常量
- 字符常量
  - 普通字符
  - 转义字符
- 字符串常量

①整型常量

②实型常量：

十进制小数形式。

十进制指数形式。例：12.34e3（$12.34*10^3$）

注：e 或 E 之前必须有数字，且 e 或 E 后面必须为整数。

③字符常量：字符常量只能是一个字符，不包括单引号。

普通字符：单引号括起来的一个字符。

注意：字符常量只能是一个字符，不包括单引号。字符常量存储在计算机存储单元中时，并不是存储字符，而是以其代码（一般采用 ASCII 代码）存储的，例如字符‘a’的 ASCII 代码是 97，因此，在存储单元中存放的是 97。

ASCII 码：字符编号。

‘a’:97　　‘A’:65

‘b’:98　　‘B’:66

‘c’:99　　‘C’:67

‘b’=‘a’+1

‘A’=‘a’-32

注意：A-Z，a-z，0-9 是依次增加的。

转义字符：以字符\开头的字符序列。将“\”后面的字符转换成其它意义。

转义字符	字符值
\’	一个单引号’
\”	一个双引号”
\\	一个反斜线\
\n	换行
\r	回车
\t	空格，跳格
\o,\oo,\ooo(o 代表一个八进制数字)	八进制数表示的字符
\xh(h 代表一个十六进制数字)	十六进制数表示的字符

④字符串常量：用双引号把若干个字符括起来，字符串常量是双引号中的全部字符，但不包括双引号。

⑤符号常量：#define 指令，指定一个符号名代表一个常量。这样用一个符号名代表一个常量的，称为符号常量。

1.2 变量

变量代表一个有名字的，具有特定属性的存储单元。它用来存放数据，也就是存放变量的值。在程序运行期间，变量的值是可以改变的。

变量必须先定义，后使用。

题 1.在 C 语言中，char 字符型数据在内存中的存储形式是（　　）。

A.补码

B.反码

C.ASCII 码

D.原码

答案：C

题 2.（　　）是正确的字符常量。

A.“\n”

B.‘12’

C.“a”

D.‘\101’

答案：D

2.标识符

（1）定义：用来对变量、符号常量名、函数、数组、类型等命名的有效字符序列。

（2）规则：①标识符只能由数字、字母和下划线组成。

②第一个字符必须是字母或者下划线。

③标识符区分大小写。

④关键字不能作为标识符。

auto	break	case	char	const
continue	default	do	double	else
enum	extern	float	for	goto
if	inline	int	long	register
restrict	return	short	signed	sizeof
static	struct	switch	typedef	union
unsigned	void	volatile	while	

注：main，define，scanf，printf 均不是关键字。

题 1.以下选项中，不能用作变量名字的是（　　）。

A.abc.c　　B.file

C.Main　　D.PRINT

答案：A。标识符只能由数字、字母和下划线组成。

题 2. 下面不是 C 语言关键字的是（　　）。

A. Do　　B. void

C. int　　D. unsigned

答案：A

题 3. 下列 4 组选项中，均是合法的用户标识符的是（　　）。

A. sd　P_0　do　　B. b_a　get　Int

C. float　Lo4　-A　　D. -123　yu　2e5

答案：B。A 选项 do 是一个关键字。C 选项 float 是一个关键字，-A 中有-号。D 选项-123 中也有-号，2e5 以数字开头。

3. 数据类型

- 数据类型
  - 整型类型
    - 基本整型(*int*)　2个字节或者4个字节
    - 短整型(*short*)　2个字节
    - 长整型(*long*)　4个字节
  - 浮点类型
    - 单精度浮点型(*float*)　4个字节
    - 双精度浮点型(*double*)　8个字节
  - 字符型(*char*)　1个字节

C 标准只要求 long 型数据长度不短于 int 型，short 型不长于 int 型。即

$$sizeof(short) \leqslant sizeof(int) \leqslant sizeof(long)$$

整型在存储单元中的存储方式是：用整数的补码形式存放。

①一个正数的补码是该数的二进制形式。

②求一个负数的补码，先将此数的绝对值写成二进制形式，然后对其后面所有各二进制数按位取反，再加一。如-5 的补码。

5 的原码	0000 0101
按位取反	1111 1010
-5 的补码	1111 1011

实数是以指数形式存放在存储单元的。

题 1.C 语言中，整型数据在内存中的存储形式是（　　）。

A.ASCII 码　　B.原码　　C.反码　　D.补码

答案：D

题 2. -3 对应的八位二进制补码是_____。

答案：1111 1101

4.运算符和表达式

4.1 基本的算术运算符

+　　-　　*　　/　　%

说明：

①两个整数的相除的结果是整数。

②%运算符要求参加运算的运算对象为整数，结果也是整数。

③除%以外的运算符的操作数都可以是任何算术类型。

题 1.若 x 是整型变量，表达式(x=5/2)的值是（　　）。

A.2.5　　B.2.0

C.3　　D.2

答案：D

题 2.表达式(ab)/(cd)在 C 语言中的正确表示是（　　）。

A.a*b/c*d　　B.a/c/d/b

C.a*b/c/d　　D.a/d*b*c

答案：C

题 3.在 C 语言中，要求运算的数必须是整数的运算符是（　　）。

A./　　B.!

C.%　　D.==

答案：C

4.2 自增、自减运算符：作用是使变量的值加 1 或减 1。

++i,--i　　在使用 i 之前，先使 i 的值加（减）1

i++,i--　　在使用 i 之后，再使 i 的值加（减）1

注意：自增，自减运算符只能用于变量，不能用于常量和表达式。

题 1.设有 int x=11,则表达式（x++*1/3）的值是（　　）。

A.3　　B.4

C.11　　D.12

答案：A

4.3 算术表达式和运算符的优先级与结合性

用算术运算符和括号将运算对象连接起来的、符合 C 语法规则的式子，称为 C 算术表达式。运算对象包括常量、变量、函数等。

C 语言规定求表达式时，先按运算符的优先级别顺序进行，若一个运算对象两侧的运算符的优先级别相同时，则按规定的“结合方向”处理。

“左结合性”：结合方向自左向右，即运算对象先与左边的运算符结合。例：算术运算符。

“右结合性”：结合方向自右向左，即运算对象先与右边的运算符结合。例：赋值运算符。

4.4 不同类型之间的混合运算

如果一个运算符的两侧的数据类型不同，则先自动进行类型转换，使二者具有同一种类型，然后进行运算。因此整型、实型、字符型数据间可以进行混合运算。

规律为：

①+、-、*、/运算的两个数中有一个数为 float 或 double 型，结果为 double 型，因为系统将所有 float 型数据都先转换为 double 型，然后进行运算。

②如果 int 型与 float 或 double 型数据进行运算，先把 int 型和 float 型数据转换为 double 型，然后进行运算，结果是 double 型。

③字符型数据与整型数据进行运算，就是把字符的 ASCII 代码与整型数据进行运算。

题 1.若有以下四个变量的定义：char a;int b;float c;double d;，则表达式“a*b-d+c”的类型为_____。

答案：double

4.5 强制类型转换运算符：将一个表达式转换成所需类型。

其一般形式为　**(类型名)(表达式)**

注意：在进行强制类型转换时，得到一个所需类型的中间变量，而原来变量的类型并未发生变化。

题 1.设 int n;float f=13.8;则执行 n=(int)f%3 后，n 的值为（　　）。

A.4　　B.1

C.4.33　　D.4.5

答案：B

题 2.令 int m=11;则(float)m/2 的数值为_____。

答案：5.5

4.6 C 运算符

①算术运算符	+ - * / % ++ --	
②关系运算符	> < == >= <= !=	
③逻辑运算符	! && ||	
④赋值运算符	=	
⑤条件运算符	? :	例：a>b?c:d
⑥逗号运算符	,	例：r=(a,b,c)
⑦指针运算符	* &	
⑧求字节数运算符	sizeof	
⑨强制类型转换运算符	（类型）	

题 1.表达式 x=(a=3,b=2,b*a)执行结束后，a 的值为_____，x 的值为_____。

答案：3,6

课时二 练习题

1.在 C 语言中，不正确的浮点数常量是（　　）。

A.-.123　　B.-1123e-3.0

C.-1.23e-1　　D.-0.123

2.下面四个选项中，不是合法转义字符的选项是（　　）。

A.‘\”’　　B.‘\\’

C.‘\058’　　D.‘\xd’

3.C 语言中基本数据类型包括（　　）。

A.整型、实型、逻辑型

B.整型、实型、字符型

C.整型、字符型、逻辑型

D.整型、实型、逻辑型、字符型

4.（　　）把 x,y 定义成 float 类型变量，并赋同一初值 3.14。

A.float x,y=3.14;　　B.float x,y=2*3.14;

C.float x=3.14,y=3.14;　　D.float x=y=3.14;

5.下面 C 语言标识符中，不合法的标识符的是（　　）。

A.student　　B.x_1

C._12xy　　D.float

6.在 C 语言中，自定义标识符 St 和 st 是等价的。（判断题）

7.下列 4 组选项中，均不是 C 语言关键字的选项是（　　）。

A.Define　IF　type　　B.getc　char　printf

C.include　scanf　case　　D.while　pow　go

8.已定义变量 a，sizeof(a)可以求出变量 a 在内存中所占的字节数。（判断题）

9.-1 的八位二进制补码是（　　）。

A.1000 0001　　B.0000 0001

C.1111 1111　　D.1111 1110

10.在 C 语言中程序中，要求运算数必须是整型的运算符是（　　）。

A./　　B.++

C.%　　D.*

11.表达式 9%45+15/6*2 的值是_____。

12.若 x 是 double 类型变量，则表达式(x=10/3)的值为 3.333333。（判断题）

13.下列语句段执行后 x 的值是（　　）。

```
int x=5,y=8;
x=x*y;
y=x/y;
x=x/y;
```

A.5　　B.8

C.40　　D.0

14.C 语言语句中，设定 x=5，则执行 a=x++后的值为（　　）。

A.a=5,x=5　　B.a=5,x=6

C.a=6,x=5　　D.a=6,x=6

15.5++是一个正确的表达式。（判断题）

16.设有声明“char w;int x;double y;”，则表达式w-x+y值的数据类型为（　　）。

A.char　　B.int

C.float　　D.double

17.若有定义：int a=7;float x=3.5,y=4.7;则表达式x+a%3*(int)(x+y)%2/4的值为（　　）。

A.3.500000　　B.2.500000

C.2.750000　　D.2.000000

18.设有语句“int m=7;”，则(float)(m/2)与(float)m/2的分别为（　　）。

A.3与3.5　　B.3与3

C.3.5与3　　D.3.5与3.5

19.设所有变量均为int型，则表达式(a=2,b=5,b++,a+b)的值为（　　）。

A.7　　B.8

C.6　　D.2

课时三 顺序结构程序设计（2）

考点	重要程度	分值	题型
5.C 语句	★★★	0～6	编程题
6.输入输出语句	必考	3～6	选择题、填空题、编程题

5.C 语句

5.1 C 语句的作用和分类

（1）控制语句。控制语句用于完成一定的控制功能。

①if()……else……　　条件语句

②for()……　　循环语句

③while()……　　循环语句

④do……while()　　循环语句

⑤continue　　结束本次循环语句

⑥break　　中止执行 switch 或循环语句

⑦switch　　多分支选择语句

⑧return　　从函数返回语句

以上语句中()表示括号中是一个“判别条件”，“……”表示内嵌的语句。

（2）函数调用语句。函数调用语句由一个函数调用加一个分号构成。

（3）表达式语句。表达式语句由一个表达式加一个分号构成。

（4）空语句。可以用来作为流程的转向点，也可用来作为循环语句中的循环体。

（5）复合语句。可以用{ }把一些语句和声明括起来称为复合语句。

注意：复合语句中最后一个语句中最后的分号不能忽略不写。

5.2 赋值语句

（1）赋值运算符

赋值符号 = 就是赋值运算符，它的作用是将一个数据赋给一个变量，也可以将一个表达式的值赋给一个变量。变量必须先定义，后使用。

```
int x;                  ①定义：数据类型 变量名;
float y;
x=1;                    ②赋值：变量名=常量;
int a=2;                ③初始化：数据类型 变量名=常量;
int b=3,c=4;
```

试问：double m=n=1.0; 这条语句的初始化是否正确？

（2）复合的赋值表达式

在赋值符 = 之前加上其他运算符，可以构成复合的运算符。

a+=b　　等价于　　a=a+b

注意：如果 b 是包含若干项的表达式，则相当于它有括号。

x%=y+3　　等价于　　x=x%(y+3)

（3）赋值表达式

赋值语句是在赋值表达式的末尾加上一个分号构成的。

由赋值运算符将一个变量和一个表达式连接起来的式子称为“赋值表达式”。它的一般形式为

变量　赋值运算符　表达式

注意：赋值运算符的左侧只能是变量，而不能是常量或者表达式。

（4）赋值过程中的类型转换

①如果赋值运算符两侧的类型一致，则直接进行赋值。

②如果赋值运算符两侧的类型不一致，但都是算术类型时，在赋值时要进行类型转换。转换的规则是：

将浮点型数据赋给整型变量时，先对浮点数取整，即舍弃小数部分，然后赋给整型变量。

将整型数据赋给实型变量时，数值不变，但以浮点型形式存储到变量中。

字符型数据赋给整型变量时，将字符的 ASCII 代码赋给整型变量。

将一个占字节多的整型数据赋给一个占字节少的整型变量或字符变量时，只将其低字节原封不动地送到被赋值的变量。

题 1.假定 int x=3,y=5;则执行表达式 y*=x++后，x 和 y 的值分别为_____和_____。

答案：4,15

题 2.C 语句“x*=y+2”还可以写作（　　）。

A. x=x*y+2　　B. x=2+y*x

C. x=x*(y+2)　　D. x=y+2*x

答案：C

题 3.若变量已正确定义并赋值，下面符合 C 语言语法的表达式是（　　）。

A.a:=b+1

B.int 18.5%3

C.a=a+7=c+b

D.a=b=c+2

答案：D

题 4.设 x,y 和 temp 均为 int 型变量，则以下语句：temp=x;x=y;y=temp;的功能是（　　）。

A.交换变量 x 和变量 y 的值

B.把 x 和 y 从大到小排列

C.把 x 和 y 从小到大排列

D.无确定结果

答案：A

6.输入输出语句

6.1 printf 函数

（1）把固定的内容输出到屏幕上。printf(“输出内容”);

```
#include<stdio.h>
int main()
{
    printf(“Hello World!”);
    printf(“Hello\nWorld!”);
    return 0;
}
```

（2）把变量的值输出到屏幕上。printf(“%x”,变量);

%d	输出 int 类型变量的值
%md	输出 int 类型变量的值
%c	输出 char 类型变量的值
%mc	输出 char 类型变量的值
%s	输出一个字符串
%f	输出 float 类型变量的值
%m.nf	输出 float 类型变量的值
%lf	输出 double 类型变量的值

```
#include<stdio.h>
int main()
{
    int a=13;
    printf(“%d”,a);
    printf(“a=%d”,a);
    printf(“a 的值是%d”,a);
    return 0;
}
#include<stdio.h>
int main()
{
    int a=65;
    char b=‘a’ ;
    float c=13.54;
    printf(“a=%d,”,a);
    printf(“b=%c,”,b);
    printf(“c=%f”,c);
    return 0;
}
//printf(“a=%d,b=%c,c=%f”,a,b,c);
```

```
//printf("a=%1d,b=%c,c=%f",a,b,c);
//printf("a=%5d,b=%c,c=%4.1f",a,b,c);
```

题 1.下列程序段的输出结果是（　　）。

```
int x=3,y=2;
printf("%d",(x-=y,x*=y+8/5));
```

A.5　　B.3

C.7　　D.1

答案：3

题 2.程序段 int a=123;printf("%4d",a);的输出结果是（　　）。

A.123　　B.1230

C. 123　　D.提示出错，无结果

答案：C

题 3.x 为整型变量，且值为 65，不正确的输出函数调用的是（　　）。

A.printf("%d",x);　　B.printf("%3d",x);

C.printf("%c",x);　　D.printf("%s",x);

答案：D

题 4.格式控制说明指定了输出数据的格式，它包含以%开头的格式控制字符，例如，int 型数据使用%d，char 类型数据使用（　　）。

A.%o　　B.%s

C.%c　　D.%f

答案：C

6.2 scanf 函数

一般形式：scanf(“%x”,&变量);

%d	int 类型变量的值
%c	char 类型变量的值
%f	float 类型变量的值
%lf	double 类型变量的值

注意：

①scanf 函数中的“格式控制”后面应当是变量地址，而不是变量名。

②如果在“格式控制字符串”中除了格式声明以外还有其他字符，则在输入数据时，在对应的位置上应输入与这些字符相同的字符。

```
#include<stdio.h>
int main()
{   int a;
    scanf(“%d”,&a);
    printf(“a=%d”,a);
    return 0;
}

#include<stdio.h>
int main()
{   int a;
    float b;
    scanf(“%d”,&a);
    scanf(“%f”,&b);
    printf(“a=%d,b=%f”,a,b);
    return 0;
}
//scanf(“%d%f”,&a,&b);
//scanf(“a=%d,b=%f”,&a,&b);
```

题 1.已知 i,j,k 为 int 变量，若从键盘输入：1,2,3<回车>，使 i 的值为 1，j 的值为 2，k 的值为 3。以下选项中正确的输入语句是（　　）。

A.scanf(“%2d,%2d,%2d”,i,j,k);

B.scanf(“%d%d%d”,&i,&j,&k);

C.scanf(“%d,%d,%d”,&i,&j,&k);

D.scanf(“i=%d,j=%d,k=%d”,&i,&j,&k);

答案：C

题 2.有输入语句:scanf(“a=%d,b=%d,c=%d”,&a,&b,&c);为使变量 a 的值为 3，b 为 6，c 为 9，从键盘输入数据的正确形式应当是（　　）。

A.369<回车>　　B.3,6,9<回车>

C.a=3,b=6,c=9<回车>　　D.a=3 b=6 c=9<回车>

答案：C

6.3 用 putchar 函数输出一个字符

putchar 函数的一般形式是

putchar(c)

注意：putchar(c)中的 c 可以是字符常量、整型常量、字符变量或整型变量（其值要在字符的 ASCII 代码范围内）。

6.4 用 getchar 函数输入一个字符

getchar 函数的一般形式是

getchar()

课时三 练习题

1.若有 int i=10，j=2;则执行完 i*=j+8;后 i 的值为 28。（判断题）

2.设有语句 int n=8,m=2;执行语句 m*=n+1;之后，m 的值为（　　）。

A.17　　B.18

C.19　　D.8

3.设 char x=‘a’;则 printf(“x=%c,y=%c\n”,x,97);的输出是（　　）。

A.x=a,y=97　　B.x=97,y=a

C.x=97,y=97　　D.x=a,y=a

4.设 char k=‘a’;float j=2.0;k+=5/j;printf(“%c”,k)，则输出（　　）。

A.a　　B.b

C.c　　D.d

5.printf 函数中用到格式控制符%5s，其中数字 5 表示输出的字符串占用 5 列，如果长度大于 5，则输出时（　　）。

A.从左起输出该字符串，右补空格

B.原字符长度从左向右全部输出

C.右对齐输出该字符串，左补空格

D.输出错误信息

6.程序段 char c1=‘a’,c2=‘b’,c3=‘c’;

printf(“a%cb%cc%cabc\n”,c3,c2,c1);运行结果是（　　）。

A.acbcccabc　　B.aabbccabc

C.abc　　D.acbbcaabc

7.有输入语句：scanf(“a=%d,b=%d,c=%d”,&a,&b,&c);为使变量 b 为 12，c 为 13，则从键盘输入数据的正确形式应当是（　　）。

A.11 12 13　　B.a=11,b=12,c=13

C.a=11 b=12 c=13　　D.11,12,13

8.有定义变量：int a;使用 scanf 函数对 a 赋值正确的是（　　）。

A.scanf(“%d”,a);　　B.scanf(“%f”,&a);

C.scanf(“%d”,&a);　　D.scanf(“%s”,&a);

9.以下程序能将 3,4,5 正确输入变量 x,y,z 的是（　　）。

```
#include<stdio.h>
void main()
{
    int x,y,z;
    scanf(“x=%d,y=%d,z=%d”,&x,&y,&z);
}
```

A.3 4 5　　B.3,4,5

C.345　　D.x=3,y=4,z=5

课时四 选择结构程序设计

考点	重要程度	分值	题型
1.用 if 语句实现选择结构	必考	15～25	选择题、编程题
2.关系运算符和关系表达式	★★★★★	6～10	选择题、填空题
3.逻辑运算符和逻辑表达式	★★★★	6～10	选择题、填空题
4.条件运算符和条件表示式	★★★	2～4	选择题
5.switch 语句实现多分支选择结构	★★★	2～8	选择题、读程序题

1.用 if 语句实现选择结构

if 语句的一般形式为：

if(表达式) 语句 1

[else 语句 2]

常见的 3 种形式：

①if(表达式)　　语句 1

②if(表达式)　　语句 1

　else　　语句 2

③if(表达式 1)　　语句 1

　else if(表达式 2)　　语句 2

　else if(表达式 3)　　语句 3

　else if(表达式 m)　　语句 m

　else　　语句 m+1

说明：

①“语句 1”、“语句 2”……“语句 m+1”等是 if 语句的“内嵌语句”。每个内嵌语句的末尾都应当有分号。

②else 子句不能作为语句单独使用，它是 if 语句的一部分，必须与 if 配对使用。

③在 if 语句中要对给定的条件进行检查，判定所给定的条件是否成立。判断的结果是一个逻辑值“是”或者“否”。在计算机语言中用“真”或者“假”来表示“是”或者“否”。

题 1.当程序输入 4 3 时，下面程序的运行结果为_____。

```
#include<stdio.h>
int main()
{
    int a,b,s;
    scanf(“%d %d”,&a,&b);
    s=a;
    if(a<b)
        s=b;
    s*=s;
    printf(“%d\n”,s);
    return 0;
}
```

答案：16

题 2.为避免 if else 语句的二义性，C 语言规定 else 与（　　）组成配对关系。

A.缩排位置相同的 if

B.在其前未配对的 if

C.在其之前未配对的最近的 if

D.同一行上的 if

答案：C

题 3.程序段（　　）的功能是将变量 u，s 中的最大值赋给变量 t。

A.if(u>s) t=u; t=s;　　B.t=u; if(t>s) t=s;

C.if(u>s) t=s; else t=u;　　D.t=s; if(u>t) t=u;

答案：D

题 4.对任意输入的 x，用下式计算并打印出 y 的值。（编程题）

$$y=\begin{cases} e^{x} & x>10 \\ 0 & x=10 \\ 3x+5 & x<10 \end{cases}$$

```
#include<stdio.h>
#include<math.h>
#define e 2.718
int main()
{
    float x,y;
    scanf("%f",&x);
    if(x>10)        y=pow(e,x);
    else if(x==10)  y=0;
    else            y=3*x+5;
    printf("%f",y);
    return 0;
}
```

2.关系运算符和关系表达式

在 C 语言中，比较符称为关系运算符。关系运算符的值为“真”（即“条件满足”），为“假”（即“条件不满足”）。

（1）关系运算符及其优先顺序

①＜　　(小于)
②＜＝　　(小于或者等于)
③＞　　(大于)
④＞＝　　(大于或者等于)
⑤＝＝　　(等于)
⑥！＝　　(不等于)

关于优先次序：

①前 4 种优先级相同，后 2 种也相同。前 4 种高于后 2 种。

②关系运算符优先级低于算术运算符。

③关系运算符优先级高于赋值运算符。

（2）关系表达式：用关系运算符将两个数值或数值表达式连接起来的式子。

关系表达式的值是一个逻辑值，即“真”或“假”。

在 C 的逻辑运算中，以“1”代表“真”，以“0”表示“假”。

3.逻辑运算符和逻辑表达式

（1）逻辑运算符及其优先次序

表 1 C 逻辑运算符及其含义

运算符	含义	举例	说明
&&	逻辑与	a&&b	a 和 b 都为真，则结果为真，否则为假
||	逻辑或	a||b	a 和 b 至少一个为真时，则结果为真；二者都为假时，结果为假
!	逻辑非	!a	如果 a 为假，则!a 为真；如果 a 为真，则!a 为假

表 2 逻辑运算的真值表

a	b	!a	!b	a&&b	a||b
真	真	假	假	真	真
真	假	假	真	假	真
假	真	真	假	假	真
假	假	真	真	假	假

关于优先次序：

①!　　&&　　||

②逻辑运算符中的“&&”和“||”低于关系运算符，“!”高于算术运算符。

（2）逻辑表达式

逻辑表达式的值是一个逻辑值“真”或“假”。

C 语言系统中，以 1 代表“真”，以 0 代表“假”，但在判断一个量是否为真时，以 0 代表“假”，以非 0 代表“真”。即将一个非零的数值认作为“真”。

题 1. 写出变量 t 能被 3 整除并且是偶数的 C 语言表达式为_____。

答案：(t%3==0)&&(t%2==0)

题 2. 为表示$x \geqslant y \geqslant z$，应使用 C 语言表达式（　　）。

A. (x>=y)&&(y>=z)　　B. (x>=y) AND (y>=z)

C. (x>=y>=z)　　D. (x>=z) & (y>=z)

答案：A

题 3. 下列表达式中，值为 0 的表达式是（　　）。

A. 3!=1　　B. 3!=3<4

C. 3>4==0　　D. 6>5>4

答案：D

题 4. 下面程序的功能是从键盘输入实数 x，计算并输出 y 的值。

$$y=\begin{cases}|x^3+4| & x<1\\ \cos(x) & 1\leqslant x\leqslant 10\\ 2x-1 & x>10\end{cases}$$

```
#include<stdio.h>
#include<math.h>
int main()
{
    float x,y;
    scanf(“%f”,&x);
    if(x<1)                    _____;
    else if(_____)             y=cos(x);
    else                       y=2*x-1;
    _____;
    return 0;
}
```

答案：y=fabs(x*x*x+4)

(x>=1)&&(x<=10)

printf(“%f”,y)

4.条件运算符和条件表达式

条件运算符有两个符号 ？和 ：组成，必须一起使用。

条件表达式的一般形式:

表达式 1 ？表达式 2 ：表达式 3

执行顺序：先求解表达式 1，若为非 0（真）则求解表达式 2，此时表达式 2 的值就作为整个条件表达式的值。若表达式 1 的值为 0（假），则求解表达式 3，表达式 3 的值就是整个条件表达式的值。

关于优先次序：

①条件运算符优先于赋值运算符。

②条件运算符的优先级别比关系运算符和算术运算符都要低。

题 1.老年人优惠景区观摩票的规定：60-65 岁票价 6.5 折，用 C 语言的条件表达式（age 表示年龄，p 表示票价）表示则（　　）。

A.(60<=age&&age<=65) ? p*0.65 : p

B.(60<age&&age<65) ? p*0.65 : p

C.(60<=age||age<=65) ? p*0.65 : p

D.(60<=age<=65) ? p* 0.65 : p

答案：A

题 2.已定义变量 int x=10,y=20;表达式 x>3?y+5:y+9 的值为_____。

答案：25

5.switch 语句实现多分支选择结构

if 只有两个分支可供选择，switch 语句直接处理多分支选择。

switch 语句的一般形式：

```
switch(表达式)
{
   case 常量 1：  语句 1
   case 常量 2：  语句 2

   case 常量 n：  语句 n
   default：      语句 n+1
}
```

说明：

①switch 后面括号内的“表达式”，其值的类型为整数类型（包括字符型）。

②switch 下面的花括号内的复合语句是 switch 语句的语句体，包含多个以关键字 case 开头的语句行和最多一个以 default 开头的行。

③可以没有 default 标号，此时如果没有与 switch 表达式相匹配的 case 常量，则不执行任何语句，流程跳转到 switch 语句的下一个语句。

④各个 case 标号出现次序不影响执行结果。

⑤每一个 case 常量必须互不相同。

⑥执行一个 case 标号的语句后，从此标号开始执行下去，不再进行判断。

⑦一般情况下，在执行一个 case 子句后，应当用 break 语句使流程跳出 switch 结构，即终止 switch 语句的执行。最后一个 case 子句（或为 default 子句）可不必加 break。

⑧多个 case 标号可以共用一组执行语句。

例如：

```
case ‘A’ :
case ‘B’ :
case ‘C’ : printf(“>60\n”); break;
```

题 1.执行下列代码后，屏幕的输出结果是（　　）。

```
#include<stdio.h>
int main()
{
    char x= ‘A’;
    switch(x)
    {
        case ‘A’: printf("It is A.\n");
        case ‘B’: printf("It is B.\n"); break;
        case ‘C’: printf("It is C.\n");
        default: printf("other.\n");
    }
}
```

A.It is A.　　B.It is B.　　C.It is C.　　D.无法运行
　It is B.　　It is C.

答案：A

题 2.下列叙述中正确的是（　　）。

A.break 语句只能用于 switch 语句。

B.在 switch 语句中必须使用 default 语句。

C.break 语句必须与 switch 语句中的 case 配对使用。

D.在 switch 语句中，不一定使用 break 语句。

答案：D

课时四 练习题

1.已知 int x=11,y=22,z=33;if(x>y) {z=x;x=y;y=z};执行后 x，y，z 的值是（　　）。

A.x=11,y=22,z=33　　B.x=22,y=33,z=33

C.x=22,y=33,z=11　　D.x=22,y=33,z=22

2.两次运行下面的程序，如果从键盘上分别输入 6 和 3，则输出结果是（　　）。

```
#include<stdio.h>
int main()
{
    int x;
    scanf("%d",&x);
    if(x++>5)  printf("%d",x);
    else  printf("%d\n",x--);
}
```

A.7 和 5　　B.6 和 3

C.7 和 4　　D.6 和 4

3.下面程序的运行结果为_____。

```
#include<stdio.h>
int main()
{
    int a=-5;
    if(a<0)
        if(a>-3)
            a+=5;
        else
            a+=10;
    else
        a-=5;
    printf(“a=%d\n”,a);
}
```

4.编辑程序计算下面分段函数的值。

$$y=\begin{cases}2x+5 & x<0\\ x^2-x+3 & x=0\\ x^3-7x & x>0\end{cases}$$

5.表示条件：10<x<100 或者 x<0 的 C 语言表达式是_____。

6.若有：int x=2,y=2,z=2;执行语句++x||++y&&++z;后，x,y,z 的值分别为(　　)。

A.3 3 3　　B.2 3 3

C.3 2 2　　D.以上都不是

7.下列程序的运行结果是（　　）。

```
#include<stdio.h>
int main()
{
    int a=4,b=5,c=0,d;
    d=!a&&!b||!c;
    printf(“%d\n”,d);
}
```

A.1　　B.0

C.非 0 的数　　D.无

8.下面程序的运行结果为（　　）。

```
#include<stdio.h>
int main()
{
    int x=-10,y=5,z=0;
    if(x=y+z)   printf(“***\n”);
    else   printf(“$$$\n”);
}
```

A.有语法错误不能通过编译　　B.输出$$$

C.输出***　　D.可以通过编译不能通过连接

9.编写程序，提示用户输入一个 3 位数，表示手机号码前 3 位，输出运营商信息。

134-139、150-152、157-159 为中国移动；

130-132、155-156、185-186 为中国联通；

133、153、180、189 为中国电信；

不属于以上任何一类的，输出“无此信号段”。

10.如果 int a=3,b=4,min;运行 min=(a>b)?a:b;语句后，min 的数值为（　　）。

A.3　　B.4　　C.1　　D.7

11.执行以下程序后，变量 a，b，c 的值分别为（　　）。

```
int x=10,y=9,a,b,c;
a=(--x==y++)?—x:++y;
b=x++;
c=y;
```

A.a=8,b=8,c=10　　B.a=9,b=9,c=9

C.a=9,b=10,c=9　　D.a=1,b=11,c=10

12.下列关于 switch 语句的描述中，正确的是（　　）。

A.switch 语句中的 default 子句不可省略。

B.switch 语句中每个 case 子句都必须包含 case 语句。

C.switch 语句中的多个 case 可共用一组执行语句。

D.switch 语句中至少应有一个 case 子句和 default 子句。

课时五　循环结构程序设计

考点	重要程度	分值	题型
1.用 while 语句实现循环	必考	15～25	选择题、编程题
2.用 do while 语句实现循环	★★★★	0～5	选择题
3.用 for 语句实现循环	必考	15～25	选择题、编程题
4.几种循环的比较	★★	0～2	选择题、判断题
5.改变循环执行的状态	★★★★	2～5	选择题、填空题、程序题

1.用 while 语句实现循环

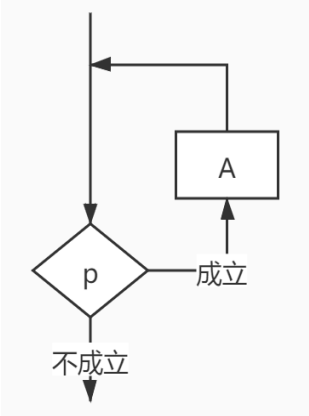
while 语句的一般形式如下：

while(循环条件表达式) 语句

只要当循环条件表达式为真（即给定的条件成立），就执行循环体语句。

while 循环特点是：先判断条件表达式，后执行循环体。

while 循环体可以一次都不执行。



说明：

①循环体如果包含一个以上的语句，应该用花括号括起来。

②不要忽略给变量赋初值，否则它们的值不可预测的。

③在循环体中应有使循环趋于结束的语句。若无该语句，则循环永不结束。

题 1.设有语句 int k=10; while(k=0)　k=k-1; 下面描述正确的是（　　）。

A.循环体语句执行一次　　B.循环是无限循环

C.循环体一次也不执行　　D.循环体语句执行 10 次

答案：C

题 2.以下程序运行时的输出结果是（　　）。

```
#include<stdio.h>
int main()
{
    int a=1,b=2,c=2,t;
    while(a<b<c)
    {
        t=a;  a=b;  b=t;
        c--;
    }
    printf(“%d,%d,%d”,a,b,c);
}
```

A.1,2,0　　B.2,1,0

C.1,2,1　　D.2,1,1

答案：A

题 3.下列程序段，如果从键盘输入 abcde?fgh<回车>，运行结果为_____。

```
#include<stdio.h>
int main()
{
    char c;
    c=getchar();
    while(c!= '?' )
    {
        putchar(c);
        c=getchar();
    }
}
```

答案：abcde

题 4.下面的程序段从键盘输入的字符统计数字字符的个数，用换行符结束循环，请填空。

```
#include<stdio.h>
int main()
{
    int n=0;
    char c=getchar();
    while(_____)
    {
        if(_____)   n++;
        c=getchar();
    }
    printf("%d",n);
}
```

答案：c!= '\n'

c>= '0' &&c<= '9'

题 5. 求 1+2+3+……+100。

```
#include<stdio.h>
int main()
{
    int i=1,s=0;
    while(i<=100)
    {
        s+=i;
        i++;
    }
    printf("%d",s);
    return 0;
}
```

2. 用 do while 语句实现循环

do while 语句的一般形式为：

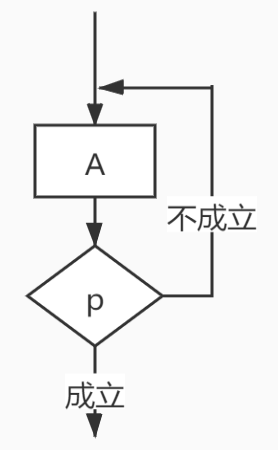
do

语句

while(表达式);

do while 语句的特点：先无条件执行循环体，然后判断循环条件是否成立。

do while 循环体至少执行一次。



题 1.do while 语句先执行循环中的语句，然后再判断表达式是否为真，如果为真则继续循环，如果为假，终止循环。（判断题）

答案：正确

题 2.以下程序的输出结果是（　　）。

```
#include<stdio.h>
int main()
{   int i=1;
    do{ printf(“%d”,i);
        i++;
        i++;
    }while(i<=5);
}
```

A.025　　B.135　　C.245　　D.345

答案：B

题 3.以下程序的输出结果是_____。

```
#include<stdio.h>
int main()
{   int number,digit;
    number=13579;
    do
    {
        digit=number%10;
        printf(“%d”,digit);
        number=number/10;
    }while(number!=0);
}
```

答案：97531

3.用 for 语句实现循环

for 语句的一般形式为：

for(表达式 1;表达式 2;表达式 3)

语句

表达式 1：设置初始条件，只执行一次。可以为零个、一个或多个变量设置初值。

表达式 2：循环条件表达式，用来判定是否继续循环。在每次执行循环体前先执行此表达式，决定是否继续执行循环。

表达式 3：循环的调整，例如循环变量增值，在执行完循环体后才进行的。

for(循环变量赋初值;循环条件;循环变量增值)

语句

说明：

①表达式 1；

```
while（表达式 2）
{
    语句
    表达式 3
}
```

②表达式 2 一般是关系表达式或者逻辑表达式，但也可以是数值表达式或字符表达式，只要其值为非 0，就执行循环体。

③允许在 for 语句的“表达式 1”中定义变量并赋值。

题 1.语句 for(i=1;i<10;i+=2);执行后 i 的值为（　　）。

A.9　　B.10　　C.11　　D.12

答案：C

题 2.设有 int iter=3,item=4,result=1;则语句

for(i=0;i<iter;i++){ item+=4; result*=item;}

运行后 result 的值是（　　）。

A.21945　　B.1155

C.30720　　D.1536

答案：D

题 3.下面程序的运行结果是_____。

```
#include<stdio.h>
int main()
{
    int a=0,k;
    for(k=0;k<3;k++)
    {
        switch(k)
        {
            case 1:
            case 2:  a++;  break;
            case 3:  a+=2;
        }
        printf(“a=%d\n”,a);
    }
}
```

答案：a=0
　　a=1
　　a=2

题 4.下面程序的功能是求出数字 0 至 9 可以组成多少个没有重复的三位偶数，请填空。

```
#include<stdio.h>
int main()
{   int i,j,k,n;
    __________;
    for(i=1;i<=9;i++)
    {   for(k=0;k<=8;k++)
        {   if(k!=i)
            {   for(j=0;j<=9;j++)
                {   if(__________)
                    {   printf("%d  ",i*100+j*10+k);
                        n++;
                        if(n%20==0)
                            putchar(10);
                    }
                }
            }
        }
    }
    printf("%d",n);
}
```

答案：n=0

　　j!=i&&j!=k&&k%2==0

题 5.把 100 到 200 之间不能被 3 整除的数进行输出。（编程题）

```
#include<stdio.h>
int main()
{
    int i;
    for(i=101;i<200;i++)
    {
        if(i%3!=0)
            printf("%d ",i);
    }
    return 0;
}
```

4.几种循环的比较

（1）3 种循环都可以用来处理同一个问题，一般情况下可以相互代替。

（2）while 和 do while 循环中，只在 while 后面的括号里指定循环条件，因此为了使循环能正常结束，应在循环体中包含使循环趋于结束的语句。

（i++,i--）

（3）while 和 do while 循环，循环变量的初始化应在循环语句之前完成。for 语句可以在表达式 1 中完成循环变量的初始化。

（4）for 循环不仅可以用于循环次数已经确定的情况，还可以用于循环次数不确定而只给出循环结束条件的情况。

（5）3 种循环都可以用 break 跳出循环，用 continue 语句结束本次循环。

题 1.循环体至少被执行一次的循环语句是（　　）。

A.for 循环　　B.while 循环

C.do while 循环　　D.任一种循环

答案：C

题 2.关于 while 和 do while 循环的描述中，不正确的是（　　）。

A.while 循环体可以用 break 语句退出。

B.do while 循环体可以用 break 语句退出。

C.while 循环至少无条件执行一次。

D.do while 循环体至少无条件执行一次。

答案：C

5.改变循环执行的状态

5.1 用 break 语句提前终止循环

一般形式为

```
break;
```

其作用是使流程跳到循环体之外，接着执行循环体下面的语句。

break 语句只能用于循环语句和 switch 语句中，而不能单独使用。

5.2 用 continue 语句提前结束本次循环

一般形式为

```
continue;
```

其作用为结束本次循环，即跳过循环体中下面尚未执行的语句。

5.3 break 语句和 continue 语句的区别

continue 语句只结束本次循环，而不是终止整个循环的执行。而 break 语句则是结束整个循环过程，不再判断执行循环的条件是否成立。

例如：

```
int i=1;
while(i<=100)
{
    if(i==3)    break;/continue;
    printf(“%d\n”,i);
    i++;
}
```

题 1.在使用循环结构中，break 和 continue 语句的区别在于（　　）。

A.break 终止整个 C 语言的运行，而 continue 终止循环体的运行。

B.break 使程序暂停，而 continue 则继续执行下一语句。

C.break 终止循环体的执行，而 continue 只结束本次循环的执行。

D.没有区别。

答案：C

题 2.下列程序的运行结果为_____。

```
#include<stdio.h>
int main()
{
    int i,j,a=0;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            if(j%2)    continue;
            a++;
        }
        a++;
    }
    printf(“%d\n”,a);
}
```

答案：9

题 3.下列程序的运行结果为_____。

```
#include<stdio.h>
int main()
{
    int n=0;
    while(n<6)
    {
        n++;
        if(n==3)    break;
        printf(“%d”,n);
    }
}
```

答案：12

课时五 练习题

1.设有程序段： int k=3;
while(k)
k=k-1;

则下面的叙述中正确的是（　　）。

A.while 循环执行 3 次　　B.循环是无限循环

C.循环体一次也不执行　　D.循环体语句执行一次

2.有如下程序：

```
#include<stdio.h>
int main()
{
    int n=9;
    while(n>6)
    {
        n--;
        printf(“%d”,n);
    }
}
```

则该程序的输出结果是（　　）。

A.987　　B.876　　C.8765　　D.9876

3.以下程序的运行结果是_____。

```
#include<stdio.h>
int main()
{   int digit=0,n=7394;
    while(n)
    {
         digit+=n%10;
         n=n/10;
    }
    printf(“digit=%d”,digit);
    return 0;}
```

4.下面程序段的功能是求 n 的阶乘，请填空。

```
#include<stdio.h>
int main()
{
    int n,i;
    int value;
    printf(“input n=”);
    _____;
    i=1;
    _____;
    while(i<=n)
    {
        _____;
        i++;
    }
    printf(“value=%d”,value);
}
```

5.用 $\frac{\pi}{4}\approx 1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\cdots$ 公式求 π 的近似值，直到发现某一项的绝对值小于 10^{-6} 为止（该项不累加）。

6.以下程序段（　　）。

```
int x=2;
do
{
    x=x*x;
}while(!x);
```

A.是死循环　　B.循环执行两次

C.循环执行一次　　D.有语法错误

7.下面程序段的输出结果是_____。

```
#include<stdio.h>
int main()
{
    int n=2018,sum=0;
    do
    {
        sum+=(n%10)*(n%10);
        n=n/10;
    }while(n);
    printf(“%d”,sum);
}
```

8.设变量已正确定义，能正确计算f=n!(n的阶乘)的程序段的是（　　）。

A.f=0;　　for(i=1;i<=n;i++)　　f*=n;

B.f=0;　　for(i=1;i<=n;i++)　　f*=i;

C.f=1;　　for(i=n;i>=2;i--)　　f*=i;

D.f=1;　　for(i=n;i>1;i++)　　f*=i;

9.以下程序的输出结果为（　　）。

```
#include<stdio.h>
int main()
{   int a=0,i;
    for(i=1;i<5;i++)
    {  switch(i){
           case 0:
           case 3: a+=2;
           case 1:
           case 2: a+=3;
           default: a+=5;}
    }
    printf(“%d\n”,a);
}
```

A.20　　B.13　　C.10　　D,31

10.以下程序段运行后，count 的值为_____。

```
#include<stdio.h>
int main()
{
    int i,j,count=0;
    for(i=1;i<=4;i++)
    {
        count+=2;
        printf(“%d ”,count);
    }
}
```

11.下列程序用于计算 1!+3!+5!+……+(2n-1)!，其中 n 的值由键盘输入，将程序补充完整。

```
#include<stdio.h>
int main()
{
    int n,i;
    long s,t;
    scanf(“%d”,&n);
    s=_____;
    t=1;
    for(i=1;i<=2*n-1;i++)
    {
        _____;
        t=t*(i+1)*(i+2);
    }
    printf(“s=ld\n”,s);
}
```

12.编程输出以下图形。

```
*
**
***
****
*****
```

13.循环体至少被执行一次的循环语句是_____。

14.C 语言中，while 与 do while 循环主要区别是（　　）。

A.do while 循环体至少无条件执行一次。

B.while 的循环控制条件比 do while 严格。

C.do while 允许从外部转向循环体内。

D.do while 循环体不能是复合语句。

15.下列错误的描述是（　　）。

A.break 语句不能用于循环语句、switch 语句之外的任何其他语句。

B.在 switch 语句中使用 break 语句或 continue 语句的作用相同。

C.循环语句中使用 continue 语句是为了结束本次循环,而不是终止整个循环。

D.在循环语句中使用 break 语句是为了使流程跳出循环体，提前结束循环。

16.下列程序的运行结果是_____。

```
#include<stdio.h>
int main()
{
    int a,b;
    for(a=1,b=1;a<=100;a++)
    {
        if(b>10)    break;
        if(b%3==1)  { b+=3;  continue; }
    }
    printf(“%d\n”,a);}
```

17.下面程序完成输入一个整数并判断是否为素数，请填空。

```
#include<stdio.h>
#include<math.h>
int main()
{
    int n,k,i=1;
    scanf(“%d”,&n);
    k=sqrt(n);
    for(i=2;i<=k;i++)
        if(__________)
            break;
    if(i>k)
        printf(“%d is a prime.\n”,n);}
```

课时六 利用数组处理批量数据（1）

考点	重要程度	分值	题型
1. 一维数组定义、引用和初始化	必考	12～18	选择题、编程题
2. 二维数组定义、引用和初始化	★★★	2～8	选择题、读程序题

1. 一维数组定义、引用和初始化

一批具有同名的同属性的数据就组成一个数组。

①数组是一组有序数据的集合。数组中各数据的排列是有一定规律的，下标代表数据在数组中的序号。

②用一个数组名和下标来唯一地确定数组中的元素。

③数组中的每一个元素都属于同一个数据类型。

1.1 定义一维数组

定义一维数组的一般形式为

类型符 数组名[常量表达式];

说明：

①数组名的命名规则和变量名相同，遵循标识符命名规则。

②在定义数组时，需要指定数组中元素的个数，方括号中的常量表达式用来表示元素的个数，即数组长度。

③常量表达式中可以包括常量和符号常量，不能包含变量。C 语言不允许对数组的大小作动态定义，即数组的大小不依赖于程序运行过程中变量的值。

题 1.下列数组的定义，错误的是（　　）。

A.int a[8+2];

B.#define N 10
int a[N];

C.int n=10;
int a[n];

D.float str[10];

答案：C

1.2 引用一维数组

引用数组元素的表示形式为

数组名[下标]

注意:

定义数组是用到的“数组名[常量表达式]”和引用数组元素时用的“数组名[下标]”形式相同，但含义不同。

题 1.若有说明：int a[10];则对 a 数组元素的正确引用是（　　）。

A.a[10]　　B.a[3.5]　　C.a(5)　　D.a[1+5]

答案：D

1.3 一维数组的初始化

在定义数组的同时，给各数组元素赋值，这称为数组的初始化。

（1）在定义数组时对数组元素赋予初值。

将数组中各元素的初值顺序放在一对花括号里，数据间用逗号分开。

（2）可以只给数组中的一部分元素赋值。

（3）若想使一个数组中全部元素值为 0，可以写成

```
int a[10]={0,0,0,0,0,0,0,0,0,0};
```

或

```
int a[10]={0};                    //未赋值的部分元素自动设为 0
```

（4）在对全部数组元素赋初值时，由于数据的个数已经确定，因此可以不指定数组长度。例如：

```
int a[5]={1,2,3,4,5};
```

可以写成

```
int a[]={1,2,3,4,5};
```

但是，如果数组长度与提供初值的个数不相同，则方括号的数组长度不能省略。例如，想定义数组长度为 10，就不能省略数组长度的定义，而必须写成

```
int a[10]={1,2,3,4,5};
```

只初始化前 5 个元素，后 5 个元素为 0。

说明：

如果在定义数值型数组时，指定了数组的长度并对之初始化，凡未被“初始化列表”指定初始化的数组元素，系统会自动把它们初始化为 0（如果是字符型数组，则初始化为‘\0’）。

题 1.以下对一维数组 a 进行正确的初始化的是（　　）。

A.int a[10]=(0,0,0,0);　　B.int a[10]={};

C.int a[]={0};　　D.int a[10]={10*2};

答案：D

题 2.已知 double 类型变量在内存中占用 8 个字节，定义数组 double a[10]={1,2,3,4,5};，则 sizeof(a)的值为（　　）。

A.8　　B.40　　C.80　　D.10

答案：C

题 3.阅读下面程序，写出程序的运行结果（　　）。

```
#include<stdio.h>
int main()
{
    int i,j,a[]={0,2,8,4,5};
    for(i=1;i<=5;i++)
    {
        j=5-i;
        printf(“%2d”,a[j]);
     }
     printf(“\n”);
     return 0;
}
```

A. 5 4 8 2 0　　B. 0 2 8 4 5

C. 1　　D. 0

答案：A

题 4.阅读下面程序，写出程序的运行结果为_____。

```
#include<stdio.h>
#define N 7
int main()
{
    int i,j,temp,a[N]={1,2,3,4,5,6,7};
    for(i=0;i<N/2;i++)
    {
        j=N-i-1;
        temp=a[i];
        a[i]=a[j];
        a[j]=temp;
    }
    for(i=0;i<N;i++)
        printf(“%2d”,a[i]);
    return 0;}
```

答案：7 6 5 4 3 2 1

题 5.定义数组 a[6]={10,7,15,20,3,1}，求出该数组的最大值，并输出其下标。（编程题）

```
#include<stdio.h>
int main()
{   int i,max,t,a[6]={10,7,15,20,3,1};
    max=a[0];
    t=0;
    for(i=0;i<6;i++)
    {   if(max<a[i])
        {
            max=a[i];
            t=i;
        }
    }
    printf("max=%d,t=%d",max,t);
    return 0;}
```

2. 二维数组定义、引用和初始化

2.1 定义二维数组

二维数组定义的一般形式

类型说明符　数组名[常量表达式][常量表达式];

C 语言中，二维数组中元素排列的顺序是按行存放的，即在内存中先顺序存放第 1 行的元素，接着再存放第 2 行的元素。

注意：

用矩阵形式表示二维数据，是逻辑上的概念，能形象地表示出行列关系。而在内存中，各元素是连续存放的，不是二维的，是线性的。

题 1. 下面二维数组的定义正确的是（　　）。

A. int a[][];　　B. int a[3,4];

C. int a(3)(4);　　D. int a[3][4];

答案：D

2.2 引用二维数组的元素

二维数组元素的表示形式为

数组名[下标][下标]

数组元素可以出现在表达式中，也可以被赋值。

注：在引用数组元素时，下标值应在已定义的数组大小的范围内。

题 1. 若有说明：a[3][4];则对数组 a 元素的正确引用的是（　　）。

A. a[2][4]　　B. a[1,2]

C. a(2)(1)　　D. a[1+1][0]

答案：D

2.3 二维数组的初始化

（1）分行给二维数组赋初值。例如：

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

这种赋初值的方法比较直观，把第一个花括号内的数据给第 1 行的元素，第 2 个花括号内的数据赋给第 2 行的元素……即按行赋初值。

（2）可以将所有数据写在一个花括号内，按数组元素在内存中的排列顺序对各元素赋初值。例如：

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

（3）可以对部分元素赋初值。例如：

```
int a[3][4]={{1},{5},{9}};
```

它的作用是只对各行第 1 列（即序号为 0 的列）的元素赋初值，其余元素值自动为 0。

也可以对各行中的某一元素赋初值，例如：

```
int a[3][4]={{1},{0,6},{0,0,11}};
```

也可以只对某几行元素赋初值，例如：

```
int a[3][4]={{1},{5,6}};
```

也可以对中间某一行不赋初值，例如：

```
int a[3][4]={{1},{ },{9}};
```

（4）如果对全部元素都赋初值（即提供全部初值数据），则定义数组时对第一维的长度可以不指定，但第二维的长度不能省。例如：

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

等价于

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

在定义时也可以只对部分元素赋初值而省略第一维的长度，但应分行赋初值。例如：

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

题 1.以下对二维数组 a 进行正确初始化的语句是（　　）。

A.int a[2][4]=[1,2,3,4];

B.int a[][3]={{1,2,3},{4,5,6}};

C.int a[2][]={{1,2,1},{4,2,3}};

D.int a[3][2]={{1,0,1},{5},{1,1}};

答案：B

题 2.有以下程序：

```
#include<stdio.h>
int main()
{
    int i,t[][3]={1,2,3,4,5,6,7,8,9};
    for(i=0;i<3;i++)
        printf("%d",t[2-i][i]);
    return 0;
}
```

程序执行后的输出结果是（　　）。

A.357　　B.753　　C.369　　D.751

答案：B

题 3.阅读下面程序，写出程序的运行结果_____。

```
#include<stdio.h>
int main()
{
    int i,sum,a[3][3]={1,2,3,4,5,6,7,8,9};
    sum=0;
    for(i=0;i<3;i++)
        sum+=a[i][i];
    printf(“%d\n”,sum);
    return 0;
}
```

答案：15

课时六 练习题

1.在一维整型数组a中存放8个整数，其正确定义是（　　）。

A.int a(8);

B.int n=8,a[n];

C.int n; scanf(“%d”,&n); int a[n];

D.#define SIZE 8　int a[SIZE];

2.若有语句int a[]={1,2,3,10,9,8,7,6,5};则a[a[2]+2]的值为_____。

3.有数组定义int a[10]={1,2,3,4};，则该数组的元素个数是（　　）。

A.3　　B.9　　C.10　　D.4

4.阅读下面程序，写出程序的运行结果（　　）。

```
#include<stdio.h>
int main()
{
    int i,a[10];
    for(i=0;i<10;i++)
        a[i]=i;
    for(i=9;i>=0;i--)
        printf(“%d”,a[i]);
    return 0;
}
```

A.10987654321　　B.987654321

C.9876543210　　D.0123456789

5.阅读下面程序，写出程序的运行结果_____。

```
#include<stdio.h>
int main()
{
    int a[5],k;
    for(k=0;k<5;k++)
        a[k]=10*k;
    printf(“%d”,a[k-1]);
    return 0;
}
```

6.由键盘输入 10 个整数到一维整型数组 a[10],输出大于平均值的数及其个数。

输入：1 5 3 3 4 6 2 2 2 7

输出：5 4 6 7（说明：最后一个数字后面保留一个空格）

number=4（说明：输出个数）

7.若有定义：double a[3][4];则数组中共有_____个数组元素。

8.若有说明：int a[3][4];则 a[3][4]是对二维数组的正确引用。（判断题）

9.以下不能正确初始化二维数组的定义的是（　　）。

A.int a[2][2]={{1},{2}};

B.int a[2][]={{1,2},{3,4}};

C.int a[2][2]={1,2,3};

D.int a[][2]={1,2,3,4};

10.有以下程序，程序运行后的输出结果是（　　）。

```
#include<stdio.h>
int main()
{
    int a[4][4]={{1,2,3,4},{5,6,7,8},{3,9,10,2},{4,2,9,6}};
    int i,s=0;
    for(i=0;i<4;i++)
        s+=a[i][3-i];
    printf("%d\n",s);
    return 0;
}
```

A.24　　B.19

C.13　　D.23

11.下面程序段执行后的输出结果是_____。

```
#include<stdio.h>
#define N 5
int main()
{   int i,j,a[N][N]={0};
    for(i=0;i<N;i++)
    {    a[i][i]=1;
         a[i][0]=1;
    }
    for(i=2;i<N;i++)
         for(j=1;j<=i-1;j++)
              a[i][j]=a[i-1][j-1]+a[i-1][j];
    for(i=0;i<N;i++)
    {    for(j=0;j<N;j++)
              printf("%2d",a[i][j]);
         printf("\n");
    }
    printf("\n");
    return 0;
}
```

课时七 利用数组处理批量数据（2）

考点	重要程度	分值	题型
3.字符数组	★★★★★	4～16	选择题、编程题

3.字符数组

字符型数据是以字符的 ASCII 代码存储在存储单元中的，一般占一个字节。

C 语言中没有字符串类型，字符串是存放在字符型数据中的。

3.1 定义字符数组

用来存放字符数据的数组是字符数组。字符数组中一个元素存放一个字节。

3.2 字符数组的初始化

对字符数组初始化，把各个字符依次赋给数组中各元素。

如果在定义字符数组时不进行初始化，则数组中各元素的值是不可预料的。如果花括号中提供的初值个数大于数组长度，则出现语法错误。如果初值个数小于数组长度，则只将这些字符赋给数组中前面那些元素，其余元素自动定为空字符（即‘\0’）。

如果提供初值的个数与预定的数组长度相同，在定义时可以省略数组长度，系统会自动根据个数确定数组长度。

题 1.下列对字符数组的初始化错误的是（　　）。

A.char ch[4]=“789”;

B.char ch[4]=“hijkl”;

C.char ch[4]={‘h’,‘i’,‘j’,‘k’};

D.char ch[]=“78hij”;

答案：B

3.3 引用字符数组的元素

可以引用字符数组中的一个元素，得到一个字符。

3.4 字符串和字符串结束标志

在 C 语言中，是将字符串作为字符数组来处理的。

C 语言规定了一个“字符串结束标志”，以字符‘\0’作为结束标志。

再对字符数组初始化的方法补充一种方法，即用字符串常量来使字符数组初始化。

例如：

char c[]={“I am happy”};

也可以省略花括号，直接写成

char c[]=“I am happy”;

等价于

char c[]={‘I’,‘ ’,‘a’,‘m’,‘ ’,‘h’,‘a’,‘p’,‘p’,‘y’,‘\0’};

而不与下面的等价：

char c[]={‘I’,‘ ’,‘a’,‘m’,‘ ’,‘h’,‘a’,‘p’,‘p’,‘y’};

说明：字符数组并不要求它的最后一个字符为‘\0’，甚至可以不包含‘\0’，

像以下这样是完全合法的：

char c[5]={‘C’,‘h’,‘i’,‘n’,‘a’};

题 1.char str[10]=“China”，数组元素个数为（　　）。

A.5　　B.10　　C.9　　D.6

答案：B

题 2.执行以下两条语句后，字符数组 a 和 b 的长度分别为（　　）。

char a[]={“happy”};

char b[]={‘h’,‘a’,‘p’,‘p’,‘y’};

A.5,5　　B.6,5　　C.5,6　　D.6,6

答案：B

3.5 字符数组的输入输出

（1）逐个字符输入输出。用格式符“%c”输入或输出一个字符。

（2）将整个字符串一次输入或输出。用格式符“%s”，意思是对字符串（string）的输入输出。

说明：

①输出的字符中不包括结束符‘\0’。

②用“%s”格式符输出字符串时，printf 函数中的输出项是字符数组名，而不是数组元素名。

③如果数组长度大于字符串的实际长度，也只输出到遇‘\0’结束。

④如果一个字符数组包含一个以上‘\0’，遇第一个‘\0’时输出就结束。

⑤可以用 scanf 函数输入一个字符串。

scanf(“%s”,c);

scanf 函数中的输入项 c 是已定义的字符数组名，输入的字符串应短于已定义的字符数组的长度。

注意：

scanf 函数中的输入项如果是字符数组名，不要再加地址符&，因为在 C 语言中数组名代表该数组的起始地址。

⑥前面介绍的输出字符串的方法：

printf(“%s”,c);

实际上是这样执行的：按字符数组名 c 找到其数组起始地址，然后逐个输出其中的字符，直到遇‘\0’为止。

题 1.下面程序段的运行结果是（　　）。

```
char c[5]={‘a’,‘b’,‘\0’,‘c’,‘\0’};
printf(“%s\n”,c);
```

A. ‘a’ ‘b’　　B.ab

C.ab　c　　D.a,b

答案：B

3.6 使用字符串处理数据

（1）puts 函数——输出字符串的函数

其一般形式为

puts(字符数组)

其作用是将一个字符串（以‘\0’结束的字符串序列）输出到终端。

题 1.若有声明：char a[10]=“123456”;则 puts(a+3)的结果为_____ 。

答案：456

（2）gets 函数——输入字符串的函数

其一般形式为

gets(字符数组)

其作用是从终端输入一个字符串到字符数组，并且得到一个函数值。该函数值是字符数组的起始地址。

注：用 puts 和 gets 函数只能输入或者输出一个字符串，不能写成

puts(str1,str2)或 gets(str1,str2)

（3）strcat——字符串连接函数

其一般形式为

strcat(字符数组 1,字符数组 2)

其作用是把两个字符数组中的字符串连接起来，把字符串 2 接到字符串 1 的后面，结果放在字符数组 1 中，函数调用后得到一个函数值——字符数组 1 的地址。

题 1.当执行下面的程序时，如果输入“ABC”，则输出结果是（　　）。

```
#include<stdio.h>
#include<string.h>
int main()
{
    char ss[10]=“12345”;
    gets(ss);
    strcat(ss,“6789”);
    printf(“%s\n”,ss);
    return 0;
}
```

答案：ABC6789

（4）strcpy 函数——字符串复制函数

其一般形式为

strcpy(字符数组 1，字符串 2)

其作用是将字符串 2 复制到字符数组 1 中去。

（5）strcmp 函数——字符串比较函数

其一般形式为

strcmp(字符串 1，字符串 2);

其作用是比较字符串 1 和字符串 2。

比较的结果有函数值带回。

①如果字符串 1=字符串 2，则函数值为 0。

②如果字符串 1>字符串 2，则函数值为一个正整数。

③如果字符串 1<字符串 2，则函数值为一个负整数。

注意：对两个字符串比较，不能用以下形式：

```
if(str1>str2)
    printf(“YES!”);
```

而只能用

```
if(strcmp(str1,str2)>0)
    printf(“YES!”);
```

题 1.为了判断两个字符串 s1 和 s2 是否相等，应当使用（　　）。

A.if(s1==s2)　　B.if(s1=s2)

C.if(strcmp(s1,s2))　　D.if(strcmp(s1,s2)==0)

答案：D

（6）strlen——测字符串长度的函数

其一般形式为

strlen(字符数组)

函数的值为字符串中的实际长度（不包括‘\0’在内）。

也可以直接测试字符串常量的长度。

注意：

在使用字符串处理函数时，应但在程序文件的开头用

#include<string.h>

把“string.h”文件包含到本文件中。

题 1.表达式 strlen(“hello”)的值为（　　）。

A.4　　B.5　　C.6　　D.7

答案：5

题 2.源程序中用到了 sqrt 和 strcat 函数，编译预处理包含的库文件为

_____和_____。

答案：math.h　　string.h

题 3.以下程序的运行结果是_____。

```
#include<stdio.h>
#include<string.h>
int main()
{
    char Line[]=“123456789”;
    int i,k=strlen(Line);
    for(i=1;i<3;i++)
    {
        Line[k-i]= ‘\0’ ;
        puts(Line+i);
    }
    return 0;
}
```

答案：2345678

　　34567

题 4.输入一行字符（小于 80 个字符），把所有非数字字符改写成空格（保留数字字符不变），然后输出修改后的字符串。（编程题）

例如：输入 123aa123bv87&&123,输出 123　123　87　123。

```
#include<stdio.h>
int main()
{   char s[80];
    gets(s);
    for(int i=0;s[i]!= ‘\0’;i++)
    {
        if(s[i]> ‘9’||s[i]< ‘0’)
            s[i]= ‘ ’;
    }
    puts(s);
    return 0;
}
```

课时七 练习题

1.以下对一维数组 a 的正确说明是（　　）。

A.char a(10);

B.int a[];

C.int n=5,a[n];

D.char a[3]={‘a’,‘b’};

2.设有数组定义：char array[10]=“china”; 则数组 array 所占的存储空间为（　　）。

A.4 个字节　　B.5 个字节

C.6 个字节　　D.10 个字节

3.若有以下的数组定义：char a[]=“abcde”;

char b[6]={‘a’,‘b’,‘c’,‘d’,‘e’}，则以下正确的叙述是（　　）。

A.a 数组和 b 数组长度不相同

B.a 数组长度大于 b 数组长度

C.a 数组长度小于 b 数组长度

D.两个数组中存放内容完全相同

4.判断字符串 str1 和 str2 是否相等，正确的语句是（　　）。

A.if(strcmp(s1,s2)==0)

B.if(strlen(s1)==strlen(s2))

C.if(s1==s2)

D.if(strcat(s1,s2)==0)

5.程序中调用了库函数 strcmp，必须包含头文件（　　）。

A.math.h　　B.string.h　　C.ctype.h　　D.stdlib.h

6.函数strlen(“C:\\data.txt”)的返回值是（　　）。

A.11　　B.12　　C.13　　D.15

7.实现字符串连接的系统函数名是（　　）。

A.strcat　　B.strcmp　　C.strcpy　　D.strlen

8.下面程序的功能是：从键盘输入一行字符，统计其单词个数，单词之间用空格分隔开，请填空。

```
#include<stdio.h>
int main()
{
    char string[80];
    int i,num=0,word=0;
    char c;
    gets(_____);
    for(i=0;(c=string[i])_____ ‘\0’ ;i++)
    {
        if(c== ‘ ’ ) word=0;
        else if(_____)
        {
            word=1;
            num++;
        }
    }
    printf(“There are %d words in the line.\n”,num);
    return 0;
}
```

课时八 用函数实现模块化程序设计（1）

考点	重要程度	分值	题型
1.定义函数	★★★★	0～10	编程题
2.调用函数	★★★★	0～8	选择题、编程题
3.对被调用函数的声明和函数原型	★★★	0～4	选择题
4.函数的嵌套调用	★★★	0～4	选择题、程序填空题

1.定义函数

所有函数都是平行的，即在定义函数时是分别进行的，是相互独立的。一个函数并不从属于另一个函数，即函数不能嵌套定义。函数之间可以相互调用，但不能调用 main 函数。main 函数是被操作系统调用的。

C 语言要求，在程序中用到的所有函数，必须“先定义，后使用”。

（1）定义无参函数

定义无参函数的一般形式为

```
类型名  函数名()                    类型名  函数名(void)
{                                   {
      函数体          或                  函数体
}                                   }
```

函数后面括号内的 void 表示“空”，即函数没有参数。

函数体包括声明部分和语句部分。

在定义函数时要用“类型名”指定函数值的类型，即指定函数带回来的值的类型。

（2）定义有参函数

定义有参函数的一般形式为

类型名　函数名（形式参数表列）

{

函数体

}

（3）定义空函数

定义空函数的一般形式为

类型名　函数名()

{ }

题 1.在函数的定义中，若函数没有参数，则可以省略函数名后面的括号。（判断题）

答案：错误

题 2.下列说法错误的是（　　）。

A.主函数可以分为两个部分：函数首部和函数体。

B.程序可以从任何非主函数开始执行。

C.非主函数可以调用其他任何非主函数。

D.主函数可以调用任何非主函数的其他函数。

答案：B

2.调用函数

2.1 函数调用的形式

函数调用的一般形式为

函数名(实参表列)

如果是调用无参函数，则“实参表列”可以没有，但括号不能省略。如果实参表列包含多个实参，则参数间用逗号隔开。

以下 3 种函数调用方式。

（1）函数调用语句

把函数调用单独作为一个语句，这时不要求函数带回值，只需要函数完成一定的操作。

（2）函数表达式

函数调用出现在另一表达式中，这时要求函数带回一个确定的值以参加表达式的运算。

（3）函数参数

函数调用作为另一个函数调用时的实参。

说明：调用函数并不一定要求包括分号，只有作为函数调用语句才需要有分号。如果作为函数表达式或函数参数，函数调用本身是不必有分号的。

题 1.若已定义的函数有返回值，则以下函数调用的叙述中错误的是（　　）。

A.函数调用可以作为单独的语句存在

B.函数调用可以作为一个函数的实参

C.函数调用可以出现在表达式中

D.函数调用可以作为一个函数的形参

答案：D

2.2 函数调用时的数据传递

（1）形式参数和实际参数

在调用有参函数时，主调函数和被调用函数之间有数据传递关系。在定义函数时函数名后面括号中的变量名称为“形式参数”（简称“形参”）。在主调函数中调用一个函数时，函数名后面括号中的参数称为“实际参数”（简称“实参”）。

（2）实参和形参间的数据传递

在调用函数过程中，系统会把实参的值传递给被调用函数的形参。或者说，形参从实参得到一个值。该值在函数调用期间有效，可以参加该函数中的运算。

说明：

①实参可以是常量、变量、表达式。

②实参与形参的类型应相同或赋值兼容。

题 1.若下列函数 fun()被调用 3 次后，最后一次输出 n 的值为（　　）。

```
void fun()
{    int n=1;
     ++n;
     printf(“%d”,n);
}
```

A.1　　B.2　　C.3　　D.4

答案：B

题 2.以下不正确的说法是（　　）。

A.实参可以是常量、变量或表达式。

B.实参可以是任何类型。

C.形参可以是常量、变量或表达式。

D.形参应与对应的实参类型一致。

答案：C

题 3.下列程序的输出结果是_____。

```
#include<stdio.h>
fun(int a,int b,int c)
{   c=a*b;   }
int main()
{
    int c=0;
    fun(2,3,c);
    printf(“%d\n”,c);
    return 0;
}
```

答案：0

题 4.设计函数 int shuixianhua(int x)，其功能是判断 x 是否为水仙花数，是则返回 1，否则返回 0。编写主函数，通过键盘输入一个三位正整数，调用该函数判断该数是否为水仙花数。所谓水仙花数就是指一个三位正整数，各位数字的立方和等于该数。（编程题）

```
#include<stdio.h>
int shuixianhua(int x)
{
    int a,b,c;
    a=x/100;
    b=(x-100*a)/10;
    c=x%10;
    if(x==a*a*a+b*b*b+c*c*c)
        return 1;
    else
        return 0;
}

int main()
{
    int n;
    scanf("%d",&n);
    if(shuixianhua(n)==1)
        printf("该数为水仙花数！");
    else
        printf("该数不是水仙花数！");
    return 0;
}
```

2.3 函数调用的过程

在定义函数中指定的形参，在未出现函数调用时，它们并不占内存中的存储单元。在发生函数调用时，函数的形参被临时分配内存单元。将实参对应的值传递给形参。如果函数不需要返回值，则不需要 return 语句。这时，函数的类型应定义为 void 类型。调用结束，形参单元被释放。

注意：

实参单元仍保留并维持原值，没有改变。实参向形参的数据传递是“值传递”，单项传递，只能由实参传给形参，而不能由形参传给实参。实参和形参在内存中占有不同的存储单元，实参无法得到形参的值。

题 1.以下说法正确的是（　　）。

A.实参与其对应的形参共同占用一个存储单元。

B.实参与对应的形参各占用独立的存储单元。

C.只有当实参与其对应的形参同名时才占用一个共同的存储单元。

D.形参是虚拟的，不占用内存单元。

答案：B

题 2.阅读下面程序，写出程序的运行结果_____。

```
#include<stdio.h>
swap(int x,int y)
{   int temp;
    temp=x;   x=y;   y=temp;
}
int main()
{   int x=7,y=11;
    swap(x,y);
    printf(“x=%d,y=%d”,x,y);
    return 0;
}
```

A.x=7,y=11　　B.x=11,y=7

C.x=1,y=1　　D.x=0,y=0

答案：A

2.4 函数的返回值

（1）函数的返回值是通过函数中的 return 语句获得的。

（2）函数值的类型

（3）在定义函数时指定的函数类型一般应该和 return 语句中的表达式一致。函数类型决定返回值的类型。

（4）对于不带回值的函数，应当用定义函数为“void 类型”（或称“空类型”）。即禁止在调用函数中使用被调用函数的返回值。此时在函数体中不得出现 return 语句。

题 1.C 语言规定，函数返回值的类型是由（　　）决定的。

A.return 语句中的表达式类型所决定的。

B.调用该函数时的主调用函数类型所决定的。

C.调用该函数时系统决定。

D.在定义该函数时指定的函数类型所决定。

答案：D

题 2.关于 return 语句的叙述中正确的是（　　）。

A.一个自定义函数中必须有一条 return 语句。

B.一个自定义函数中可以根据不同情况设置多条 return 语句。

C.定义成 void 类型的函数可以有带返回值的 return 语句。

D.没有 return 语句的自定义函数在执行结束时，不能返回到调用处。

答案：B

3.对被调用函数的声明和函数原型

在一个函数中调用另一个函数（即被调用函数）需要具备如下条件：

（1）首先被调用的函数必须是已经定义的函数。

（2）如果使用库函数，应该在本文件开头用#include 指令将调用有关库函数时所需用到的信息“包含”到本文件中来。

（3）如果使用用户自己定义的函数，而该函数的位置在调用它的函数（即主调函数）的后面，应该在主调函数中对被调用的函数作声明。声明的作用是把函数名、函数参数的个数和参数类型等信息通知编译系统。

函数的声明和函数定义中的第一行（函数首部）基本上是相同的，函数声明比函数定义首行多一个分号。函数的首行（即函数首部）称为函数原型。

函数原型的一般形式有两种，分别为：

（1）函数类型 函数名(参数类型 1 参数名 1,参数类型 2 参数名 2, ……,参数类型 n 参数名 n);

（2）函数类型 函数名(参数类型 1,参数类型 2,……,参数类型 n);

注意：

对函数的“定义”和“声明”不是一回事。函数的定义是指对函数功能的确定，包括指定函数名、函数值类型、形参及其类型以及函数体等，它是一个完整的、独立的函数单位。而函数的声明的作用则是把函数的名字、函数类型以及形参的类型、个数和顺序通知编译系统，以便在调用该函数时系统按此进行对照检查（例如，函数名是否正确，实参与形参的类型和个数是否一致），它不包含函数体。

4.函数的嵌套调用

C 语言的函数定义是相互平行的、独立的。在定义函数时，一个函数内不能再定义另一个函数，也就是不能嵌套定义，但可以嵌套调用，也就是说，在调用一个函数的过程中，又调用另一个函数。

题 1.在 C 程序中，函数既不能嵌套定义，也不能嵌套调用。（判断题）

答案：错误

题 2.执行下列程序段代码后，屏幕的输出结果是_____。

```
#include<stdio.h>
int func2(int a,int b)
{   int c;
    c=a+b;
    return c;
}
int func1(int a,int b)
{
    int c;
    a-=a;
    b+=b;
    c=func2(a,b);
    return c+2;
}
int main()
{
    int x=4,y=3;
    printf(“func1(%d,%d)=%d\n”,x,y,func1(x,y));
    return 0;
}
```

答案：func1(4,3)=8

课时八 练习题

1.以下正确的函数声明形式是（　　）。

A.double f(int x,int y)　　B.double f(int x;int y)

C.double f(int x,int y);　　D.double f(int x,y)

2.函数的编写需要注意格式。若程序需要实现求两个整数的平均值，则程序的错误在于（　　）。

```
#include<stdio.h>
int Func(int x,int y) (   return x+y;   )
int main()
{
    int x=10,y=20;
    printf(“average=%f”,Func(x,y)/2.0);
}
```

A.函数 Func 返回值必须是 float

B.函数 Func 的函数体没有大括号包围

C.输出格式转换说明符应该是%f

D.没有错误

3.下列说法正确的是（　　）。

A.C 语言程序的每行中只能写一条语句。

B.C 语言中 main 函数一定在函数的开始部分。

C.C 语言程序总是从 main 函数开始执行。

D.函数调用，实参向形参传递值或者地址，所以实参和形参不能同名。

4.以下对C语言函数的描述中，正确的是（　　）。

A.C语言程序有一个或一个以上的函数组成。

B.函数既可以嵌套定义又可以递归调用。

C.函数必须有返回值，否则不能使用函数。

D.C程序中调用关系的所有函数必须放在同一程序文件中。

5.函数返回值的数据类型取决于return语句中表达式的数据类型。（判断题）

6.一个return语句可以返回多个值。（判断题）

7.阅读以下程序，写出程序的运行结果_____。

```
#include<stdio.h>
void fun(int x)
{
    printf(“%d\n”,x);
    x+=2;
    printf(“%d\n”,x);
}
int main()
{
    int x=2;
    fun(x);
    printf(“%d”,x);
    return 0;
}
```

8.以下函数值的类型为（　　）。

```
fun(float x)
{
    float y;
    y=3*x-4;
    return(y);
}
```

A.不确定　　B.float　　C.int　　D.void

9.当从键盘输入：6,3，下列程序的运行结果为_____。

```
#include<stdio.h>
int Sub(int a,int b)
{
    return(a-b);
}
int main()
{
    int x,y,result=0;
    scanf(“%d,%d”,&x,&y);
    result=Sub(x,y);
    printf(“result=%d\n”,result);
    return 0;
}
```

10.编程：定义函数 int prime(int x)，判断 x 是否为素数，若是则函数返回 1，否则函数返回 0.在主函数中输入任意整数 x，调用该函数判断是否为素数，根据返回值输出“该数是素数”或“该数不是素数”。编写函数 int prime(int x)和主函数。

11.在 C 语言程序中（　　）。

A.函数的定义可以嵌套，但函数的调用不可以嵌套。

B.函数的定义不可以嵌套，但函数的调用可以嵌套。

C.函数的定义和函数的调用均不可以嵌套。

D.函数的定义和函数的调用均可以嵌套。

课时九 用函数实现模块化程序设计（2）

考点	重要程度	分值	题型
5. 函数的递归调用	★★★	0～8	选择题、读程序题
6. 数组作为函数参数	★★★★	0～6	选择题、程序填空题
7. 局部变量和全局变量	★★★	0～4	选择题、读程序题
8. 变量的存储类别和生存期	★★	0～2	选择题

5. 函数的递归调用

在调用一个函数的过程中又出现直接或间接地调用该函数本身，称为函数的递归调用。

题 1. 对以下递归函数 f，调用 f(2)的返回值是_____。

```
int f(int x)
{   return (x<=0)?x:f(x-1)+f(x-2);    }
```

答案：-1

题 2. 调用函数 f(27)的输出结果是（　　）。

```
void f(int n)
{   if(n<5)
        printf("%d",n);
    else
    {   printf("%d",n%5);
        f(n/5);
    }
}
```

A. 102　　B. 201　　C. 21　　D. 20

答案：B

题 3.有 5 个学生坐在一起，问第 5 个学生多少岁，他说比第 4 个学生大 2 岁。问第 4 个学生岁数，他说比第 3 个学生大 2 岁。问第 3 个学生岁数，他说比第 2 个学生大 2 岁。问第 2 个学生岁数，他说比第 1 个学生大 2 岁。最后问第 1 个学生岁数，他说是 10 岁。请问第 5 个学生多大。

```
#include<stdio.h>
int main()
{
    printf("No.5.age:%d\n",age(5));
    return 0;
}

int age(int n)
{
    int c;
    if(n==1)
        c=10;
    else
        c=age(n-1)+2;
    return c;
}
```

6.数组作为函数参数

6.1 数组元素作为函数实参

数组元素可以用作函数实参，不能用作形参。因为形参是在函数被调用时临时分配内存单元的，不可能为一个数组元素单独分配存储单元（数组是一个整体，在内存中占连续的一段存储单元）。在用数组元素作函数实参时，把实参的值传给形参，是“值传递”方式。数据传递的方向是从实参传到形参，单向传递。

6.2 数组名作函数参数

用数组元素作实参时，向形参变量传递的是数组元素的值，而用数组名作函数实参时，向形参（数组名或指针变量）传递的是数组首元素的地址。

6.3 多维数组名作函数参数

可以用多维数组名作为函数的实参和形参，在被调用函数中对形参数组定义时可以指定每一维的大小，也可以省略第一维的大小说明。

题 1.当调用函数时，实参是一个数组名，则向函数传递的是（　　）。

A.数组的长度　　B.数组的首地址

C.数组每一个元素的地址　　D.数组中每个元素的值

答案：B

题 2.以下程序运行后，运行结果是_____。

```
#include<stdio.h>
int main()
{   int i,a[5]={3,9,11,6,22},n=5;
    reverse(a,n);
    for(i=0;i<n;i++)
        printf("%4d",a[i]);
    return 0;
}
void reverse(int p[],int n)
{   int i,j,t;
    for(i=0,j=n-1;i<j;i++,j--)
    {   t=p[i];
        p[i]=p[j];
        p[j]=t;
    }
}
```

答案：　22　6　11　9　3

7.局部变量和全局变量

7.1 局部变量

在一个函数内部定义的变量只在本函数范围内有效。在复合语句内定义的变量只在本复合语句范围内有效。以上这些称为“局部变量”。

注意：

（1）主函数中定义的变量也只在主函数中有效，并不因为在主函数中定义而在整个文件或程序中有效。主函数也不能使用其他函数中定义的变量。

（2）不同函数中可以使用同名的变量，它们代表不同的对象，互不干扰。

（3）形式参数也是局部变量。

（4）在一个函数内部，可以在复合语句中定义变量，这些变量只在本复合语句中有效。

7.2 全局变量

在函数内定义的变量是局部变量，而在函数之外定义的变量称为全局变量。全局变量可以为本文件中其他函数所共用。它的有效范围为从定义变量的位置开始到本源文件结束。

题 1.在 C 语言中，一定是在所有函数之外声明的是（　　）。

A.全局变量　　B.局部变量

C.形参　　D.实参

答案：A

题 2.以下叙述错误的是（　　）。

A.不同的函数中可以使用相同的变量名。

B.形式参数也是局部变量。

C.一个函数内部定义的变量只能在本函数范围内有效。

D.在一个函数内部的复合语句中定义的变量可以在本函数范围内有效。

答案：D

8.变量的存储方式和生存期

8.1 动态存储方式与静态存储方式

从变量的作用域（即从空间）的角度来观察，变量可以分为全局变量和局部变量。

从变量值存在的时间（即生存期）来观察，变量的存储可以分为静态存储方式和动态存储方式。静态存储方式是指在程序运行期间由系统分配固定的存储空间的方式，而动态存储方式则是在程序运行期间根据需要进行动态的分配存储空间的方式。

全局变量全部存放在静态存储区中，在程序开始执行时给全局变量分配存储区，程序执行完毕就释放。

每一个变量和函数都有两个属性：数据类型和数据的存储类别。存储类别指的是数据在内存中存储的方式。

C 的存储类别包括 4 种，自动的（auto）、静态的（static）、寄存器的（register）、外部的（extern）。

8.2 局部变量的存储类别

（1）自动变量（auto）

函数中的局部变量，若不专门声明为 static（静态）存储类别，都是动态地分配存储空间的，数据存储在动态存储区中。函数中的形参和在函数中定义的局部变量，都属于此类。

实际上，关键字“auto”可以省略，不写 auto 则隐含指定为“自动存储类别”，它属于动态存储方式。

（2）静态局部变量（static 局部变量）

函数中的局部变量的值在函数调用结束后不消失而继续保留原值，即其占用的存储单元不释放，在下一次再调用该函数时，该变量已有值（就是上一次函数调用结束时的值）。这时指定该局部变量为“静态局部变量”，用关键字 static 声明。

（3）寄存器变量（register 变量）

将局部变量的值放在 CPU 的寄存器中，这样的变量叫做寄存器变量。

自动变量存储在动态存储区；静态局部变量存储在静态存储区；寄存器变量存储在寄存器。

8.3 全局变量的存储类别

外部变量是在函数的外部定义的全局变量，它的作用域是从变量的定义处开始，到本程序文件的结尾。

（1）在一个文件内扩展外部变量的作用域。

（2）将外部变量的作用域扩展到其他文件。

（3）将外部变量的作用域限制在本文件中。

题 1.以下叙述错误的是（　　）。

A.在函数外可以声明变量。

B.变量声明的位置决定了该变量名的使用范围。

C.函数调用时在函数内声明的变量所得到的值将无法保存到该函数的下一次调用。

D.在函数外声明的变量，其值可以保存到该程序运行结束。

答案：C

题 2.C 语言中形参的默认存储类别是（　　）。

A.自动（auto）　　B.外部（extern）

C.寄存器（register）　　D.静态（static）

答案：A

题 3.执行以下程序后，输出结果是_____。

```
#include<stdio.h>
int fun(int n)
{
    static int a=5;
    a+=n;
    return a;
}
int main()
{
    printf(“%d\n”,fun(2)+fun(3)+fun(4));
    return 0;
}
```

答案：31

课时九 练习题

1.执行下列程序段代码后，屏幕的输出结果是（　　）。

```
#include<stdio.h>
int f(int k)
{
    if(k==0||k==1)
        return 2;
    else
        return k+f(k-1);
}
int main()
{
    printf(“f(%d)=%d\n”,5,f(5));
    return 0;
}
```

A.f(5)=10　　B.f(5)=16

C.f(5)=25　　D.f(5)=21

2.下面程序的运行结果为_____。

```
#include<stdio.h>
bin(int x)
{
    if(x/2>0)
         bin(x/2);
    printf(“%d”,x%2);
}
int main()
{
    bin(13);
    return 0;
}
```

3.当输入为 43 16 时，下面程序运行结果为_____。

```
#include<stdio.h>
int fun(int x,int m)
{
    int a=x%m;
    if(x==0)    return 0;
    fun(x/m,m);
    printf("%c",a<10 ? a+'0' : a-10+'A');
}
int main()
{
    int x,m;
    scanf("%d %d",&x,&m);
    fun(x,m);
    return 0;
}
```

4.若使用一维数组名作函数实参，则以下正确的说法是（　　）。

A.必须在主调函数中说明数组的大小。

B.实参数组类型与形参数组类型可以不匹配。

C.实参数组名必须与形参数组名相一致。

D.在被调函数中，不需要考虑形参数组的大小。

5.下列程序中，函数 int max(int a[],int n)的功能是返回数组 a 中的最大值，在主函数中输入 100 个整数，调用该函数输出其中的最大值，将程序补充完整。

```
#include<stdio.h>
int max(int a[],int n);
int main()
{
    int x[100],i,m;
    printf("输入 100 个整数：\n");
    for(i=0;i<100;i++)
        scanf("%d",&x[i]);
```

```
    _____;
    printf(“max=%d\n”,m);
    return 0;
}
int max(int a[],int n)
{
    int m,i;
    m=_____;
    for(i=1;i<n;i++)
        if(_____)
            m=a[i];
    return m;
}
```

6.C 语言程序内，在函数内部定义的变量称为全局变量。（判断题）

7.凡是在函数中未指定存储类别的局部变量，其隐含的存储类别是（　　）。

A.静态（static）　　B.外部（extern）

C.自动（auto）　　D.寄存器（register）

课时十 指针

考点	重要程度	分值	题型
1. 指针概念	★★	0～2	填空题
2. 指针变量	★★★★	2～6	填空题、读程序题
3. 通过指针引用数组	★★★	2～8	填空题、读程序题
4. 通过指针引用字符串	★★★	2～8	填空题、读程序题

1. 指针概念

内存区的每一个字节有一个编号，这就是“地址”。

由于通过地址能找到所需的变量单元，可以说，地址指向该变量单元。

因此，将地址形象化地称为“指针”，通过它能找到以它为地址的内存单元。

一个变量的地址称为该变量的“指针”。如果有一个变量专门用来存放另一变量的地址（即指针），则称它为“指针变量”。指针变量的值是地址。

2. 指针变量

2.1 定义指针变量

定义指针变量的一般形式为

类型名　*指针变量名;

左边的类型名是在定义指针变量时必须指定的“基类型”。指针变量的基类型用来指定此指针变量可以指向的变量的类型。

注意：

①指针变量前面的“*”表示该变量的类型为指针型变量。

②在定义指针变量时必须指定基类型。

一个变量的指针的含义包括两个方面，一是以存储单元编号表示的地址，一是它指向的存储单元的数据类型。

③指向整型数据的指针类型表示“int *”。

④指针变量中只能存放地址（指针），不要将一个整数赋给一个指针变量。

2.2 引用指针变量

在引用指针变量时，可能有 3 种情况：

（1）给指针变量赋值。如： p=&a;

（2）引用指针变量指向的变量。

如果已执行“p=&a;”，即指针变量 p 指向了整型变量 a，则

```
printf(“%d”,*p);
*p=1;
```

表示将整数 1 赋给 p 当前所指向的变量，如果 p 指向变量 a,则相当于把 1 赋给 a，即“a=1;”。

（3）引用指针变量的值。如：

```
printf(“%o”,p);
```

作用是以八进制数形式输出指针变量 p 的值，如果 p 指向了 a，就是输出了指针 a 的地址，即&a。

注意：要熟练掌握两个有关的运算符：

①&　取地址运算符。 &a 是变量 a 的地址。

②*　指针运算符。　*p 代表指针变量 p 指向的对象。

题 1.若有说明：int *p,m=5,n;以下正确的程序段是（　　）。

A.p=&n; scanf(“%d”,&p);

B.p=&n; scanf(“%d”,*p);

C.scanf(“%d”,&n); *p=n;

D.p=&n; *p=m;

答案：D

题 2.若有定义：int x=1,*p=&x;则语句 printf(“%d\n”,*p);的输出结果为（　　）。

A.1　　B.p 的地址　　C.x 的地址　　D.0

答案：A

2.3 指针变量作为函数参数

注意：

①不能企图通过改变指针形参的值而使指针实参的值改变。

②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。

题 1.输入 a 和 b 两个整数，按先大后小的顺序输出 a 和 b。现用函数处理，而且用指针类型的数据作函数参数。

```
#include<stdio.h>
int main()
{
    void swap(int *p1,int *p2);
    int a,b;
    int *pointer_1,*pointer_2;
    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 temp;
    temp=*p1;
    *p1=*p2;
    *p2=temp;
}
```

3.通过指针引用数组

3.1 数组元素的指针

数组元素的指针就是数组元素的地址。

3.2 在引用数组元素时指针的运算

（1）如果指针变量 p 已指向数组的一个元素，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。注意：执行 p+1 时并不是将 p 的值（地址）简单地加 1，而是加上一个数组元素所占用的字节数。

（2）如果 p 的数值是&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）如果指针变量 p1 和 p2 都指向同一数组，如执行 p2-p1，结果是 p2-p1 的值（两个地址之差）除以数组元素的长度。

注意：两个地址不能相加。

题 1. 设 p 是指向 float 类型一维数组的指针变量，则 p+1 移动的字节数是(　　)。

A. 1　　B. 2　　C. 4　　D. 8

答案：C

题 2. 下列程序的运行结果为_____。

```
#include<stdio.h>
int main()
{
    char s[80]=“Hello”,*p;
    for(p=s+1;*p!= ‘\0’ ;p++)
        printf(“%c”,*p);
    return 0;
}
```

答案：ello

3.3 通过指针引用数组元素

引用一个数组元素，可以用下面两种方法：

（1）下标法，如 a[i]形式；

（2）指针法，如*(a+i)或*(p+i)。其中 a 是数组名，p 是指向数组元素的指针变量，其初值 p=a。

如果不用 p 变化的方法而用数组名 a 变化的方法（例如，用 a++）是不行的。因为数组名 a 代表数组首元素的地址，它是一个指针型常量，它的值在程序运行过程期间是固定不变的。既然 a 是常量，所以 a++是无法实现的。

3.4 用数组名作函数参数

表 1 以变量名和数组名作为函数参数的比较

参数类型	变量名	数组名
要求形参的类型	变量名	数组名或指针变量
传递的信息	变量的值	实参数组首元素的地址
通过函数调用能否改变实参的值	不能改变实参变量的值	能改变实参数组的值

注意：实参数组名代表一个固定的地址，或者说是指针常量，但形参数组名并不是一个固定的地址，而是按指针变量处理。

题 1.下列程序的运行结果为_____。

```
#include<stdio.h>
fun(char *s,int n1,int n2)
{   char c;
    while(n1<n2)
    {   c=s[n1];
        s[n1]=s[n2];
        s[n2]=c;
        n1++;
        n2--;}
}
int main()
{   char a[]=“ABCD”;
    fun(a,0,3);
    printf(“%s\n”,a);
    return 0;
}
```

答案：DCBA

3.5 通过指针引用多维数组

（1）多维数组元素的地址

表 2 二维数组 a 的有关指针

表示形式	含义
a	0 行首地址
a[0],*(a+0),*a	0 行 0 列元素地址
a+1,&a[1]	1 行首地址
a[1],*(a+1)	1 行 0 列元素 a[1][0]的地址
a[1]+2,*(a+1)+2,&a[1][2]	1 行 2 列元素 a[1][2]的地址
(a[1]+2),(*(a+1)+2),a[1][2]	1 行 2 列元素 a[1][2]的值

（2）指向多维数组元素的指针变量

题 1.有一个 3 行 4 列的二维数组，要求用指向元素的指针变量输出二维数组各元素的值。

```
#include<stdio.h>
int main()
{
    int a[3][4]={1,3,5,7,9,11,13,15,17,19,21,23};
    int *p;
    for(p=a[0];p<a[0]+12;p++)
    {
        if((p-a[0])%4==0)    printf(“\n”);
        printf(“%4d”,*p);
    }
    printf(“\n”);
    return 0;
}
```

4.通过指针引用字符串

4.1 字符串的引用方式

在 C 程序中，字符串是存放在字符数组中的。想引用一个字符串，可以用以下两种方法。

（1）用字符数组存放一个字符串，可以通过数组名和下标引用字符串中一个字符，也可以通过数组名和格式声明“%s”输出该字符串。

（2）用字符指针变量指向一个字符串常量，通过字符指针变量引用字符串常量。

分析定义 string 的行：

char *string =“I love China!”;

等价于

char *string;

string =“I love China!”;

可以对指针变量进行再赋值。

4.2 字符指针作函数参数

如果想把一个字符串从一个函数“传递”到另一个函数，可以用地址传递的办法，即用字符数组名作参数，也可以用字符指针变量作参数。在被调用的函数中可以改变字符串的内容，在主调函数中可以引用改变后的字符串。

表 3 调用函数时实参与形参的对应关系

实参	形参	实参	形参
字符数组名	字符数组名	字符指针变量	字符指针变量
字符数组名	字符指针变量	字符指针变量	字符数组名

4.3 使用字符指针变量和字符串的比较

（1）字符数组由若干个元素组成，每个元素中放一个字符，而字符指针变量中存放的是地址（字符串第一个字符的地址），绝不是将字符串放到字符指针变量中。

（2）赋值方式。可以对字符指针变量赋值，但不能对数组名赋值。

（3）初始化的含义。对字符指针变量赋初值：

char *a=“I love China!”;

等价于

char *a;

a=“I love China!”;

而对数组的初始化：

char str[14]=“I love China!”;

不等价于

char str[14];

str[]=“I love China!”;

数组可以在定义时对各元素赋初值，但不能用赋值语句对字符数组中全部元素整体赋值。

（4）存储单元的内容。编译时为字符数组分配若干存储单元，以存放各元素的值，而对指针变量，只分配一个存储单元。

（5）指针变量的值是可以改变的，而数组名代表一个固定的值（数组首元素的地址），不能改变。

（6）字符数组中各元素的值是可以改变的（可以对它们再赋值），但字符指针变量指向的字符串常量中的内容是不可以被取代的（不能对它们再赋值）。

（7）引用数组元素。对字符数组可以用下标法（用数组名和下标）引用一个数组元素（如 a[5]），也可以用地址法（如*(a+5)）引用数组元素 a[5]。如果定义了字符指针变量 p，并使它指向数组 a 的首地址，则可以用指针变量带下标的形式引用数组元素（如 p[5]），同样，可以用地址法（如*(p+5)）引用数组元素 a[5]。

题 1.下面程序的运行结果是_____。

```
#include<stdio.h>
#include<string.h>
int main()
{
    char *str=“Language”;
    printf(“%d\n”,strlen(str));
    printf(“%c,%s\n”,*(str+2),str+3);
    return 0;
}
```

答案：8

n,guage

题 2.将一串数字字符串中各数字求和，并输出。如“2019”，求和为 12。请填空。

```
#include<stdio.h>
main()
{
    char *str=“2019”,*pa;
    int s=0;
    _____;
    while(_____)
    {
        _____;
        pa++;
    }
    printf(“%d\n”,s);
    return 0;
}
```

答案：pa=str

*pa!=‘\0’

s=s+*pa-‘0’

课时十 练习题

1.已知 int x=10,*p=&x;则下列表达式值为 10 的是（　　）。

A.p　　B.*p　　C.&p　　D.&x

2.若有说明语句：int *p,a;则能通过 scanf 语句正确给输入项，读入数据的程序段的是（　　）。

A.*p=&a;　scanf(“%d”,p);

B.p=&a;　scanf(“%d”,p);

C.*p=&a;　scanf(“%d”,p);

D.p=&a;　scanf(“%d”,*p);

3.下面程序的运行结果是_____。

```
#include<stdio.h>
void swap(int *x,int *y)
{
    int temp;
    temp=*x;
    *x=*y;
    *y=temp;
}
int main()
{
    int i,a[5]={1,2,3,4,5};
    for(i=0;i<5/2;i++)
       swap(&a[i],&a[4-i]);
    for(i=0;i<5;i++)
       printf(“%d”,a[i]);
    return 0;
}
```

4.对于基类型相同的两个指针变量 p1、p2，则下列运算不合理的是（　　）。

A.p1=p2　　B.p1<p2　　C.p1+p2　　D.p2-p1

5.设有语句int a[]={1,3,5,7};int *p=&a[0];那么*(p+2)的值为_____。

6.有以下程序：

```
#include<stdio.h>
void f(int *x,int *y)
{
    int t;
    t=*x;
    *x=*y;
    *y=t;
}
main()
{
    int a[8]={8,7,6,5,4,3,2,1},i,*p,*q;
    p=a;
    q=&a[7];
    while(p<q)
    {
        f(p,q);
        p++;
        q--;
    }
    for(i=0;i<8;i++)
        printf("%d ",a[i]);
}
```

程序运行后的输出结果是（　　）。

A.8 2 3 4 5 6 7 1　　B.8 7 6 5 4 3 2 1

C.5 6 7 8 1 2 3 4　　D.1 2 3 4 5 6 7 8

7.设有定义：char *s="\t\"Name\\Address\"\n"，则strlen(s)的值为_____。

8.下列程序的功能是求字符串的长度，请将下面的程序补充完整。

```
#include<stdio.h>
#include<stdlib.h>
int strlength(char *str)  //求字符串的长度
{
    int len;
    len=0 ;
    while(_____)
    {
       len++;
       str++;
    }
    _____;
}
int main()
{
    char str[100];
    gets(str);
    puts(str);
    printf("%d\n",_____);
    return 0;
}
```

课时十一 用户自己建立数据类型

考点	重要程度	分值	题型
1.定义和使用结构体变量	★★★★	2～6	选择题、读程序题
2.使用结构体数组	★★★	0～4	选择题、读程序题
3.结构体指针	★★★	0～4	选择题、读程序题

1.定义和使用结构体变量

1.1 自己建立结构体类型

C 语言允许用户自己建立由不同类型数据组成的组合型的数据结构，它称为结构体。例如：

```
struct Student
{
    int num;
    char name[20];
    char sex[2];
    int age;
    float score;
    char addr[30];
};                                  //注意最后有个分号
```

声明一个结构体类型的一般形式为

struct 结构体名

{成员表列};

注意：

结构体类型的名字是由一个关键字 struct 和结构体名组合而成的。结构体名是由用户指定的，又称“结构体标记”，以区别于其他结构体类型。

花括号内是该结构体所包含的子项，称为结构体的成员。

对各成员都应进行类型声明，即

类型名 成员名;

成员名命名规则与变量名相同。

说明：成员可以属于另一个结构类型。

1.2 定义结构体类型变量

（1）先声明结构体类型，再定义该类型的变量

在定义了结构体变量后，系统会为之分配内存单元。

（2）在声明类型的同时定义变量

其一般形式为

struct 结构体名

{

成员表列

}变量名表列;

（3）不指定类型名而直接定义结构体类型变量

其一般形式为

struct

{

成员表列

}变量名表列;

指定了一个无名的结构体类型，不出现结构体名。显然不能再以此结构体类型去定义其他变量。

说明：

①结构体类型与结构体变量是不同的概念。只能对变量赋值、存取或运算，而不能对一个类型赋值、存取或运算。

②结构体类型中的成员名可以与程序的变量名相同，但二者不代表同一对象。

③对结构体变量中的成员，可以单独使用，它的作用与地位相当于普通变量。

1.3 结构体变量的初始化和引用

（1）在定义结构体变量时，可以对它的成员初始化。初始化列表是用花括号括起来的一些常量，这些常量一次赋给结构体变量的各成员。注意：是对结构体变量初始化，而不是对结构体类型初始化。

（2）可以引用结构体变量中成员的值，引用为

结构体变量.成员名

“.”是成员运算符，它在所有的运算符中优先级最高。

注意：不能企图输出结构体变量名来达到输出结构体变量所有成员的值。只能对结构体变量中的各个成员分别进行输入和输出。

（3）如果成员本身又属一个结构体类型，则要用若干个成员运算符，一级一级地找到最低的一级的成员。只能对最低级的成员进行赋值或存取以及运算。

（4）对结构体变量的成员可以像普通变量一样进行各种运算。

（5）同类的结构体变量可以相互赋值。

（6）可以引用结构体变量成员的地址，也可以引用结构体变量的地址。

题 1.如有以下定义：

```
struct
{
    float k;
    char c[4];
}r;
```

则 r 分配的字节数是（　　）。

A.4　　B.1　　C.8　　D.0

答案：C

题 2.已有定义：

```
struct stu;
{ int x; float y; char z; }student;
```

struct stu 是结构体类型是正确的。（判断题）

答案：错误

2.使用结构体数组

2.1 定义结构体数组

（1）定义结构体数组的一般形式是

①struct 结构体名

{成员表列} 数组名[数组长度];

②先声明一个结构体类型，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];

（2）对结构体数组初始化的形式是在定义数组的后面加上：

={初值表列};

题 1.根据下面的定义，能打印出字母 M 的语句是（　　）。

```
struct person{ char name[9]; int age; };
struct person class[10]={“John”,17,“Paul”,19,“Mary”,18,“Adam”,20};
```

A.printf(“%c”,class[3].name[0]);

B.printf(“%c”,class[3].name[1]);

C.printf(“%c”,class[2].name[1]);

D.printf(“%c”,class[2].name[0]);

答案：D

题 2.利用结构体数组存放 5 个学生的学号和成绩，计算平均成绩并显示输出。

```
#include<stdio.h>
struct student
{
   int num;
   float score;
};
int main()
{
    struct student _____;
    float ave=0;
    int i;
    for(i=0;i<5;i++)
    {
        scanf(“%d%f”,&s[i].num,&s[i].score);
        ave+=_____;
    }
    printf(“ave=%.3f”,ave/5);
    return 0;
}
```

答案：s[5]

　　s[i].score

题 3.有三个候选人，每个选民只能投票选一人，要求编一个统计票数的程序，先后输入候选人的名字，最后输出各人得票结果。

```
#include<stdio.h>
#include<stdio.h>
struct Person
{
    char name[20];
    int count;
} leader[3]={“Li”,0,“Zhang”,0,“Sun”,0};
int main()
{
    int i,j;
    char leader_name[20];
    for(i=1;i<=10;i++)
    {
        scanf(“%s”,leader_name);
        for(j=0;j<3;j++)
            if(strcmp(leader_name,leader[j].name)==0)
                leader[j].name++;
    }
    printf(“\nResult:\n”);
    for(i=0;i<3;i++)
        printf(“%5s:%d\n”,leader[i].name,leader[i].count);
    return 0;
}
```

3.结构体指针

3.1 指向结构体变量的指针

指针变量的基类型必须与结构体变量的类型相同。

说明：

为了使用方便和直观，C 语言允许把(*p).num 用 p->num 来代替，“->”代表一个箭头，p->num 表示 p 所指向的结构体变量中 num 成员。同样，(*p).name 等价于 p->name，”->”称为指向运算符。

如果 p 指向一个结构体变量 stu，以下 3 种方法等价：

①stu.成员名

②(*p).成员名

③p->成员名

3.2 指向结构体数组的指针

可以用指针变量指向结构体数组的元素。

p+1 意味着 p 所增加的值为结构体数组一个元素所占的字节数。

题 1.对于以下的变量定义，表达式（　　）不符合 C 语言语法。

```
struct node
{
    int len;
    char *pk;
}x={2,“right”},*p=&x;
```

A.p->pk　　B.(*p).pk

C.*p->pk　　D.x.pk

答案：C

课时十一 练习题

1.当定义一个结构体变量时，系统分配给它的内存是（　　）。

A.各成员所需内存的总和

B.结构体中第一个成员所需的内存量

C.成员中占内存量最大者所需的容量

D.结构体中最后一个成员所需内存量

2.已知学生纪录描述为

```
struct Birthday
{
    int year;
    int month;
    int day;
}
struct Birth
{
    struct Birthday birth;
}s;
```

设变量 s 中的“生日”应为“1984 年 11 月 11 日”。正确对生日赋值的是（　　）。

A.year=1984; month=11; day=11;

B.s.birth.year=1984; s.birth.month=11; s.birth.day=11;

C.s.year=1984; s.month=11; s.day=11;

D.birth.year=1984; birth,month=11; birth.day=11;

3.下列程序的运行结果为_____。

```
#include<stdio.h>
#define N 4
struct stud
{
    char num[10];
```

```
    int score[2];
};
int main()
{
    struct stud stu[N]={{“201701”,89,50},{“201702”,87,80},
                        {“201703”,98,89},{“201704”,90,100}};
    int i,j,sum=0,avg;
    for(i=0;i<N;i++)
        sum+=stu[i].score[1];
    avg=sum/N;
    printf(“%d,%d\n”,sum,avg);
    return 0;
}
```

4.对候选人得票的统计程序。设有 3 个候选人，每次输入一个得票的候选人姓名，要求最后输出各人的得票结果，请填空。

```
#include<stdio.h>
#include<stdio.h>
struct Person
{
    char name[20];
    int count;
} leader[3]={“Li”,0,“Zhang”,0,“Sun”,0};
int main()
{
    int i,j;
    char leader_name[20];
    for(i=1;i<=10;i++)
    {
        scanf(“%s”,__________);    //输入所选的名字
        for(j=0;j<3;j++)
            if(__________)          //所选名字与候选人名字比较
                leader[j].name++;
    }
    printf(“\nResult:\n”);
    for(i=0;i<3;i++)
```

```
        printf("%5s:%d\n",leader[i].name,leader[i].count);
    return 0;
}
```

5.以下对结构变量st1中成员score的非法引用的是（　　）。

```
struct Student
{
    char *no;
    int score;
}st1,*p;p=&st1;
```

A.p.score　　B.(*p).score

C.p->score　　D.st1.score

6.以下程序的输出结果是_____。

```
#include<stdio.h>
struct Stu
{
    int x;
    int *y;
}*p;
int dt[4]={10,20,30,40};
struct Stu a[4]={50,&dt[0],60,&dt[1],70,&dt[2],80,&dt[3]};
int main()
{
    p=a;
    printf("%d,",++p->x);
    printf("%d,",(++p)->x);
    printf("%d\n",++*(p->y));
    return 0;
}
```

课时十二 对文件的输入输出

考点	重要程度	分值	题型
1.C 文件的有关基本知识	★★	0～4	选择题
2.打开与关闭文件	★★★	0～4	选择题、填空题
3.顺序读写数据文件	★★★	0～4	选择题、程序填空题

1.C 文件的有关基本知识

1.1 什么是文件

文件有不同的类型，在程序设计时，主要用到两种文件：

（1）程序文件。包括源程序文件（后缀为.c）、目标文件（后缀.obj）、可执行文件（后缀.exe）等。这种文件的内容是程序代码。

（2）数据文件。文件的内容不是程序，而是供程序运行时读写的数据。

1.2 文件名

一个文件要有一个唯一的文件标识，以便用户识别和引用。

文件标识包括 3 部分：（1）文件路径

（2）文件名主干

（3）文件后缀

文件路径表示文件在外部存储设备中的位置。

1.3 文件的分类

根据数据的组织形式，数据文件可分为 ASCII 文件和二进制文件。

二进制文件：数据在内存中是以二进制形式存储的，如果不加转换地输出到外存，就是二进制文件。

ASCII 文件：如果要求在外存上以 ASCII 代码形式存储，则需要在存储前进行转换，就是 ASCII 文件，ASCII 文件又称文本文件。

1.4 文件类型指针

每个被使用的文件都在内存中开辟一个相应的文件信息区，用来存放文件的有关信息（如文件的名字、文件状态及文件当前位置等）。这些信息是存放在一个结构体变量中的。该结构体类型是由系统声明的，取名为 FILE。

一般不对 FILE 类型变量命名，也就是不通过变量的名字来引用这些变量，而是设置一个指向 FILE 类型变量的指针变量，然后通过它来引用这些 FILE 类型变量。

```
FILE *fp;
```

题 1.C 语言中系统默认的文件类型有文本文件和二进制文件。（判断题）

答案：正确

2.打开与关闭文件

2.1 用 fopen 函数打开数据文件

fopen 函数的调用方式为

```
fopen(文件名，使用文件方式);
```

表 1 使用文件方式

文件使用方式	含义	文件不存在
“r”（只读）	为了输入数据，打开一个已存在的文本文件	出错
“w”（只读）	为了输出数据，打开一个文本文件	建立新文件
“a”（追加）	向文本文件尾添加数据	出错
“rb”（只读）	为了输入数据，打开一个二进制文件	出错
“wb”（只写）	为了输出数据，打开一个二进制文件	建立新文件
“ab”（追加）	向二进制文件尾添加数据	出错
“r+”（读写）	为了读和写，打开一个文本文件	出错
“w+”（读写）	为了读和写，建立一个新的文本文件	建立新文件
“a+”（读写）	为了读和写，打开一个文本文件	出错
“rb+”（读写）	为了读和写，打开一个二进制文件	出错
“wb+”（读写）	为了读和写，建立一新的二进制文件	建立新文件
“ab+”（读写）	为读写打开一个二进制文件	出错

如果不能实现“打开”的任务，fopen 函数将会带回一个出错信息。此时 fopen 函数将带回一个空指针值 NULL。

常用下面的方法打开一个文件：

```
if((fp=fopen(“file1”,“r”))==NULL)
{
    printf(“cannot open this file\n”);
    exit(0);
}
```

2.2 用 fclose 函数关闭数据文件

“关闭”就是撤销文件信息区和文件缓冲区，使文件指针变量不再指向该文件，也就是文件指针变量与文件“脱钩”，此后不能再通过该指针对原来与其相联系的文件进行读写操作。

关闭文件用 fclose 函数。fclose 函数调用的一般形式为

fclose(文件指针);

fclose 函数也带回一个值，当成功地执行了关闭操作，则返回值为 0；否则返回 EOF(-1)。

题 1.定义 FILE *fp;以写方式打开文件 C:\aa.dat 的正确语句是（　　）。

A.fp=fopen(“C:\aa.dat”,“w”);

B.fp=fopen(“C:\aa.dat”,“r”);

C.fp=fopen(“C:\\aa.dat”,“w”);

D.fp=fopen(“C:\\aa.dat”,“r”);

答案：C

题 2.若 fp 已正确定义为一个文件指针，d.txt 为 C 盘根目录下的文本文件。请填空，以便为“读”而打开此文件：

fp=fopen(“_____”,“_____”);

答案：C:\\d.txt,r

3.顺序读写数据文件

3.1 向文件读写一个字符

表 2 读写一个字符的函数

函数名	调用形式	功能	返回值
fgetc	fgetc(fp)	从 fp 指向的文件读入一个字符	读成功，带回所读的字符，失败则返回文件结束标志 EOF(即-1)
fputc	fputc(ch,fp)	把字符 ch 写到文件指针变量 fp 所指向的文件中	输出成功，返回值就是输出的字符;输出失败,则返回 EOF(即-1)

3.2 向文件读写一个字符串

表 3 读写一个字符串的函数

函数名	调用形式	功能	返回值
fgets	fgets(str,n,fp)	从 fp 指向的文件读入一个长度为（n-1）的字符串，存放到字符数组 str 中。	读成功，返回地址 str，失败则返回 NULL
fputs	fputs(str,fp)	把 str 所指向的字符串写到文件指针变量 fp 所指向的文件中	输出成功，返回 0；输出失败，否则返回非 0 值

3.3 用格式化的方式读写文件

一般的调用方式为：

fprintf(文件指针,格式字符串,输出列表);

fscanf(文件指针,格式字符串,输入列表);

3.4 用二进制方式向文件读写一组数据

C 语言允许用 fread 函数从文件中读一个数据块，用 fwrite 函数向文件写一个数据块。在读写时是以二进制形式进行的。再想磁盘写数据时，直接将内存中一组数据原封不动、不加转换地复制到磁盘文件上，在读入也是将磁盘文件中若干字节的内容一批读入内存。

它们的一般调用形式为

fread(buffer,size,count,fp);

fwrite(buffer,size,count,fp);

题 1.C 语言中，调用_____函数来检测文件位置指针有没有到文件尾。

答案：feof

题 2.以下叙述中错误的是（　　）。

A.gets 函数用于从键盘输入字符串。

B.getchar 函数用于从文件读入字符。

C.fputs 函数用于把字符串输出到文件。

D.fwrite 函数用于以二进制形式输出数据到文件。

答案：B

题 3.将整型数组 a 中各元素存入文件 data.txt 中。（改错题）

```
#include<stdio.h>
int main()
{
    int a[]={1,2,3,4,5,6,7,8};
    FILE *fp;
    fp=fopen(“E:\data.txt”,w);
    if(fp!=NULL)
    {
        for(i=0;i<8;i++)
            fprintf(“%d”,a[i]);
    }
    fclose(fp);
}
```

答案：fp=fopen(“E:\\data.txt”,”w”);

fwrite(&a[i],sizeof(int),1,fp);

课时十二 练习题

1.C 语言中的文件类型有（　　）。

A.索引文件和文本文件两种

B.二进制文件一种

C.文本文件一种

D.ASCII 文件和二进制文件两种

2.若要打开名为 abc.txt 的文本文件进行读、写操作，下面符合此要求的函数调用的是（　　）。

A.fopen(“abc.txt”,“r”);

B.fopen(“abc.txt”,“r+”);

C.fopen(“abc.txt”,“rb”);

D.fopen(“abc.txt”,“w”);

3.若执行 fopen()函数时发生错误，则函数的返回值是（　　）。

A.地址值　　B.NULL

C.1　　D.EOF

4.fscanf 函数的正确调用形式是（　　）。

A.fscanf(fp,格式控制符,输出列表);

B.fscanf(格式控制符,输出列表,fp);

C.fscanf(格式控制符,文件指针,输出列表);

D.fscanf(文件指针,格式控制符,输入列表);

5.在向文件写时，下列哪个函数名是写单字符函数（　　）。

A.fread　　B.fgetc　　C.fputc　　D.fwrite

6.在 C 语言中，对文件操作的一般步骤是（　　）。

A.打开文件，读文件，写文件，关闭文件

B.定义文件指针，读文件，写文件，关闭文件

C.定义文件指针，打开文件，读写文件，关闭文件

D.操作文件，定义文件指针，修改文件，关闭文件

7.以下程序的功能是，从键盘输入一组字符，将字符本身及字符 ASCII 码写入 D 盘中 test.txt 文本文件保存，输入字符以#结束。

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    __________;
    char ch;
    fp=fopen(“D:\\test.txt”,“__________”);
    while((ch=getchar())!= ‘#’ )
        fprintf(fp,“%c %d”,__________);
    fclose(fp);
    return 0;
}
```