1. **程序设计与C语言**

**1.1 选择题**

考点：标识符的命名规则

1. 只能由字母、数字、下划线构成
2. 数字不能作为标识符的开头
3. 关键字不能作为标识符

选项A中的“-” ，选项B中“[”与“]”不满足（1）；选项D中的int为关键字，不满足（3）

1.下列正确的标识符是（C　）。

A.-a1 B.a[i] C.a2\_i D.int t

2.下列C语言用户标识符中合法的是（ B ）。

A)3ax B)x C)case D)-e2 E)union

选项A中的标识符以数字开头不满足（2）；选项C，E均为为关键字，不满足（3）；选项D中的“-”不满足（1）；

3.下列四组选项中，正确的C语言标识符是（ C ）。

A） %x B） a+b C） a123 D） 123

选项A中的“%” ，选项B中“+”不满足（1）；选项D中的标识符以数字开头不满足（2）

4.下列四组字符串中都可以用作C语言程序中的标识符的是（ A ）。

A、print \_3d db8 aBc B、I\am one\_half start$it 3pai

C、str\_1 Cpp pow while D、Pxq My->book line# His.age

选项B中的“\”，”$” ，选项D中“>”，”#”，”.”，”-”不满足（1）；选项C中的while为关键字，不满足（3）

知识点：简单C语言程序的构成和格式

5.在一个C程序中\_\_\_B\_\_\_。

A、main函数必须出现在所有函数之前 B、main函数可以在任何地方出现

C、main函数必须出现在所有函数之后 D、main函数必须出现在固定位置

6.以下说法中正确的是\_\_D\_\_\_\_。

A、#define和printf都是C语句 B、#define是C语句，而printf不是

C、printf是C语句，但#define不是 D、#define和printf都不是C语句

7.一个C程序的执行是从（ A ）。

A本程序的main函数开始，到main函数结束

B本程序文件的第一个函数开始，到本程序文件的最后一个函数结束

C本程序的main函数开始，到本程序文件的最后一个函数结束

D本程序文件的第一个函数开始，到本程序main函数结束

8.在 C 语言中，每个语句必须以（ D ）结束。

A. 回车符 B. 冒号 C. 逗号 D. 分号

9.C 语言规定：在一个源程序中，main函数的位置（ C ）。

A. 必须在最开始 B. 必须在系统调用的库函数的后面

C. 可以任意 D. 必须在最后

10.一个C 语言程序是由（ B ）。

A. 一个主程序和若干子程序组成 B. 函数组成

C. 若干过程组成 D. 若干子程序组成

11.下列说法中错误的是( D )。

A. 主函数可以分为两个部分：主函数说明部分和主函数体

B. 主函数可以调用任何非主函数的其他函数

C. 任何非主函数可以调用其他任何非主函数

D. 程序可以从任何非主函数开始执行

12.用 C 语言编写的源文件经过编译，若没有产生编译错误，则系统将( C )。

A. 生成可执行目标文件 B. 生成目标文件

C. 输出运行结果 D. 自动保存源文件

**1.2 填空题**

1、C 语言只有 32 个关键字和 9 种控制语句。

2、每个源程序有且只有一个 main 函数，系统总是从该函数开始执行C语言程序。

3、C 语言程序的注释可以出现在程序中的任何地方，它总是以 \\* 符号作为开始标记，以 \*/ 符号作为结束标记。

4、C 语言中，输入操作是由库函数scanf完成的，输出操作是由库函数 printf 完成的。

5、系统默认的C 语言源程序文件的扩展名是 .c ，经过编译后生成的目标文件的扩展名是 .obj ，经过连接后生成的可执行文件的扩展名是 .exe 。

6、C 语言的标识符只能由字母、数字和下划线三种字符组成。

1. **CH2 数据的存储与运算**

**2.1 选择题**

1.C语言中的简单数据类型包括（D ）。

A、整型、实型、逻辑型 B、整型、实型、逻辑型、字符型

C、整型、字符型、逻辑型 D、整型、实型、字符型

2.在C语言程序中，表达式5%2的结果是 C 。

A)2.5 B)2 C)1 D)3

%为求余运算符，该运算符只能对整型数据进行运算。且符号与被模数相同。5%2=1； 5%（-2）=1；（-5）%2=-1；（-5）%（-2）=-1；

/为求商运算符，该运算符能够对整型、字符、浮点等类型的数据进行运算，5/2=2

3.设int a=12，则执行完语句a+=a-=a\*a后，a的值是\_\_D\_\_\_\_。

A、552 B、264 C、144 D、-264

4.以下变量x、y、z均为double类型且已正确赋值，不能正确表示数字式子x/(y\*z)的C语言表达式是\_\_\_A\_\_\_。

A、x/y\*z B、x\*(1/(y\*z)) C、x/y\*1/z D、x/y/z

5.若有定义：int a=8,b=5,c;,执行语句c=a/b+0.4;后，c的值是\_\_B\_\_\_\_。

A、1.4 B、1 C、2.0 D、2

6.以下选项中，不能作为合法常量的是\_\_B\_\_\_\_。

A、1.234e04 B、1.234e0.4 C、1.234e+4 D、1.234e0

7.以下符合C语言语法的实型常量是\_\_\_C\_\_\_。

A、1.2E0.5 B、3.14159E C、.5E-3 D、E15

8.表达式3.6-5/2+1.2+5%2的值是\_\_D\_\_\_\_。

A、4.3 B、4.8 C、3.3 D、3.8

9.下列定义变量的语句中错误的是\_\_D\_\_\_\_。

A、int \_int B、double int \_ C、char For D、float US$

10.以下不合法的用户标识符是\_\_C\_\_\_\_。

A、j2\_KEY B、Double C、4d D、\_8\_

11.以下4组用户定义标识符中，全部合法的一组是\_\_A\_\_\_\_。

A、\_main enclude sin B、If -max turbo

C、txt REAL 3COM D、int k\_2 \_001 ???

12.以下定义语句中正确的是\_\_C\_\_\_\_。

A、char a='A' b='B'; B、float a=b=10.0;(正解a=10.0 b=10.0)

C、int a=10,\*b=&a; D、float \*a,b=&a;？？？

13.以下不能定义为用户标识符的是\_\_D\_\_\_\_。

A、scanf B、Void C、\_3com\_ D、int

14.以下有4组用户标识符，其中合法的一组是\_\_\_C\_\_\_。

A、For -sub Case B、4d DO Size C、f2\_G3 IF abc D、WORD Void define

15.C语言中最简单的数据类型包括\_B\_\_\_\_\_。

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

16.下列选项中，合法的C语言关键字是\_\_D\_\_\_\_。

A、VAR B、cher C、integer D、default

17.以下不能定义为用户标识符的是\_\_D\_\_\_\_。

A、Main B、\_0 C、\_int D、sizeof

18下列叙述中正确的是\_D\_\_\_\_\_。

A、C语言中既有逻辑类型也有集合类型 B、C语言中没有逻辑类型但有集合类型

C、C语言中有逻辑类型但没有集合类型 D、C语言中既没有逻辑类型也没有集合类型

19.设int a=12，则执行完语句a+=a-=a\*a后，a的值是\_\_D\_\_\_\_。

A、552 B、264 C、144 D、-264

20.以下变量x、y、z均为double类型且已正确赋值，不能正确表示数字式子x/(y\*z)的C语言表达式是\_\_\_A\_\_\_。

A、x/y\*z B、x\*(1/(y\*z)) C、x/y\*1/z D、x/y/z

21.若有定义：int a=8,b=5,c;,执行语句c=a/b+0.4;后，c的值是\_\_B\_\_\_\_。

A、1.4 B、1 C、2.0 D、2

22.设有定义：float a=2,b=4,h=3;,以下C语言表达式与代数式 1/2((a+b)h) 计算结果不相符的是\_\_\_B\_\_\_。

A、(a+b)\*h/2 B、(1/2)\*(a+b)\*h C、(a+b)\*h\*1/2 D、h/2\*(a+b)

23.假定x和y为double型，表达式x=2,y=x+3/2的值是\_\_\_D\_\_\_。

A、3.500000 B、3 C、2.000000 D、3.000000

24.若已定义x和y为double类型，则表达式x=1,y=x+3/2的值是\_\_\_C\_\_\_。

A、1 B、2 C、2.000000 D、2.500000

25.若有以下程序段

int c1=1,c2=2,c3;

c3=1.0/c2\*c1;

则执行后，c3中的值是\_\_\_A\_\_\_。

A、0 B、0.5 C、1 D、2

26.设有int x=11;则表达式（x++ \* 1/3）的值是\_\_\_A\_\_\_。？？

A、3 B、4 C、11 D、12

27.下列程序的输出结果是\_\_\_C\_\_\_。

main()

{

double d=3.2; int x,y;

x=1.2;y=(x+3.8)/5.0;

printf("%d \n",d\*y);

}

A、3 B、3.2 C、0 D、3.07

28.请读程序片段：

int i=65536;

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

上面程序片段的输出结果是\_\_\_B\_\_\_。

A、65536 B、0 C、有语法错误，无输出结果 D、-1

29.若有以下定义和语句:

int u=010,v=0x10,w=10;

printf("%d,%d,%d\n",u,v,w);

则输出结果是: \_\_A\_\_\_\_。

A、8,16,10 B、10,10,10 C、8,8,10 D、8,10,10

30.在C语言中，不正确的int类型的常数是\_\_\_A\_\_\_。

A、32768 B、0 C、037 D、0xAF

31.以下程序的输出结果是\_\_D\_\_\_\_。

main( )

{

int k=17;

printf("%d,%o,%x \n",k,k,k);

}

A、17,021,0x11 B、17,17,17 C、17,0x11,021 D、17,21,11

32.下面程序的输出是\_\_C\_\_\_\_。

main()

{

unsigned a=32768;

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

}

A、a=32768 B、a=32767C、a=-32768 D、a=-1

33.在C语言中，合法的长整型常数是\_\_\_A\_\_\_。

A、0L B、4962710 C、324562& D、216D

34.C语言中运算对象必须是整型的运算符是\_\_A\_\_\_\_。

A、%= B、/ C、= D、<=

35.以下选项中不属于C语言的类型名称是\_\_\_D\_\_\_。

A、signed short int B、unsigned long int C、unsigned int D、long short

36.有以下程序

main()

{

int m=0256,n=256;

printf("%o %o\n",m,n);

}

程序运行后的输出结果是\_\_\_C\_\_\_。

A、0256 0400 B、0256 256 C、256 400 D、400 400

37.若执下面程序时从键盘上输入5，

main()

{

int x;

scanf("%d",&x);

if(x++>5) printf("%d\n",x);

else printf("%d\n",x--);

}

则输出是\_\_B\_\_\_\_。

A、7 B、6 C、5 D、4

38.有以下程序

main ()

{

char a1='M', a2='m';

printf("%c\n", (a1,a2));

}

以下叙述中正确的是\_\_\_B\_\_\_。

A、程序输出大写字母M B、程序输出小写字母m

C、格式说明符不足，编译出错 D、程序运行时产生出错信息

39.以下程序的输出结果是 \_\_B\_\_\_\_。

main()

{

int a;

printf("%d\n",(a=3\*5,a\*4,a+5));

}

A、65 B、20 C、15 D、10

40.设有如下程序段

int x=2002,y=2003;

printf("%d\n",(x,y));

则以下叙述中正确的是\_\_D\_\_\_\_。

A、输出语句中格式说明符的个数少于输出项的个数，不能正确输出

B、运行时产生错误信息

C、输出值为2002

D、输出值为2003

41有以下程序

main()

{

int x,y,z;

x=y=1;

z=x++,y++,++y;

printf("%d,%d,%d\n",x,y,z);

}

程序运行后的输出结果是\_\_\_C\_\_\_。

A、2,3,3 B、2,3,2 C、2,3,1 D、2,2,1

42.设有以下定义

int a=0;

double b=1.25;

char c='A';

#define d 2

则下面语句中错误的是\_\_D\_\_\_\_。

A、a++; B、b++; C、c++; D、d++;

43.以下程序的输出结果是\_\_\_D\_\_\_。

main()

{

int x=10,y=10;

printf("%d %d\n",x--,--y);

}

A、10 10 B、99 C、9 10 D、10 9

44.若x和y都是int型变量，x=100、y=200，且有下面的程序片段

printf("%d",(x,y));

上面程序片段的输出结果是\_\_\_A\_\_\_。

A、200 B、100 C、100 200 D、输则格式符不够, 输出不确定的值

45.设有如下定义:

int x=10,y=3,z;

则语句

printf("%d\n",z=(x%y,x/y));

的输出结果是\_\_\_D\_\_\_。

A、1 B、0 C、4 D、3

46.两次运行下面的程序,如果从键盘上分别输入6和4,则输出结果是\_\_\_A\_\_\_。

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 D、6和4

47.若t为double类型，表达式t=1,t+5,t++的值是\_\_\_D\_\_\_。

A、1 B、6.0 C、2.0 D、1.0

48.以下程序的输出结果是\_\_A\_\_\_\_。

#include<stdio.h>

main()

{

int i=10,j=10;

printf("%d,%d\n",++i,j--);

}

A、11,10 B、9,10 C、010,9 D、10,9

49.有以下程序

main()

{

int y=10;

while(y--);

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

}

程序执行后的输出结果是\_\_B\_\_\_。

A、y=0 B、y=-1 C、y=1 D、while构成无限循环

50.下列程序的输出结果是（ B）。

main( )

{ int a=7,b=5;

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

}

A、 5 B、 1 C、 0 D、不确定值

51.以下选项中，不正确的 C 语言浮点型常量是（ C ）。

A．160 B. 0.12 C. 2e4.2 D. 0.0

52.以下选项中，（ D ）是不正确的 C 语言字符型常量。

A. 'a' B. '\x41' C. '\101' D. "a"

53.在 C 语言中，字符型数据在计算机内存中，以字符的（ C ）形式存储。

A. 原码 B. 反码 C. ASCII 码 D. BCD码

54.若x、i、j和k都是int型变量，则计算下面表达式后，x的值是（ C ）。

x=（i=4，j=16，k=32）

A. 4 B. 16 C.32 D.52

55.算术运算符、赋值运算符和关系运算符的运算优先级按从高到低依次为( B )。

A. 算术运算、赋值运算、关系运算 B. 算术运算、关系运算、赋值运算

C. 关系运算、赋值运算、算术运算 D. 关系运算、算术运算、赋值运算

56.若有代数式 ，则不正确的C语言表达式是（ C ）。

A.a/b/c\*e\*3 B. 3\*a\*e/b/c

C.3\*a\*e/b\*c D. a\*e/c/b\*3

57.表达式!x||a==b 等效于( D )。

A. !((x||a)==b) B. !(x||y)==b

C. !(x||(a==b)) D. (!x)||(a==b)

58.设整型变量 m,n,a,b,c,d 均为1，执行 (m=a>b)&&(n=c>d)后, m,n 的值是（ A）。

A. 0，0 B. 0，1 C. 1，0 D. 1，1

59.设有语句 int a=3；，则执行了语句 a+=a-=a\*=a;后，变量 a 的值是( B )。

A. 3 B. 0 C. 9 D. -12

60.在以下一组运算符中，优先级最低的运算符是( D )。

A. \* B. != C. + D. =

61.设整型变量 i 值为2，表达式(++i)+(++i)+(++i)的结果是( B )。

A. 6 B. 12 C. 15 D. 表达式出错

62.若已定义 x 和 y为double 类型，则表达式 x=1，y=x+3/2 的值是( C )。

A. 1 B. 2 C. 2.0 D. 2.5

63.sizeof (double)的结果值是( A )。

A. 8 B. 4 C. 2 D. 出错

64.设a=1，b=2，c=3，d=4，则表达式：a<b? a : c<d? a : d的结果为( D )。

A. 4 B. 3 C. 2 D. 1

65.设a 为整型变量，不能正确表达数学关系：10<a<15的 C 语言表达式是( A )。

A. 10<a<15 B. a= =11|| a= =12 || a= =13 || a= =14

C. a>10 && a<15 D. !(a<=10) && !(a>=15)

66.设 f是实型变量，下列表达式中不是逗号表达式的是( D )。

A. f= 3.2, 1.0 B. f>0, f<10

C. f=2.0, f>0 D. f=(3.2, 1.0)

67.表达式18/4\*sqrt(4.0)/8值的数据类型是（ C ）。

A. int B. float C. double D. 不确定

68.已知字母A的ASCII码为十进制数65，且c2为字符型，则执行语句

C2=’A’+’6’-‘3’;后c2中的值是（ A ）。

A. D B. 68 C. 不确定的值 D. C

69.以下用户标识符中，合法的是（ B ）。

A. int B. nit C. 123 D. a+b

70.C 语言中，要求运算对象只能为整数的运算符是（ A ）。

A. % B. / C. > D. \*

71.若有说明语句：char c='\72'；则变量c在内存占用的字节数是（ A ）。

A. 1 B. 2 C. 3 D. 4

72.字符串"ABC"在内存占用的字节数是（ B ）。

A. 3 B. 4 C. 6 D. 8

73.要为字符型变量 a赋初值，下列语句中哪一个是正确的（ B ）。

A. char a="3"; B. char a=’3’;

C. char a=%; D. char a=\*;

74.下列不正确的转义字符是（ C ）。

A. \\ B. \' C. 074 D. \0

**2.2 填空题**

1、C 语言中的逻辑值“真”是用 1 表示的，逻辑值“假”是用 0 表示的。

2、若x和n都是int型变量，且x的初值为12，n的初值为5，则计算表达式x%=(n%=2)后x的值为 0 。

3、设 c='w',a=1,b=2,d=-5, 则表达式 'x'+1>c, 'y'!=c+2, -a-5\*b<=d+1, b==a=2的值分别为 1 、 0 、 1 、 1 。

4、设 float x=2.5,y=4.7; int a=7;，表达式 x+a%3\*(int)(x+y)%2/4的值为 2.5 。

5、判断变量 a、b 的值均不为 0 的逻辑表达式为a!=0&&b!=0 。

6、求解赋值表达式 a=(b=10)%(c=6)，表达式值、a、b、c 的值依次为 4,4,10,6 。

7、求解逗号表达式 x=a=3,6\*a 后，表达式值、x、a 的值依次为18,3,3。

8、数学式a/(b\*c)的 C 语言表达式 a/b/c 。

9、若有语句

int i=-19, j=i%4;

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

则输出结果是\_\_\_\_\_\_。

答案：-3

10、设y 是int型变量，请写出判断y为奇数的关系表达式\_ y%2\_\_\_\_\_。

1. **CH3 顺序程序设计**

**3.1 选择题**

1.如果int a=3,b=4；则条件表达式"a<b? a:b"的值是\_\_A\_\_。

A) 3 B) 4 C) 0 D) 1

表达式1？表达式2：表达式3

先计算表达式1，

若表达式1成立， 则选择计算表达式2，并表达式2的值作为整个大表达式的值；

若表达式1不成立，则选择计算表达式3，并将表达式3的值作为整个大表达式的值

此题中的a<b相当于表达式1，a相当于表达式2，b相当于表达式3.

a为3，b为4。a<b表达式1成立，因此计算表达式2，并将表达式2的值即a中的值，并作为整个表达式的值，因此整个表达式的值为3

2.若int x=2,y=3,z=4 则表达式x<z?y:z的结果是( B ).

A)4 B)3 C)2 D)0 E)1

3.C语言中，关系表达式和逻辑表达式的值是（ B ） 。

A) 0 B) 0或1 C) 1 D) ‘T’或’F’

4.下面( D )表达式的值为4.

   A) 11/3           B) 11.0/3

   C) (float)11/3    D) (int)(11.0/3+0.5)

（1）相同数据类型的元素进行数学运算（+、-、\*、/）得到结果还保持原数据类型。

（2）不同数据类型的元素进行数学运算，先要统一数据类型，统一的标准是低精度类型转换为高精度的数据类型。

选项A，11与3为两个整数，11/3结果的数据类型也应为整数，因此将3.666666的小数部分全部舍掉，仅保留整数，因此11/3=3.

选项B，11.0为实数，3为整数，因此首先要统一数据类型，将整型数据3转换为3.0，转换后数据类型统一为实型数据，选项B变为11.0/3.0,结果的数据类型也应为实型数据，因此选项B 11.0 /3=3.666666

选项C，先将整数11强制类型转换，转换为实型11.0，因此选项C变为11.0/3，其后计算过程、结果与选项B同

选项D，首先计算11.0/3，其计算过程、结果与选项B同，得到3.666666；再计算3.666666+0.5=4.166666，最后将4.166666强制类型转换为整型，即将其小数部分全部舍掉，结果为4

5.设整型变量 a=2，则执行下列语句后，浮点型变量b的值不为0.5的是（ B　）

A.b=1.0/a B.b=(float)(1/a)

C.b=1/(float)a D.b=1/(a\*1.0)

6.若“int n; float f=13.8;”,则执行“n=(int)f%3”后，n的值是（A）

A.1 B.4 C.4.333333 D.4.6

“(int)f“表示将f中的值强制类型转换为整型，即将13.8的小数部分舍掉，转换为13；然后计算13%3，结果为1，再将结果赋给变量n，因此n的值为1

7.有如下程序

main()

{

int a; char c=10;

float f=100.0; double x;

a=f/=c\*=(x=6.5);

printf("%d %d %3.1f%3.1f\n",a,c,f,x);

}

程序运行后输出结果是\_\_\_B\_\_\_。

A、1 65 1 6.5 B、1 65 1.5 6.5 C、1 65 1.0 6.5 D、2 65 1.5 6.5

8.若变量a、i已正确定义，且i已正确赋值，合法的语句是\_\_\_B\_\_\_。

A、a==1 B、++i; C、a=a++=5; D、a=int(i);

9.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\_\_B\_\_\_\_。

A、a:=b+1 B、a=b=c+2 C、int 18.5%3 D、a=a+7=c+b

10.设变量x为float型且已赋值，则以下语句中能将x中的数值保留到小数点后两位，并将第三位四舍五入的是\_\_C\_\_\_\_。

A、x=x\*100+0.5/100.0 B、x=(x\*100+0.5)/100.0

C、x=(int)(x\*100+0.5)/100.0？？？ D、x=(x/100+0.5)\*100.0

11.若k是int型变量，且有下面的程序片段

k=-3

if(k<=0) printf("####")

else printf("&&&&");

上面程序片段的输出结果是\_\_\_D\_\_\_。

A、#### B、&&&& C、####&&&& D、有语法错误,无输出结果

12.以下4个选项中，不能看作一条语句的是\_\_\_D\_\_\_。

A、{;} B、a=0,b=0,c=0; C、if(a>0); D、if(b==0) m=1;n=2;？？？

13.有以下程序

main()

{

int a=0,b=0;

a=10; /\* 给a赋值 \*/

b=20; /\* 给b赋值 \*/

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

}

程序运行后的输出结果是\_\_\_B\_\_\_。

A、a+b=10 B、a+b=30 C、30 D、出错

14.当c的值不为0时，在下列选项中能正确将c的值赋给变量a、b的是\_\_\_\_C\_\_。

A、c=b=a; B、(a=c)‖(b=c); C、(a=c)&&(b=c); D、a=c=b;

15.以下选项中非法的表达式是\_\_\_D\_\_\_。

A、0<=x<100 B、i=j==0 C、(char)(65+3) D、x+1=x+1

16.以下程序的输出结果是\_\_\_B\_\_\_。

#include <stdio.h>

7main()

{

int a,b,d=241;

a=d/100%9;

b=(-1)&&(-1);

printf("%d,%d\n",a,b);？？？

}

A、6,1 B、2,1 C、6,0 D、2,0

17.以下循环体的执行次数是\_\_\_C\_\_\_。

main()

{

int i,j;

for (i=0,j=1; i<=j+1;i+=2,j--)

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

}

A、3 B、2 C、1 D、0

答案：C

18.有以下定义语句

double a,b; int w; long c;

若各变量已正确赋值，则下列选项中正确的表达式是\_\_\_C\_\_\_。

A、a=a+b=b++ B、w%((int)a+b) C、(c+w)%(int)a D、w=a=b;

19.以下能正确计算定义且赋初值的语句是\_\_\_B\_\_\_。

A、int n1=n2=10; B、char c=32;

C、fioat f=f+1.1; D、double x=12.3E2.5

20.以下合法的赋值语句是\_\_\_B\_\_\_。

A、x=y=100 B、d--; C、x+y; D、c=int(a+b);

21.下列语句中符合C语言语法的赋值语句是\_\_\_D\_\_\_。

A、a=7+b+c=a+7; B、a=7+b++=a+7;

C、a=7+b,b++,a+7 D、a=7+b,c=a+7;

22.请选出合法的C语言赋值语句\_\_\_B\_\_\_。

A、a=b=58 B、i++; C、a=58,b=58 D、k=int(a+b);

23.下面程序的输出是\_\_\_C\_\_\_。

main()

{ int x=10,y=3;

printf("%d\n",y=x/y);

}

A、0 B、1 C、3 D、不确定的值

24.已知字母A的ASCII码为十进制的65，下面程序的输出是\_\_\_A\_\_\_。

main()

{

char ch1,ch2;

ch1='A'+'5'-'3';

ch2='A'+'6'-'3';

printf("%d,%c\n",ch1,ch2);

}

A、67,D B、B,C C、C,D D、不确定的值

25.下列程序的运行结果是\_\_\_B\_\_\_。

#include <stdio.h>

main()

{ int a=2,c=5;

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

}

A、a=%2,b=%5 B、a=2,b=5 C、a=d,b=d D、a=%d,b=5d

26.以下程序的输出结果是\_\_\_D\_\_\_。

#include <stdio.h>

main()

{ int a=2,c=5;

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

}

A、a=%2,b=%5 B、a=2,b=5 C、a=%%d,b=%%d D、a=%d,b=%d

27.有以下程序

main( )

{ int x=102,y=012;

printf("%2d,%2d\n",x,y);

}

执行后输出结果是\_\_\_C\_\_\_。

A、10,01 B、02,12 C、102,10 D、02,10

28.有以下程序

main()

{ int a=666,b=888;

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

}

程序运行后的输出结果是\_\_\_B\_\_\_。

A、错误信息 B、666 C、888 D、666,888

29.有以下程序

#include <stdio.h>

main()

{ char c1='1',c2='2';

c1=getchar();

c2=getchar();

putchar(c1);

putchar(c2);

}

当运行时输入：a <回车> 后，以下叙述正确的是\_\_\_A\_\_\_。

A、变量C1 被赋予字符a，c2被赋予回车符

B、程序将等待用户输入第2个字符

C、变量c1被赋予字符a, c2中仍是原有字符2

D、变量c1 被赋予字符a,c2中将无确定值

30.执行下列程序时输入：123<空格>456<空格>789<回车>，输出结果是\_\_\_D\_\_\_。

main()

{ char s[100];

int c,i;

scanf("%c",&c);

scanf("%d",&i);

scanf("%s",s);

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

}

A、123,456,789 B、1,456,789 C、1,23,456,789 D、1,23,456

31.有定义语句：int b;char c[10];，则正确的输入语句是\_\_B\_\_\_\_。

A、scanf("%d%s",&b,&c); B、scanf("%d%s",&b,c);

C、scanf("%d%s",b,c); D、scanf("%d%s",b,&c);

32.若要求从键盘读入含有空格字符的字符串，应使用函数\_\_\_B\_\_\_。

A、getc() B、gets() C、getchar() D、scanf()

33.有以下程序

#include <stdio.h>

main()

{ char c1,c2,c3,c4,c5,c6; scanf("%c%c%c%c",&c1,&c2,&c3,&c4);

c5=getchar();

c6=getchar();

putchar(c1);

putchar(c2);

printf("%c%c\n",c5,c6);

}

程序运行后，若从键盘输入(从第1列开始) \_\_\_D\_\_\_。

123<回车>

45678<回车>

则输出结果是

A、1267 B、1256 C、1278 D、1245

34.有以下程序

main()

{

int m,n,p;

scanf("m=%dn=%dp=%d",&m,&n,&p);

printf("%d%d%d\n",m,n,p);

}

若想从键盘上输入数据，使变量 m 中的值为 123，n 中的数值为 456，p 中的值为 789，则正确的输入是\_\_\_A\_\_\_。

A、m=123n=456p=789 B、m=123 n456 p=789

C、m=123,n=456,p=789 D、123 456 789

35.有以下程序段

int m=0,n=0;

char c='a';

scanf("%d%c%d",&m,&c,&n);

printf("%d,%c,%d\n",m,c,n);

若从键盘上输入：10A10<回车>，则输出结果是\_\_\_A\_\_\_。

A、10,A,10 B、10,a,10 C、10,a,0 D、10,A,0

36.有定义语句：int x,y;，若要通过scanf("%d,%d",&x,&y);语句使变量x得到数据11，变量y得到数值12，下面四组输入形式中，错误的是\_\_\_A\_\_\_。

A、11 12<回车>B、11,12<回车>C、11, 12<回车>D、11,<回车>12<回车>

37.若有以下定义和语句

char c1='b',c2='e';

printf("%d,%c\n",c2-c1,c2-'a'+'A');

则输出结果是: \_\_B\_\_\_\_。

A、2,M B、3,E C、2,E D、输出项与对应的格式控制不一致,输出结果不确定

38.已定义c为字符型变量，则下列语句中正确的是\_\_C\_\_\_\_。

A、c='97'; B、c="97" ; C、c=97; D、c="a";

39.数字字符 0 的ASCII值为48，若有以下程序\_\_\_C\_\_\_。

main()

{

char a='1',b='2';

printf("%c,",b++);

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

}

程序运行后的输出结果是

A、3,2 B、50,2 C、2,2 D、2,50

40.以下程序的输出结果是\_\_\_A\_\_\_。

main()

{ char c='z';

printf("%c",c-25);

}

A、a B、Z C、z-25 D、y

41.若变量a是int类型，并执行了语句：a='A'+1.6;，则正确的叙述是\_\_\_\_\_\_。

A、a的值是字符C B、a的值是浮点型

C、不允许字符型和浮点型相加 D、a的值是字符'A'的ASCII值加上1

42.下列程序执行后的输出结果是\_\_\_A\_\_\_。

main()

{

int x='f';

printf("%c \n",'A'+(x-'a'+1));

}

A、G B、H C、I D、J

43.以下程序的输出结果是\_\_\_B\_\_\_。

main()

{ int i;

for(i='A';i<'I';i++,i++) printf("%c",i+32);

printf("\n");

}

A、编译不通过，无输出 B、aceg C、acegi D、abcdefghi

44.有如下程序

#include <stdio.h>

main()

{

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

}

程序运行后的输出结果是\_\_A\_\_\_\_。

A、0 B、1 C、－1 D、NULL没定义，出错

45.设有说明语句：char a='\72';则变量a\_\_\_A\_\_\_。

A、包含1个字符 B、包含2个字符 C、包含3个字符 D、说明不合法

46.当把以下四个表达式用作if语句的控制表达式时，有一个选项含义不同，这个选项是\_\_\_\_\_\_。

A、K%2 B、K%2==1 C、(K%2)!=0 D、!K%2==1

47. 以下叙述中错误的是\_\_\_C\_\_\_。

A、C语句必须以分号结束 B、复合语句在语法上被看作一条语句

C、空语句出现在任何位置都不会影响程序运行

D、赋值表达式末尾加分号就构成赋值语句

48.若变量x、y已正确定义并赋值，以下符合C语言语法的表达式是\_\_\_A\_\_\_。

A、++x,y=x-- B、x+1=y C、x=x+10=x+y D、double(x)/10

49.以下叙述中正确的是\_\_\_A\_\_\_。

A、C程序中注释部分可以出现在程序中任意合适的地方

B、花括号“{”和“}”只能作为函数体的定界符

C、构成C程序的基本单位是函数，所有函数名都可以由用户命名

D、分号是C语言之间的分隔符，不是语句的一部分

50.假设变量a,b均为整型，表达式(a=5,b=2,a>b?a++:b++,a+b)的值是（ ）。

A 7 B 8 C 9 D 2

51.设a为int型变量，执行下列赋值语句后，a的取值分别是（ ）。

a=125.534; a=(int)125.521%4; a=5<<2;

A 125,31,1 B 125,1,20 C 125,31,20 D 125.534,2,20

52.以下程序的输出结果为（ ）。

#define ADD(x) x\*x

main( )

{ int a=4,b=6,c=7,d=ADD(a+b)\*c;

printf("d=%d",d);

}

A d=70 B d=80 C d=140 D d=700

**3.2 读程序**

读程序写出结果，或者在程序的空白处填入正确代码。

1. 有如下程序

#include "stdio.h"

main()

{char c;

c=getchar();

while(c!='?') {putchar(c); c=getchar(); }

}

如果从键盘输入abcde？fgh（回车）

运行结果为：abcde

2. 有如下程序

#include <stdio.h>

main()

{ char c;

while((c=getchar())!=’$’)

{ if(‘A’<=c&&c<=‘Z’) putchar(c);

else if(‘a’<=c&&c<=‘z’) putchar(c-32); }

}

当输入为ab\*AB%cd#CD$时，运行结果为：ABABCDCD

3.以下程序的输出结果是\_\_\_\_\_\_。

main()

{ int a=0;

a+=(a=8);

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

}

答案：16

4.若有定义：int a=10,b=9,c=8;接着顺序执行下列语句后，变量b中的值是\_\_\_\_\_\_。

c=(a-=(b-5));

c=(a % 11)+(b=3);

答案：3

5.以下程序段的输出结果是\_\_\_\_\_\_。

int i=9;

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

答案：11

6.有以下程序

#include <stdio.h>

main()

{ char ch1,ch2; int n1,n2;

ch1=getchar(); ch2=getchar();

n1=ch1-'0'; n2=n1\*10+(ch2-'0');

printf("%d\n",n2);

}

程序运行时输入：12<回车>，执行后输出结果是\_\_\_\_\_\_。

答案：12

7.以下程序运行后的输出结果是\_\_\_\_\_\_。

main()

{char m;

m='B'+32;

printf("%c\n",m);

}

答案：b

8.已知字母A的ASCII码为65.以下程序运行后的输出结果是\_\_\_\_\_\_。

main()

{char a,b;

a=’A’+’5’-‘3’; b=a+’6’-‘2’；

printf(“%d%c\n”,a,b);

}

答案：67G

9.若有以下程序

main()

{ char a;

a='H'-'A'+'0';

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

}

执行后输出结果是\_\_\_\_\_\_。

答案：7

10. 以下程序运行后的输出结果是\_\_\_\_\_\_。

main( )

{ int a=012,b=0x12,k=12;

char c=‘\102’,d=‘\r’;

printf(“a=%d\tb=%d\tk=%d\n”,a,b,k);

printf(“c=%c,d=%c%o\n”,c,d,a);

a=‘A’; c=49;

printf(“a=%d\tc=%c\n”,a,c);

}

11. 以下程序运行后的输出结果是\_\_\_\_\_\_。

main( )

{ int i=29,j=6,k=2,s;

s=i+i/j%k-9;

printf(“s=%d\n”,s);

}

12.以下程序运行后的输出结果是\_\_\_\_\_\_。

main（）

{ char \*s2=“I love China!”,\*\*s1=&s2；

char \*s3,c,\*s4=“w”;

s3=&c;

\*s3=‘H’;

s2=s2+2;

printf(“%s\t%c\t%s\t%c\n”,s2,\*s3,s4,\*\*s1);

}

**3.3 程序改错**

画横线的代码有误，请写出正确的代码

1.从键盘输入整数x的值,并输出y的值.

main()

{ float x,y;

scanf(“%d”,&x);

y=3.5+x;

printf(“y=%d”);

}

正确的: int x; float y;

printf(“y=%f”,y);

1. **CH4 选择结构程序设计**

**4.1 读程序**

1.有如下代码：

#include<stdio.h>

main( )

{ char ch;

ch=getchar( );

switch(ch)

{ case ‘A’ : printf(“%c”,’A’);

case ‘B’ : printf(“%c”,’B’); break;

default: printf(“%s\n”,”other”);}

}

当从键盘输入字母A时，运行结果为：AB

**详见switch语句**

**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**

**}**

**其中表达式，常量1，…，常量n都为整型或字符型**

**case相当于给出执行程序的入口和起始位置，若找到匹配的常量，则从此处开始往下执行程序，不再匹配常量，直至遇到break或switch结束**

本题过程：

首先从键盘接收一个字符’A’并将其放在变量ch中。

执行switch语句。Switch后面的条件表达式为ch,因此表达式的值即为字符’A’. 用字符’A’依次与下面的 case中的常量匹配。

与第1个case后的常量匹配，则从其后的语句开始往下执行程序（在执行过程中不再进行匹配。）因此先执行printf(“%c”,’A’)，屏幕上输出A；再往下继续执行printf(“%c”,’B’)，屏幕上输出B；再继续执行break， 此时跳出switch语句。

2. 有如下代码：

#include <stdio.h>

main( )

{ int a=1,b=0；

scanf(“%d”,&a);

switch(a)

{ case 1: b=1；break；

case 2: b=2；break；

default : b=10；}

printf("%d ", b)；

}

若键盘输入5，运行结果为：10

本题过程：

首先用scanf函数为变量a赋值为5。

执行switch语句。switch后面的条件表达式为a,因此表达式的值即为5. 用5依次与下面 case中的常量匹配。没有找到匹配的常量，因此两个case后的语句都不执行。执行default后面的语句b=10；将10赋给变量b。

输出变量b,结果为10

3. 有如下代码：

#include <stdio.h>

main()\_

{ char grade=’C’;

switch(grade)

{

case ‘A’: printf(“90-100\n”);

case ‘B’: printf(“80-90\n”);

case ‘C’: printf(“70-80\n”);

case ‘D’: printf(“60-70\n”); break;

case ‘E’: printf(“<60\n”);

default : printf(“error!\n”);

}

}

运行结果为：

**70-80**

**60-70**

本题过程：

首先从键盘接收一个字符’C’并将其放在变量grade中。

执行switch语句。switch后面的条件表达式为grade,因此表达式的值即为字符’C’. 用字符’C’依次与下面的 case中的常量匹配。

与第3个case后的常量匹配，则从其后的语句开始往下执行程序（在执行过程中不再进行匹配。）因此先执行printf(“70-80\n”);，屏幕上输出70-80,并换行；再往下继续执行printf(“60-70\n”)，屏幕上输出60-70，并换行；再继续执行break， 此时跳出switch语句。

4.在嵌套使用if语句时，C语言规定else总是\_\_\_C\_\_\_。

A、和之前与其具有相同缩进位置的if配对 B、和之前与其最近的if配对

C、和之前与其最近的且不带else的if配对 D、和之前的第一个if配对

5.假定所有变量均已正确说明，下列程序段运行后x的值是\_\_\_B\_\_\_。

a=b=c=0;x=35;

if(!a) x--;

else if(b) ;

if(c) x=3;

else x=4;

A、34 B、4 C、35 D、3

6.有如下程序

main()

{

int a=2,b=-1,c=2;

if(a<b)

if(b<0) c=0;

else c++;

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

}

该程序的输出结果是\_\_\_C\_\_\_。

A、0 B、1 C、2 D、3

7.有以下程序

main()

{ int k=5,n=0;

while(k>0)

{ switch(k)

{ default : break;

case 1: n+=k;

case 2:

case 3: n+=k;

}

k- -;

}

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

}

程序运行后的输出结果是\_\_\_D\_\_\_。

A、0 B、4 C、6 D、7

8.有以下程序

main()

{

int i;

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

switch(i)

{

case 1: printf("%d",i);

case 2: printf("%d",i);

default: printf("%d",i);

}

}

执行后输出结果是\_\_\_A\_\_\_。

A、011122 B、012 C、012020 D、120

9.如下程序

main()

{ int x=1,a=0,b=0;

switch(x)

{ case 0: b++;

case 1: a++;

case 2: a++;b++;

}

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

}

该程序的输出的结果是\_\_\_A\_\_\_。

A、a=2,b=1 B、a=1,b=1 C、a=1,b=0 D、a=2,b=2

10.以下程序的输出结果是\_\_\_A\_\_\_。

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、31 B、13 C、10 D、20

11.下列叙述中正确的是\_\_\_D\_\_\_。

A、break语句只能用于switch语句

B、在switch语句中必须使用default

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

12.设a为整型变量，不能正确表达数学关系:10<a<15的C语言表达式是\_\_\_A\_\_\_。

A、10<a<15 B、a= =11||a==12||a= =13||a==14

C、a>10 && a<15 D、!(a<=10)&&!(a>=15)

13.以下程序的输出结果是\_\_\_A\_\_\_。

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、-1

14.语句:printf("%d",(a=2)&&(b=-2));的输出结果是\_\_\_D\_\_\_。

A、无输出 B、结果不确定 C、-1 D、1

15.设a、b、c、d、m、n均为int型变量，且a=5、b=6、c=7、d=8、m=2、n=2，则逻辑表达式(m=a>b)&&(n=c>d)运算后，n的值为\_\_\_C\_\_\_。

A、0 B、1 C、2 D、3

16.若变量c为char类型，能正确判断出c为小写字母的表达式是\_\_D\_\_\_\_。

A、'a'<=c<='z' B、(c>='a')||(c<='z')

C、('a'<=c)and('z'>=c) D、(c>='a')&&(c<='z')

17.能正确表示逻辑关系：“a≥10 或 a≤0”的C语言表达式是\_\_\_D\_\_\_。

A、a>=10 or a<=0 B、a>=0 | a<=10

C、a>=10 && a<=0 D、a>=10 || a<=10

18.有如下程序

main()

{

int a=1,b=2,m=0,n=0,k;

k=(n=b>a)||(m=a<b);

printf("%d,%d\n",k,m);

}

程序运行后的输出结果是\_\_\_C\_\_\_。

A、0,0 B、0,1 C、1,0 D、1,1

19.设int x=1,y=1;表达式(!x||y--)的值是\_\_\_B\_\_\_。

A、0 B、1 C、2 D、-1

20.设有定义：int a=2,b=3,c=4;，则以下选项中值为0的表达式是\_\_\_A\_\_\_。

A、(!a==1)&&(!b==0) B、(a<b)&&!c||1

C、a&&b D、a||(b+b)&&(c-a)

21.若要求在if后一对圆括号中表示a不等于0的关系，则能正确表示这一关系的表达式为\_\_D\_\_\_。

A、a<>0 B、!a C、a=0 D、a

22.有如下程序段

int a=14,b=15,x;

char c='A';

x=(a&&b)&&(c<'B');

执行该程序段后，x的值为\_\_\_D\_\_\_。

A、true B、false C、0 D、1

23.能正确表示a和b同时为正或同时为负的逻辑表达式是\_\_\_D\_\_\_。

A、(a>=0‖b>=0)&&(a<0‖b<0) B、(a>=0&&b>=0)&&(a<0&&b<0)

C、(a+b>0)&&(a+b<=0) D、a\*b>0

24.以下程序的输出结果是\_\_\_C\_\_\_。

main()

{int a=5,b=4,c=6,d; printf("%d\n",d=a>b?(a>c?a:c):(b));

}

A、5 B、4 C、6 D、不确定

25.若整型变量a、b、c、d中的值依次为：1、4、3、2。则条件表达式 a<b?a:c<d?c:d 的值是\_\_\_\_A\_\_。

A、1 B、2 C、 3 D、4

26.若有定义int x,y; 并已正确给变量赋值，则以下选项中与表达式(x-y)?(x++):(y++)中的条件表达式(x-y)等价的是\_\_\_C\_\_\_。

A、(x-y>0) B、(x-y<0) C、(x-y<0||x-y>0) D、(x-y= =0)

27.有以下程序段

int k=0,a=1,b=2,c=3;

k=a<b?b:a;

k=k>c?k:b;

执行该程序段后，k的值是\_\_\_B\_\_\_。

A、3 B、2 C、1 D、0

28.以下程序段中与语句 k=a>b?(b>c?1:0):0; 功能等价的是\_\_\_A\_\_。

A、if((a>b)&&(b>c)) k=1;else k=0;

B、if((a>b)||(b>c)) k=1;else k=0;

C、if(a<=b) k=0;else if(b<=c) k=1;

D、if(a>b) k=1;else if(b>c) k=1;else k=0;

29.假定w、x、y、z、m均为int型变量，有如下程序段：

w=1;x=2;y=3;z=4;

m=(w<x)?w:x;

m=(m<y)?m:y;

m=(m<z)?m:z;

则该程序执行后，m的值是\_\_\_D\_\_\_。

A、4 B、3 C、2 D、1

30.下面程序的输出结果是\_\_\_A\_\_\_。

#include <stdio.h>

main()

{

int i=2;

printf("%d",f(i,i+1));

}

int f(int a,int b)

{

int c;

c=a;

if(a>b) c=1;

else if(a==b) c=0;

else c=-1;

return(c);

}

A、-1 B、0 C、1 D、2

31.下面的程序运行结果为\_\_\_D\_\_\_。

main()

{

int x=3,y=0,z=0;

if(x=y+z) printf("\* \* \* \*");

else printf("# # # #");

}

A、有语法错误不能通过编译 B、输出\* \* \* \*。

C、可以通过编译，但是不能通过连接，因而不能运行。 D、输出# # # #。

32. 若执行以下程序时从键盘上输入9，则输出结果是\_\_\_\_B\_\_。

main()

{ int n;

scanf("%d",&n);

if(n++<10) printf("%d\n",n);

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

}

A、11 B、10 C、9 D、8

33.下列条件语句中，功能与其他语句不同的是\_\_\_D\_\_\_。

A、if(a) printf("%d\n",x); else printf("%d\n",y);

B、if(a==0) printf("%d\n",y); else printf("%d\n",x);

C、if(a!=0) printf("%d\n",x); else printf("%d\n",y);

D、if(a==0) printf("%d\n",x); else printf("%d\n",y);

34.以下程序运行后的了输出结果是\_\_\_\_\_\_。

main()

{ int a=1,b=2,c=3;

if(c=a) printf("%d\n",c);

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

}

答案：1

35.以下程序运行后的输出结果是\_\_\_\_\_\_。

main()

{int a=1,b=3,c=5;

if (c=a+b) printf("yes\n");

else printf("no\n");

}

答案：yes

36.若从键盘输入58，则以下程序输出的结果是\_\_\_\_\_\_。

main()

{ int a;

scanf("%d",&a);

if(a>50) printf("%d",a);

if(a>40) printf("%d",a);

if(a>30) printf("%d",a);

}

答案：585858

37.以下程序的输出结果是\_\_\_\_\_\_。

main()

{ int a=5,b=4,c=3,d;

d=(a>b>c);

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

}

答案：0

38.若有以下程序

main()

{ int p,a=5;

if(p=1!=0)

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

else

printf("%d\n",p+2);

}

执行后输出结果是\_\_\_\_\_\_。

答案：1

39.表示条件:10<x<100或x<0的C语言表达式是\_\_\_\_\_\_。

答案：(x>10&&x<100||x<0)

40.若已知a=10,b=20,则表达式 !a<b 的值为\_\_\_\_\_\_.

答案：1

41.表示“整数x的绝对值大于5”时值为“真”的C语言表达式是\_\_\_\_\_\_。

答案：x>5||x<-5

42.以下程序运行后的输出结果是\_\_\_\_\_\_。

main()

{int p=30;

printf("%d\n",(p/3>0 ? p/10:p%3));

}

答案：3

43.以下程序运行后的输出结果是\_\_\_\_\_\_。

main()

{

int x,a=1,b=2,c=3,d=4;

x=(a<b)?a:b; x=(x<c)?x:c; x=(d>x)?x:d;

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

}

答案：1

**4.2 程序填空**

1．输入一个字符，判断该字符是数字、字母、空格还是其他字符。

main( )

{ char ch;

ch=getchar();

if( ch>=’a’&&ch<=’z’|| ch>=’A’&&ch<=’Z’ )

printf("It is an English character\n");

else if( ch>=’0’&&ch<=’9’ )

printf("It is a digit character\n");

else if( ch== ‘ ’ )

printf("It is a space character\n");

else

printf("It is other character\n"); }

第1空：字符在计算机中以ASCII码的形式存储。所以当输入的字符，即ch中字符所对应的ASCII码的范围在英文字母的ASCII码的范围内即可，参照p377。由于英文字母又分为大写字母和小写字母，因此此处用一个逻辑或表达式，表示ch中是小写字母或者大写字母，都能使得表达式成立。ch>=97&&ch<=122|| ch>=65&&ch<=90

需要注意的是，对于本题区间所对应的表达式，不可写作97<=ch<=122，也不可写作’A’<=ch <=’Z’. 对于97<=ch<=122因为在计算此表达式时的顺序是从左向右，因此先计算97<=ch。无论ch中的取值如何，表达式97<=ch的值只有两种情况：0或1.所以无论是0还是1，都小于122，因此97<=ch<=122恒成立。

第3空，判断ch中是否为空格，也是通过ch中字符与空格字符的ASCII码来判断。在判断表达式的值是否相等时，用关系符号==；不要用赋值符号=。

2.输入x、y两个整数，按先大后小的顺序输出x、y。

#include “stdio.h”

main( )

{ int x,y, \*px,\*py, (1) ;

scanf(“%d%d”,&x,&y);

(2) ; (3) ;

if(x<y)

{ p=px; px=py; py=p; }

printf(“x=%d,y=%d\n”,x,y);

printf(“MAX=%d,MIN=%d\n”,\*px,\*py);

}

**4.3 程序改错**

1.编程计算下面分段函数, 输入x, 输出y

****

**main()**

**{ int x,y;**

**printf(“\n Input x:\n”);**

**scanf(“%d”, x);** // 错误同上题scanf

**if(x<0)**

**y=x-1;**

**else if(x>=0||x<=10)** // ||表示逻辑或，当左边表达式成立或者右边表达式成立时，整个表达式成立。 &&表示逻辑与，当左边表达式和右边表达式同时成立时，整个表达式成立。此处用逻辑表达式来表示x的区间[0,10], 因此应改用逻辑与符号

**y=2x-1;** // C语言中乘号不能省略，且用\*表示乘法运算

**else**

**y=3x-1;** // C语言中乘号不能省略，且用\*表示乘法运算

**printf(“y=%d”,&y);** //printf与scanf不用，printf后面给出的是变量名列表或表达式列表，无需地址符号

**}**

第一处改为：scanf(“%d”,& x);

第二处改为：x>=0&&x<=10

第三处改为：y=2\*x-1;

第四处改为：y=3\*x-1;

第五处改为：printf(“y=%d”,y);

2.求100~300间能被3整除的数的和。

**main()**

**{ int n;**

**long sum;** / /若定义变量的语句有错误，常见考点有两个：（1）变量的类型，（2）在定义用于存放运算结果的变量时，一定要赋初值, 一般赋值0或者循环初值。

**for(n=100,n<=300,n++)** // for语句的格式，三个表达式之间用分号，且分号不可省略

**{**

**if(n%3=0)**  // = 是赋值符号，用于将右边的值赋给左边的变量；== 是关系符号，用来判断两个值是否相等 。改错中if后面表达式中的赋值符号是常见的考点。

**sum=sum\*n;**

**}**

**printf(“%ld ”,sum);**

**}**

第一处改为：long sum=0;

第二处改为：for(n=100;n<=300;n++)

第三处改为：if(n%3==0)

第四处改为：sum=sum+n;

1. **CH5 循环结构程序设计**

**5.1 读程序**

1.有如下程序

**#include <stdio.h>**

**main()**

**{ int num=0；**

**while(num<=2){ num++；printf("%d\n",num)；}**

**}**

运行结果为：

**1**

**2**

**3**

**详见教材循环结构**

当循环条件num<=2成立的时候，执行循环体{ num++；printf("%d\n",num)；}中的语句。

循环初值num为0;

循环条件num<=2成立

第1次循环：执行num++;即将num中的值加1，执行后num为1；

执行printf("%d\n",num)；在屏幕上输出num中的值，即输出1，之后换行

此时num中的值为1，循环条件num<=2成立

第2此循环：执行num++;即将num中的值加1，执行后num为2；

执行printf("%d\n",num)；在屏幕上输出num中的值，即输出2，之后换行

此时num中的值为2，循环条件num<=2成立

第3此循环：执行num++;即将num中的值加1，执行后num为3；

执行printf("%d\n",num)；在屏幕上输出num中的值，即输出3，之后换行

此时num中的值为3，循环条件num<=2不成立，结束循环。

2.有如下程序

**#include <stdio.h>**

**main( )**

**{ int sum=10,n=1;**

**while(n<3) {sum=sum-n; n++; }**

**printf(“%d,%d”,n,sum);**

**}**

运行结果为：**3,7**

当循环条件n<3成立的时候，执行循环体{sum=sum-n; n++; }中的语句。

循环初值sum为10,n为1;

循环条件n<3成立

第1次循环：执行sum=sum-n=10-1=9；

执行n++,即将n中的值加1，执行后n为2；

此时n中的值为2，sum中的值为9，循环条件n<3成立，继续执行循环

第2次循环：执行sum=sum-n=9-2=7；

执行n++,即将n中的值加1，执行后n为3；

输出此时n,sum中的值，即为3,7。需要注意，在printf(“%d,%d”,n,sum); 中要求输出的数据彼此间用逗号间隔，因此结果的两个数据间一定要有逗号

**3.** 有如下程序

**#include <stdio.h>**

**main()**

**{ int num,c;**

**scanf("%d",&num);**

**do {c=num%10; printf("%d",c); }while((num/=10)>0);**

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

**}**

从键盘输入23，则运行结果为：**32**

**详见教材循环结构；复合的赋值运算符**

**do{ }while(表达式);**

**先无条件执行循环体，再判断循环条件。注意while（表达式）后有分号**

定义整型变量num，c;

为num赋一个整型值；

执行{c=num%10; printf("%d",c); }直到循环条件(num/=10)>0不成立；

输出换行

已知为num赋值23

第1次执行循环体

执行c=num%10=23%10=3;

执行printf("%d",c);输出3

判断循环条件 num/=10等价于num=num/10; 因此num=23/10=2, 2大于0，因此循环条件(num/=10)>0成立，继续执行循环体。执行完第1次循环时，num为2，c为3

第2次执行循环体

执行c=2%10=2;

执行printf("%d",c);再输出2

判断循环条件num=2/10=0,0等于0，因此循环条件(num/=10)>0不成立。结束循环

4.有如下程序

**#include <stdio.h>**

**main()**

**{ int s=0,a=5,n;**

**scanf("%d",&n);**

**do { s+=1; a=a-2; }while(a!=n);**

**printf("%d，%d\n",s,a);**

**}**

若输入的值1，运行结果为： **2,1**

**详见教材循环结构；复合的赋值运算符**

执行{ s+=1; a=a-2; }直到循环条件a!=n不成立；

已知为n赋值1,s为0，a为5

第1次执行循环体

执行s+=1;等价于s=s+1=0+1

执行a=a-2; a=5-2=3

判断循环条件,3不等于1，因此循环条件a!=n成立,继续执行循环体。

执行完第1次循环时，s为1，a为3

第2次执行循环体

执行s+=1;等价于s=s+1=1+1=2

执行a=a-2; a=3-2=1

判断循环条件,1等于1，因此循环条件a!=n不成立,结束循环。

执行完第2次循环时，s为2，a为1

输出此时s,a中的值，结果为2,1

5.有如下程序

**#include <stdio.h>**

**main()**

**{ int x, y =0;**

**for(x=1;x<=10;x++)**

**{ if(y>=10)**

**break;**

**y=y+x;**

**}**

**printf(“%d %d”,y,x);**

**}**

运行结果为：**10 5**

**详见教材for语句**

**详见教材break，continue语句**

**for( 表达式1;表达式2;表达式3)**

**{**

**}**

**(1) 先求解表达式1**

**(2) 求解表达式2，若其值为真，执行循环体，然后执行 (3). 若为假，则结束循环，转到(5)**

**(3) 求解表达式3**

**(4) 转回上面 (2)继续执行**

**(5) 循环结束，执行for语句下面的一个语句**

**break ，跳出循环体；continue,结束本次循环（第i次循环），继续执行下一次循环(第i+1次循环)**

此题 表达式1为x=1，表达式2（循环条件）为x<=10，表达式3为x++

初值x为1，y为0，循环条件（即表达式2）x<=10成立，进入循环体

第1次循环

执行if语句。0小于10，if语句的条件表达式不成立，不执行break;

执行y=y+x; y=0+1=1

转向表达式3，执行x++, x=x+1=1+1=2。循环条件x<=10成立，进入第2次循环

第2次循环

执行if语句。1小于10，if语句的条件表达式不成立，不执行break;

执行y=y+x; y=1+2=3

转向表达式3，执行x++, x=x+1=2+1=3。循环条件x<=10成立，进入第3次循环

第3次循环

执行if语句。3小于10，if语句的条件表达式不成立，不执行break;

执行y=y+x; y=3+3=6

转向表达式3，执行x++, x=x+1=3+1=4。循环条件x<=10成立，进入第4次循环

第4次循环

执行if语句。6小于10，if语句的条件表达式不成立，不执行break;

执行y=y+x; y=6+4=10

转向表达式3，执行x++, x=x+1=4+1=5。循环条件x<=10成立，进入第5次循环

第5次循环

执行if语句。10等于10，if语句的条件表达式成立，执行break，跳出循环。

从break跳出至for语句的下一条语句。执行printf(“%d %d”,y,x);

输出当前的y与x.结果为10 5

6.有如下程序

**#include <stdio.h>**

**main()**

**{ int y=9;**

**for(;y>0;y- -)**

**if(y%3==0)**

**{ printf(%d”,- -y);**

**}**

**}**

运行结果为：

**852**

**详见教材，自增自减符号**

此题 表达式1被省略，表达式2（循环条件）为y>0，表达式3为y--

初值y为9，循环条件（即表达式2）y>0成立，进入循环体

第1次循环

执行if语句。9%3==0，if语句的条件表达式成立，执行printf(%d”,- -y)，即y先自减1变为8，然后在输出，因此屏幕上输出8

转向表达式3，执行y--, y=y-1=8-1=7。循环条件y>0成立，进入第2次循环

第2次循环

执行if语句。7%3不为0，if语句的条件表达式不成立，不执行printf(%d”,- -y)

转向表达式3，执行y--, y=y-1=7-1=6。循环条件y>0成立，进入第3次循环

第3次循环

执行if语句。6%3==0，if语句的条件表达式成立，执行printf(%d”,- -y)，即y先自减1变为5，然后在输出，因此屏幕上输出5

转向表达式3，执行y--, y=y-1=5-1=4。循环条件y>0成立，进入第4次循环

第4次循环

执行if语句。4%3不为0，if语句的条件表达式不成立，不执行printf(%d”,- -y)

转向表达式3，执行y--, y=4-1=3。循环条件y>0成立，进入第5次循环

第5次循环

执行if语句。3%3==0，if语句的条件表达式成立，执行printf(%d”,- -y)，即y先自减1变为2，然后在输出，因此屏幕上输出2

转向表达式3，执行y--, y=y-1=2-1=1。循环条件y>0成立，进入第5次循环

第6次循环

执行if语句。1%3不为0，if语句的条件表达式不成立，不执行printf(%d”,- -y)

转向表达式3，执行y--, y=1-1=0。循环条件y>0不成立，循环结束。

7.有如下程序

**#include <stdio.h>**

**main()**

**{ int i,sum=0; i=1;**

**do{ sum=sum+i; i++; }while(i<=10);**

**printf(“%d”,sum);  
}**

运行结果为： **55**

8.有如下程序

**#include <stdio.h>**

**#define N 4**

**main()**

**{ int i;**

**int x1=1,x2=2;**

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

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

**{ printf("%4d%4d",x1,x2);**

**if(i%2==0)**

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

**x1=x1+x2;**

**x2=x2+x1;**

**}**

**}**

运行结果为：

**1 2 3 5**

**8 13 21 34**

此题 首先为整型变量赋初值x1=1,x2=2

表达式1为i=1，表达式2（循环条件）为i<=N即i<=4，表达式3为i++

循环变量初值i为1，循环条件（即表达式2）i<=4成立，进入第1次循环

第1次循环

执行printf("%4d%4d",x1,x2);因此屏幕上输出1 2

执行if语句。1%2不为0，if语句的条件表达式不成立，不执行printf("\n");

执行x1=x1+x2=1+2=3;此时x1中的值已变为3

执行x2=x2+x1=2+3=5。

转向表达式3，执行i++, i为2。循环条件i<=4成立，进入第2次循环

第2次循环

执行printf("%4d%4d",x1,x2);因此屏幕上输出3 5

执行if语句。2%2==0，if语句的条件表达式成立，执行**printf("\n");换行**

执行x1=x1+x2=3+5=8;此时x1中的值已变为8

执行x2=x2+x1=5+8=13。

转向表达式3，执行i++, i为3。循环条件i<=4成立，进入第3次循环

第3次循环

执行printf("%4d%4d",x1,x2);因此屏幕上输出8 13

执行if语句。3%2不为0，if语句的条件表达式不成立，不执行printf("\n");

执行x1=x1+x2=8+13=21;此时x1中的值已变为21

执行x2=x2+x1=21+13=34。

转向表达式3，执行i++, i为4。循环条件i<=4成立，进入第4次循环

第2次循环

执行printf("%4d%4d",x1,x2);因此屏幕上输出21 34

执行if语句。4%2==0，if语句的条件表达式成立，执行**printf("\n");换行**

执行x1=x1+x2=21+34=55;此时x1中的值已变为55

执行x2=x2+x1=34+55=89。

转向表达式3，执行i++, i为5。循环条件i<=4不成立，结束循环

9.有如下程序

**#include <stdio.h>**

**main( )**

**{ int x, y;**

**for(x=30, y=0; x>=10, y<10; x--, y++)**

**x/=2, y+=2;**

**printf(“x=%d,y=%d\n”,x,y);**

**}**

**运行结果为：**

**x=0,y=12**

10.有如下程序

**#include <stdio.h>**

**#define N 4**

**main( )**

**{ int i,j;**

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

**{ for(j=1;j<i;j++)**

**printf(" ");**

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

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

**}}**

**运行结果为：**

**\***

**\***

**\***

**\***

**详见教材符号常量**

用宏处理指令定义符号常量N为4，在编译过程中，遇到N即视为整数4。

外层for循环，表达式1为i=1，表达式2（循环条件）为i<=N，表达式3为 i++

内层for循环，表达式1为j=1，表达式2（循环条件）为j<i， 表达式3为j++

首先计算外层循环的表达式1，i为1，使得循环条件i<=4成立，进入外层for循环体

外层for循环第1次 此时i为1

内层循环j=1,使得循环条件j<i不成立，因此不执行内层循环体（不输出空格）

执行printf("\*");

执行printf("\n");换行

至此外层循环体执行完，计算外层循环的表达式3，i++，此时i为2. 使得循环条件i<=4

成立，再次进入外层for循环体

外层for循环第2次 此时i为2

内层循环j=1,使得循环条件j<i成立

第1次执行内层循环体printf(" ");

执行内层循环表达式3，j++为2，j<i不成立，跳出内层循环

执行printf("\*");

执行printf("\n");换行

至此外层循环体执行完，计算外层循环的表达式3，i++，此时i为3. 使得循环条件i<=4

成立，进入外层for循环体

外层for循环第3次 此时i为3

内层循环j=1,使得循环条件j<i成立

第1次执行内层循环体printf(" ");

执行内层循环表达式3，j++为2，j<i成立，再次执行内层循环

第2次执行内层循环体printf(" ");

执行内层循环表达式3，j++为3，j<i不成立，跳出内层循环

执行printf("\*");

执行printf("\n");换行

至此外层循环体执行完，计算外层循环的表达式3，i++，此时i为4. 使得循环条件i<=4

成立，进入外层for循环体

外层for循环第4次 此时i为4

内层循环j=1,使得循环条件j<i成立

第1次执行内层循环体printf(" ");

执行内层循环表达式3，j++为2，j<i成立，再次执行内层循环

第2次执行内层循环体printf(" ");

执行内层循环表达式3，j++为3，j<i成立，再次执行内层循环

第3次执行内层循环体printf(" ");

执行内层循环表达式3，j++为4，j<i不成立，跳出内层循环

执行printf("\*");

执行printf("\n");换行

至此外层循环体执行完，计算外层循环的表达式3，i++，此时i为5. 使得循环条件i<=4

不成立，跳出外层for循环体

11.有如下程序

main()

{

int i;

for(i=1;i<40;i++)

{

if(i++%5==0)

if(++i%8==0)

printf("%d ",i);

}

printf("\n");

}

执行后的输出结果是\_\_\_C\_\_\_。

A、5 B、24 C、32 D、40

12.请读程序:

#include<math.h>

#include<stdio.h>

main()

{ float x,y,z;

scanf("%f%f",&x,&y);

z=x/y;

while(z)

{ if(fabs(z)>1.0)

{ x=y; y=z; z=x/y; }

else break;

}

printf("%f\n",y);

}

若运行时从键盘上输入3.6 2.4<CR>(<CR>表示回车)，则输出的结果是\_\_\_B\_\_\_。

A、1.500000 B、1.600000 C、2.000000 D、2.400000

13.有以下程序

main()

{

int k=4,n=0;

for(;n<k;)

{

n++;

if(n%3!=0) continue;

k--;

}

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

}

程序运行后的输出结果是\_\_\_C\_\_\_。

A、1,1 B、2,2 C、3,3 D、4,4

14.有如下程序

main()

{

int x=23;

do

{

printf("%d",x--);

}

while(!x);

}

该程序的执行结果是\_\_B\_\_\_\_。

A、321 B、23 C、不输出任何内容 D、陷入死循环

15.有以下程序

main()

{ int s=0,a=1,n;

scanf("%d",&n);

do

{

s+=1;

a=a-2;

}

while(a!=n);

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

}

若要使程序的输出值为2，则应该从健盘给n输入的值是\_\_\_B\_\_\_。

A、-1 B、-3 C、-5 D、0

16.有以下程序段

int n,t=1,s=0;

scanf("%d",&n);

do

{

s=s+t;

t=t-2;

}

while(t!=n);

为使此程序不陷入死循环，从键盘输入的数据应该是\_\_\_D\_\_\_。

A、任意正奇数 B、任意负偶数 C、任意正偶数 D、任意负奇数

17.有以下程序

main()

{ int i=0,s=0;

do

{

if(i%2) { i++; continue; }

i++;

s+=i;

}

while(i<7);

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

}

执行后输出结果是\_\_A\_\_\_\_。

A、16 B、12 C、28 D、21

18.下面的程序的运行结果是\_\_\_B\_\_\_。

main()

{

int x=3;

do

{

printf("%d\n",x-=2);

}

while(!(--x));

}

A、输出的是1 B、输出的是1和-2 C、输出的是3和0 D、是死循环

19.以下叙述正确的是\_\_\_D\_\_\_。

A、do-while语句构成的循环不能用其它语句构成的循环来代替

B、do-while语句构成的循环只能用break语句退出

C、用do-while语句构成的循环，在while后的表达式为非零时结束循环

D、用do-while语句构成的循环，在while后的表达式为零时结束循环

20.以下程序的输出结果是\_\_\_D\_\_\_。

main()

{ int x=10,y=10,i;

for(i=0;x>8;y=++i)

printf("%d %d ",x--,y);

}

A、10 1 9 2 B、9 8 7 6 C、10 9 9 0 D、10 10 9 1

21.以下程序的输出结果是\_\_\_B\_\_\_。

main()

{ int i, k,a[10],p[3];

k=5;

for(i=0;i<10;i++) a[i]=i;

for(i=0;i<3; i++) p[i]=a[i\*(i+1)];

for(i=0;i<3; i++) k+=p[i]\*2;

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

}

A、20 B、21 C、22 D、23

22.设变量已正确定义，则以下能正确计算f=n!的程序段是\_\_\_D\_\_\_。

A、f=0;for(i=1;i<=n;i++) f\*i;

B、f=1;for(i=1;i<n;i++) f\*=i;

C、f=1;for(i=n;i>1;i++) f\*i;

D、f=1;for(i=n;i>=2;i--) f\*=i;

23.若有如下程序段，其中s、a、b、c均已定义为整型变量，且a、c 均已赋值（c大于0）

s=a;

for(b=1;b<=c;b++) s=s+1;

则上述程序段功能等价的赋值语句是\_\_\_ B \_\_\_。

A、s=a+b; B、s=a+c; C、s=s+c; D、s=b+c;

24.有以下程序

main()

{

int i,s=0;

for(i=1;i<10;i+=2) s+=i+1;

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

}

程序执行后的输出结果是\_\_\_D\_\_\_。

A、自然数1～9的累加和 B、自然数1～10的累加和

C、自然数1～9中奇数之和 D、自然数1～10中偶数之和

25.有如下程序

main()

{ int i,sum;

for(i=1;i<=3;sum++) sum+=i;

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

}

该程序的执行结果是\_\_\_C\_\_\_。

A、6 B、3 C、死循环 D、0

26.若i，j已定义为int类型，则以下程序段中内循环体的总的执行次数是\_\_\_A\_\_\_。

for(i=5;i;i--)

for(j=0;j<4;j++){ …… }

A、20 B、25 C、24 D、30

27.有以下程序

main()

{

int k=5;

while(--k) printf("%d",k-=3);

printf("\n");

}

执行后的输出结果是\_\_\_A\_\_\_。

A、1 B、2 C、4 D、死循环

28.以下程序中，while循环的循环次数是\_\_\_D\_\_\_。

main()

{ int i=0;

while(i<10)

{

if(i<1) continue;

if(i==5) break;

i++;

}

……

}

A、1 B、10 C、6 D、死循环，不能确定次数

29.t为int类型，进入下面的循环之前，t的值为0

while(t=1)

{ …… }

则以下叙述中正确的是\_\_\_B\_\_\_。

A、循环控制表达式的值为0 B、循环控制表达式的值为1

C、循环控制表达式不合法 D、以上说法都不对

30.有如下程序

main()

{ int n=9;

while(n>6)

{ n--;

printf("%d",n);

}

}

该程序段的输出结果是\_\_\_B\_\_\_。

A、987 B、876 C、8765 D、9876

31.有以下程序

main()

{int p[7]={11,13,14,15,16,17,18}; int i=0,k=0;

while(i<7&&p[i]%2)

{

k=k+p[i];

i++;

}

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

}

执行后输出结果是\_\_\_D\_\_\_。

A、58 B、56 C、45 D、24

32.要求通过while循环不断读入字符，当读入字母N时结束循环。若变量已正确定义，以下正确的程序段是\_\_\_A\_\_\_。

A、while((ch=getchar())!='N' printf("%c",ch) ;

B、while(ch=getchar()!='N' printf("%c",ch) ;

C、while(ch=getchar())=='N' printf("%c",ch);

D、while((ch=getchar())=='N' printf("%c",ch);

33.有以下程序段

int k=0;

while(k=1) k++;

while循环执行的次数是\_\_\_A\_\_\_。

A、无限次 B、有语法错，不能执行 C、一闪也不执行 D、执行1次

34.以下程序的输出结果是\_\_\_C\_\_\_。

main()

{

int n[3][3],i,j;

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

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

n[i][j]=i+j;

for(i=0;i<2;i++)

for(j=0;j<2;j++) n[i+1][j+1]+=n[i][j];

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

}

A、14 B、0 C、6 D、值不确定

35.有以下程序

main()

{ int i,j,x=0;

for(i=0;i<2;i++)

{ x++;

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

{ if(j%2)

continue;

x++;

}

x++;

}

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

}

程序执行后的输出结果是\_\_\_B\_\_\_。

A、x=4 B、x=8 C、x=6 D、x=12

36.下列程序的输出结果是\_\_\_C\_\_\_。

main()

{ int i,j,m=0,n=0;

for(i=0;i<2;i++)

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

if(j>=i) m=1;

n++;

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

}

A、4 B、2 C、1 D、0

37.有以下程序

main()

{

int i,n=0;

for(i=2;i<5;i++)

{

do

{

if(i%3)

continue;

n++;

}

while(!i);

n++;

}

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

}

程序执行后的输出结果是\_\_\_D\_\_\_。

A、n=5 B、n=2 C、n=3 D、n=4

38.阅读下列程序:

main()

{ int n[3],i,j,k;

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

n[i]=0;

k=2;

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

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

n[j]=n[i]+1;

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

}

下述程序运行后输出结果是\_\_\_D\_\_\_。

A、2 B、1 C、0 D、3

39.下面程序的功能是输出以下形式的金字塔图案是：

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

main()

{ int i,j;

for(i=1;i<=4;i++)

{ for(j=1;j<=4-i;j++)

printf(" ");

for(j=1;j<=\_\_\_\_;j++) printf("\*");

printf("\n");

}

}

在下划线处应填入的是\_\_\_B\_\_\_。

A、i B、2\*i-1 C、2\*i+1 D、i+2

40.有以下程序

main()

{int num[4][4]={{1,2,3,4},{5,6,7,8}, {9,10,11,12},{13,14,15,16}},i,j;

for(i=0;i<4;i++)

{ for(j=1;j<=i;j++) printf("%4c",' ');

for(j=\_\_\_\_;j<4;j++) printf("%4d",num[i][j]);

printf("\n");

}

}

若要按以下形式输出数组右上半三角

1 2 3 4

6 7 8

11 12

16

则在程序下划线处应填入的是\_\_\_B\_\_\_。

A、i-1 B、I C、i+1 D、4-i

41.有以下程序

main()

{ int a=1,b;

for(b=1;b<=10;b++)

{ if(a>=8) break;

if(a%2==1)

a+=5; continue;}

a-=3;

}

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

}

程序运行后的输出结果是\_\_B\_\_\_\_。

A、3 B、4 C、5 D、6

42.以下程序的功能是：给r输入数据后计算半径为r的圆的面积s程序在编译时出错。

main()

/\* Bejinning \*/

{ int r;

float s;

scanf("%d",&r);

s=\*π\*r\*r;

printf("s=%f\n",s);

}

出错的原因是\_\_\_D\_\_\_。

A、注释语句书写位置错误 B、存放圆半径的变量r不应该定义为整形

C、输出语句中格式描述符非法 D、计算圆面积的赋值语句中使用了非法变量

43.要求以下程序的功能是计算：s=1+1/2+1/3+...+1/10

main()

{ int n;

float s;

s=1.0;

for(n=10;n>1;n--)

s=s+1/n;

printf("%6.4f\n",s);

}

程序运行后输出结果错误，导致错误结果的程序运行是\_\_\_C\_\_\_。

A、s=1.0; B、for(n=10;n>1;n--); C、s=s+1/n; D、printf("%6.4f\n",s);

44.有以下程序

main()

{ int a,b,d=25;

a=d/10%9;

b=a&&(-1);

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

}

程序运行后的输出结果是\_\_B\_\_\_\_。

A、6,1 B、2,1 C、6,0 D、2,0

45.若执行下面的程序时从键盘上输入3和4，则输出是\_\_\_B\_\_\_。

main()

{ int a,b,s;

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

s=a;

if(a<b)

s=b;

s=s\*s;

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

}

A、14 B、16 C、18 D、20

46.以下程序的输出结果是\_\_\_C\_\_\_。

#include <stdio.h>

main()

{ int i;

for(i=1;i<5;i++)

{ if(i%2) printf("\*");

else continue;

printf("#");

}

printf("$\n");

}

A、\*#\*#\*#$ B、#\*#\*#\*$ C、\*#\*#$ D、#\*#\*$

47.以下程序的输出结果是\_\_\_A\_\_\_。

main()

{ int i;

for(i=1;i<6;i++)

{

if(i%2) { printf("#"); continue; }

printf("\*");

}

printf("\n");

}

A、#\*#\*# B、##### C、\*\*\*\*\* D、\*#\*#\*

48.以下程序的输出结果是\_\_\_B\_\_\_。

#include <stdio.h>

main()

{

int i=0,a=0;

while(i<20)

{

for(;;)

{

if(i%10==0) break;

else i--;

}

i+=11;

a+=i;

}

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

}

A、21 B、32 C、33 D、11

49.设有如下程序段，下面描述中正确的是 （B ）。

int k=10; while(k=0) k=k-1;

A 循环执行一次 B循环是无限循环 C循环体语句一次也不执行 D循环体语句执行一次

50.以下程序的输出结果为（B ）。

int i;

void prt( )

{ for(i=5;i<8;i++) printf("%c",'\*');

printf("\t");

}

main( )

{ for(i=5;i<=8;i++) prt( );

}

A \*\*\* B \*\*\* \*\*\* \*\*\* \*\*\* C \*\*\* \*\*\* D \* \* \*

**5.2 程序填空**

1.下列程序的功能是从输入的整数中，统计大于零的整数个数和小于零的整数个数。用输入0来结束输入，用i,j来放统计数，请填空完成程序。

　　 void main()

{ int n,i=0,j=0;

printf(“input a integer,0 for end\n”);

scanf(“%d”,&n);

while ( n或n!=0 ) {

if(n>0) i= i+1 ;

else j=j+1;

}

printf(“i=%4d,j=%4d\n”,i,j);

}

此题用i来记录大于零的整数，用j记录小于零的整数。所以循环条件是n（或者n!=0）即当n不为0时执行循环体。在循环体中是一个选择语句。如果n>0，则令i加1，相当于令正整数的个数加1；否则（即n<0）,令j加1，相当于令负整数的个数加1。

2.编程计算1＋3＋5+……＋101的值

#include <stdio.h>

void main()

{ int i, sum = 0;

for (i = 1; i<=101 ; i=i+2; )

sum = sum + i;

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

for语句的一般形式。

表达式1为i = 1，为循环变量赋初值，即循环从1开始，本题从1到101，因此终值是101，表达式2是循环条件，用来控制循环的结束，因此循环条件为i<=101；表达式3为循环变量的自增

3.编程计算1＋3＋5…＋99的值

main()

{ int i, sum = 0;

i=1;

while ( i<100 )

{ sum = sum + i;

I=i+2 ; }

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

}

4.从键盘输入一个字符，判断它是否是英文字母。

#include <stdio.h>

void main()

{char c;

printf("input a character:");

c=getchar();

if(c>= ‘A’ &&c<= ‘Z’ || c>=‘a’ &&c<= ‘z’) printf("Yes\n");

else printf("No");

}

5.程序读入20个整数，统计非负数个数，并计算非负数之和。

#include <stdio.h>

main( )

{ int i, a[20], s, count;

s=count=0;

for(i=0; i<20; i++)

scanf(“%d”, &a[i] );

for(i=0; i<20; i++)

{ if( a[i]<0 ) continue ;

s+=a[i] ;

count++;

}

printf(“s=%d\t count=%d\n”, s, count”);

}

6.输入一个正整数n (1<n≤10)，再输入n个整数，用选择法将它们从小到大排序后输出。

#include <stdio.h>

int main(void){

int i, index, k, n, temp;

/\* 定义1个数组a，它有10个整型元素\*/

printf(“Enter n: ");

printf(“Enter %d integers: ", n);

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

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

for(k = 0; k < n-1; k++){ /\* 对n个数排序 \*/

index = k;

for( )

if( ) index = i;

}

printf("After sorted: ");

for(i = 0; i < n; i++) /\* 输出n个数组元素的值 \*/

return 0;}

7.下面程序的输出是\_\_\_\_\_\_。

main()

{ int x=2;

while(x--);

printf("%d\n", x);}

答案：-1

8.以下程序的输出结果是\_\_\_\_\_\_。

void fun()

{ static int a=0;

a+=2; printf("%d",a);

}

main()

{ int cc;

for(cc=1;cc<4;cc++) fun();

printf("\n")

}

答案：246

9.有以下程序

main()

{ int t=1,i=5;

for(;i>=0;i--) t\*=i;

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

}

执行后输出结果是\_\_\_\_\_\_。

答案：0

10.以下程序的输出结果是\_\_\_\_\_\_。

main()

{ int s,i;

for(s=0,i=1;i<3; i++,s+=i);

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

}

答案：5

11.以下程序运行后的输出结果是\_\_\_\_\_\_。

main()

{int p[7]={11,13,14,15,16,17,18};

int i=0,j=0;

while(i<7&&p[i]%2==1)j+=p[i++];

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

}

答案：24

12.执行以下程序后，输出'#'号的个数是\_\_\_\_\_\_。

#include <stdio.h>

main()

{int i,j;

for (i=1;i<5;i++)

for (j=2;j<=i;j++) putchar('#');

}

答案：6

13.以下程序的功能是计算：1+12+123+1234+12345。请填空。

Main()

{int t= 0,s=0,I;

for (I=1;I<=5;I++)

{t=I+\_\_\_\_\_\_;s=s+t;}

printf(“s=%d\n”,s);

}

答案：t\*10~10\*t

14.用选择排序法对任意10个整数按照由小到大排序。

main()

{ int a[11],i,j,k,t;

printf("Please input 10 numbers:\n");

for(i=1;i<11;i++) scanf("%d",&a[i]);

printf("\n");

for(i=1;i<=9;i++)

{ (7) ;

for(j=i+1; (8) ;j++)

if(a[j]>a[k]) k=j;

if(k!=i)

{ t=a[k]; a[k]=a[i]; a[i]=t;}

}

printf("The sorted numbers:\n");

for(i=1;i<11;i++) printf("%d ",a[i]);

}

15.百马百担问题：有100匹马，驮100担货，大马驮3担，中马驮2担，两匹小马驮1担，问大、中、小马各多少匹?

main( )

{ int hb,hm,hl,n=0;

for(hb=0;hb<=100;hb+= (1) )

for(hm=0;hm<=100-hb;hm+= (2) )

{ hl=100-hb- (3) ;

if(hb/3+hm/2+2\* (3) ==100)

{ n++;

printf("hb=%d,hm=%d,hl=%d\n",hb/3,hm/2,2\*hl);

}

}

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

}

**5.4 程序改错**

下面每个程序的划线处有语法或逻辑错误，请找出并改正，使其得到符合题意的执行结果。

1.求1234n

**main()**

**{ long int sum;** / /若定义变量的语句有错误，常见考点有两个：（1）变量的类型，（2）在定义用于存放运算结果的变量时，一定要赋初值。一般赋值0或者循环初值。

**int n,i=1;**

**scanf("%d",n);** //若scanf语句有错误，常见考点有两个：（1）格式声明符号要与后面欲赋值的变量的类型一致，此题%d与n的类型int一致（详见p69-78）；（2）变量的前面要有地址符号&

**printf(“\n”);**

**while(i<n)** // 循环条件用于控制循环的次数，若以i<n 为循环条件，则意味着i的终值为n-1, 由于且i初值为1，因此一共能够循环n-1次。比要求少了1次，因此应改为i<=n 或者i<n+1

**{ sum=sum\*i;** // 若不为sum赋初值，则此处无法计算sum\*i。

**i++;**

**}**

**printf("sum=%d",sum);**  //若printf语句有错误，常见考点有1个：格式声明符号要与后面欲输出的变量的类型一致，此题%d与sum的类型long int不一致，应改为%ld（详见p69-78）；

**}**

sum应初始化 即加入sum=1

第四行改为：scanf(“%d”,&n);

第六行改为：while(i<=n)或者 while(i<n+1)

第十行改为：printf("sum=%ld",sum);

2.程序读入20个整数，统计非负数个数，并计算非负数之和。

**#include “stdio.h”**

**main()**

**{**

**int i, s , count ,n=20;**

**int a[n];** //数组定义的一般形式，详见p143, 其中的常量表达式不能为变量

**s=count=1;**

**for( i=1, i<20, i- -)** // for语句的格式，三个表达式之间用分号，且分号不可省略

**scanf(“%d”, a[i] );** //若scanf语句有错误，常见考点有两个：（1）格式声明符号要与后面欲赋值的变量的类型一致，此题%d与n的类型int一致（详见p69-78）；（2）变量的前面要有地址符号&

**for(i=0;i<20;i++)**

**{**

**if(a[i]<0)**

**break;**  // break与continue的区别 p128. 在改错题中若错误出现在break语句，则通常是将break换为continue；反之，若错误出现在continue，通常是将其换为break

**s +=a[i];**

**count++;**

**}**

**printf(“s=%f count=%f\n”, s, count);** //若printf语句有错误，常见考点有1个：格式声明符号要与后面欲输出的变量的类型一致

**}**

答案：int a[20]

s=count=0;

for(i=0;i<20;i--)

scanf(“%d”,&a[i]);

continue;

printf(“s=%d count=%d\n”,s,count);

1. **CH6 数组**

**6.1 选择题**

1.以下对一维数组a的正确说明是： D

A）char a（10）； B）int a[]；

C）int k＝5，a[k]； D）char a[3]={‘a’,’b’,’c’};

详见教材，一维数组的定义、初始化

类型符 数组名 [常量表达式]

类型符是指数组中数组元素的类型；数组名要符合标识符命名规则；常量表达式是指数组的长度（数组中包含元素的个数），其值只能是整数，不可以是变量，而且从1开始计数。

选项A，常量表达式只能放在中括号 [ ]中

选项B，只有在对数组初始化（即赋值）的时候才可以省略数组的长度，B中并未对a进行初始化。

选项C，常量表达式不能为变量。

2.以下能对一维数组a进行初始化的语句是: ( C )

A. int a[5]=(0,1,2,3,4,) B. int a(5)={}

C. int a[3]={0,1,2} D. int a{5}={10\*1}

详见教材，一维数组的定义、初始化

选项B,D，常量表达式只能放在中括号 [ ]中

选项A，数组可以看做是若干个相同数据类型元素的有序集合，因此以集合的形式对其初始化，使用{ }对其初始化，选项A用了（）.

3.在C语言中对一维整型数组的正确定义为 D 。

A)int a(10); B)int n=10,a[n];

C)int n;a[n]; D)#define N 10

int a[N];

4.已知：int a[10]; 则对a数组元素的正确引用是（ D ）。

A、a[10] B、a[3.5] C、a(5) D、a[0]

详见教材，数组元素的引用

数组名[下标]

引用数组元素时，[ ]中的下标为逻辑地址下标，只能为整数，可以为变量，且从0开始计数

int a[10]表示定义了一个包含10个整型数据的数组a，数组元素的逻辑地址下标范围为0~9，即a[0] 表示组中第1个元素; a[1] 表示组中第2个元素; a[2] 表示组中第3个元素; ......;a[9] 表示组中第10个元素.

选项A，超过了数组a的逻辑地址下标范围；

选项B，逻辑地址下标只能为整数

选项C，逻辑地址下标只能放在[ ]中

5.若有以下数组说明，则i=10;a[a[i]]元素数值是（C　）。

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

A.10 B.9 C.6 D.5

先算a[a[i]]内层的a[i]，由于i=10,因此a[i]即a[10].

a[10]对应下面数组中的元素为9. 因此a[a[i]]即为a[9]

a[9]对应下面数组中的元素为6. 因此a[9]即为6

6.若有说明：int a[][3]={{1,2,3},{4,5},{6,7}}; 则数组a的第一维的大小为: ( B )

A. 2 B. 3 C. 4 D.无确定值

5 7 D) 3 6 9

二维数组的一维大小，即指二维数组的行数，在本题中，按行对二维数组赋值，因此内层有几个大括号，数组就有几行

7.对二维数组的正确定义是（ C　）

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

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

详见教材，二维数组的定义、初始化

类型符 数组名 [常量表达式][常量表达式]

二维数组可以看做是矩阵

类型符是指数组中数组元素的类型；数组名要符合标识符命名规则；第一个常量表达式是指数组的行数；第二个常量表达式是指数组的列数；常量表达式的值只能是整数，不可以是变量，而且从1开始计数。

一维数组初始化时可以省略数组长度

二维数组初始化时可以省略行数，但不能省略列数

选项A,B，都省略了列数

选项D，不符合二维数组定义的一般形式，行、列常量表达式应该放在不同的[]中

8.已知int a[3][4];则对数组元素引用正确的是\_\_C\_\_\_

A)a[2][4] B)a[1,3] C)a[2][0] D)a(2)(1)

详见教材，数组元素的引用

数组名[下标] [下标]

引用数组元素时，[ ]中的下标为逻辑地址下标，只能为整数，可以为变量，且从0开始计数

第一个[下标]表示行逻辑地址下标，第二个[下标]表示列逻辑地址下标。

因此a的行逻辑地址范围0~2；a的列逻辑地址范围0~3；

选项A，列逻辑地址下标超过范围

选项B,D，的引用形式不正确。

9. 以下不能正确定义二维数组的选项是\_\_D\_\_\_\_。

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.以下能正确定义二维数组的是\_\_\_B\_\_\_。

A、int a[][3]; B、int a[][3]={2\*3}; C、int a[][3]={}; D、int a[2][3]={{1},{2},{3,4}};

11.以下程序的输出结果是\_\_\_A\_\_\_。

main()

{

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

for (i=1;i<3;i++)

for(j=0;j<=i;j++) s+=a[i][j];

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

}

A、18 B、19 C、20 D、21

12.以下程序的输出结果是\_\_C\_\_\_\_。

main()

{ int i,x[3][3]={1,2,3,4,5,6,7,8,9};

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

printf("%d,",x[i][2-i]);

}

A、1,5,9 B、1,4,7 C、3,5,7 D、3.6.9

13.以下数组定义中错误的是\_\_\_B\_\_\_。

A、int x[][3]={0}； B、int x[2][3]={{1,2},{3,4},{5,6}};

C、int x[][3]={{1,2,3},{4,5,6}}; D、int x[2][3]={1,2,3,4,5,6};

14.以下数组定义中不正确的是\_\_\_D\_\_\_。

A、int a[2][3]; B、int b[][3]={0,1,2,3};

C、int c[100][100]={0}; D、int d[3][]={{1,2},{1,2,3},{1,2,3,4}};

15.有以下程序

main()

{

int m[][3]={1,4,7,2,5,8,3,6,9};

int i,j,k=2;

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

{ printf("%d ",m[k][i]); }

}

执行后输出结果是\_\_\_C\_\_\_。

A、4 5 6 B、2 58 C、3 6 9 D、7 8 9

16.以下能正确定义数组并正确赋初值的语句是\_\_\_D\_\_\_。

A、int N=5,b[N][N]; B、int a[1][2]={{1},{3}};

C、int c[2][]={{1,2},{3,4}}; D、int d[3][2]={{1,2},{34}};

17.以下程序的输出结果是\_\_\_A\_\_\_。

main()

{

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

printf("%d%d%d%d\n",a[0][3],a[1][2],a[2][1],a[3][0]);

}

A、0650 B、1470 C、5430 D、输出值不定

18.定义如下变量和数组:

int i;

int x[3][3]={1,2,3,4,5,6,7,8,9};

则下面语句的输出结果是\_\_C\_\_\_\_。

for(i=0;i<3;i++) printf("%d ",x[i][2-i]);

A、1 5 9 B、1 47 C、3 5 7 D、3 6 9

19.有以下程序

main()

{

int x[3][2]={0},i;

for(i=0;i<3;i++) scanf("%d",x[i]);

printf("%3d%3d%3d\n",x[0][0],x[0][1],x[1][0]);

}

若运行时输入：2 4 6<回车>，则输出结果为\_\_\_B\_\_\_。

A、 2 0 0 B、 2 0 4 C、 2 4 0 D、 2 4 6

20.当调用函数时，实参是一个数组名，则向函数传送的是\_\_\_B\_\_\_。

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、数组每个元素中的值

21.有以下程序

void sum(int a[])

{ a[0]=a[-1]+a[1]; }

main()

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

sum (&a[2]);

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

}

程序运行后输出结果是\_\_A\_\_\_\_。

A、6 B、7 C、5 D、8

22.有以下程序

main()

{ char s[]="abcde";

s+=2;

printf("%d\n",s[0]);

}

执行后的结果是\_\_\_D\_\_\_。

A、输出字符a的ASCII码 B、输出字符c的ASCII码 C、输出字符c D、程序出错

23.请读程序：

#include <stdio.h>

f(int b[],int n)

{ int i,r;

r=1;

for(i=0;i<=n;i++) r=r\*b[i];

return r;

}

main()

{ int x,a[]={2,3,4,5,6,7,8,9};

x=f(a,3);

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

}

上面程序的输出结果是\_\_\_B\_\_\_。

A、720 B、120 C、24 D、6

24.若要求定义具有10个int形元素的一维数组a，则以下的定义语句中错误的是\_\_\_\_D\_\_。

A、#define N 10 int a[N]; B、#define n 5 int a[2\*n];

C、int a[5+5]; D、int n=10,a[n];

25.合法的数组定义是\_\_D\_\_\_\_。

A、int a[]="string"; B、int a[5]={0,1,2,3,4,5};

C、char s="string"; D、char a[]={0,1,2,3,4,5};

26.以下能正确定义一维数组的选项是\_\_\_B\_\_\_。

A、int a[5]={0,1,2,3,4,5}; B、char a[]={0,1,2,3,4,5};

C、char a={'A','B','C'}; D、int a[5]="0123";

27.有如下程序

main()

{ int n[5]={0,0,0},i,k=2;

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

n[i]=n[i]+1;

printf("%d\n",n[k]);

}

该程序的输出结果是\_\_\_D\_\_\_。

A、不确定的值 B、2 C、1 D、0

28.以下程序的输出结果是\_\_\_C\_\_\_。

main()

{ int i,a[10];

for(i=9;i>=0;i--) a[i]=10-i;

printf("%d%d%d",a[2],a[5],a[8]);

}

A、258 B、741 C、852 D、369

29.执行下面的程序段后，变量k中的值为\_\_\_A\_\_\_。

int k=3,s[2];

s[0]=k; k=s[1]\*10;

A、不定值 B、32 C、30 D、10

30.若有定义和语句:

char s[10]:s="abcd";printf("%s\n",s);

则结果是(以下u代表空格) \_\_\_D\_\_\_。

A、输出abcd B、输出a C、输出abcduuuuu D、编译不通过

31.函数调用：strcat(strcpy(str1,str2),str3)的功能是\_\_\_C\_\_\_。

A、将串str1复制到串str2中后再连接到串str3之后

B、将串str1连接到串str2之后再复制到串str3之后

C、将串str2复制到串str1中后再将串str3连接到串str1之后

D、将串str2连接到串str1之后再将串str1复制到串str3中

32.以下正确定义字符串的语句是\_\_D\_\_\_\_。

A、char str[]={'\064'}； B、char str="\x43";

C、char str=''； D、char str[]="\0";

33.有以下程序

main()

{

char s[]="\n123\\";

printf("%d,%d\n",strlen(s),sizeof(s));

}

执行后输出结果是\_\_\_C\_\_\_。

A、赋初值的字符串有错 B、6,7 C、5,6 D、6,6

34.有以下程序

#include <string.h>

main()

{charp[]={'a','b','c'}; char q[10]={'a','b','c'};

printf("%d%d\n",strlen(p),strlen(q));

}

以下叙述中正确的是\_\_\_B\_\_\_。

A、在给p和q数组置初值时，系统会自动添加字符串结束符，故输出的长度都为3

B、由于p数组中没有字符串结束符，长度不能确定，但q数组中字符串长度为3

C、由于q数组中没有字符串结束符，长度不能确定，但p数组中字符串长度为3

D、由于p和q的数组中都没有字符串结束符，故长度不能确定

35.有以下程序

main()

{char p[]={'a','b','c'},q[]="abc";

printf("%d%d\n",sizeof(p),sizeof(q));

};

程序运行后和输出结果是\_\_\_C\_\_\_。

A、4 4 B、3 3 C、3 4 D、4 3

36.已知职工记录描述如下，在Turbo C中，系统为变量w分配（ ）字节的空间。

struct worker

{ int no;

char name[20];

char sex;

union

{ int day; int month; int year;}birth;

} w;

A 29 B 20 C 25 D 6

37.下面选项中正确的赋值语句是(设 char a[5],\*p＝a;)（ ）。

A p="abcd" B a="abcd"

C \*p="abcd" D \*a="abcd";

38.设有以下程序段，则值为6的表达式是（ ）。

struct st { int n; struct st \*next;};

static struct st a[3]={5,&a[1],7,&a[2],9,0 }，\*p;

p=&a[0];

A p++->n B ++p->n

C p->n++ D (\*p).n++

39. int a[4]={5,3,8,9};其中 a[3]的值为( )。D

A. 5 B. 3 C. 8 D. 9

40.以下 4 个字符串函数中，( )所在的头文件与其他 3 个不同。A

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

41.以下 4 个数组定义中，( )是错误的。 D

A. int a[7] B. #define N 5 long b[N]

C. char c[5] D. int n,d[n];

42.对字符数组进行初始化，( )形式是错误。 B

A. char c1[ ]={'1', '2', '3'} B. char c2[ ]=123

C. char c3[ ]={ '1', '2', '3', '\0'} D. char c4[ ]="123";

43.在数组中，数组名表示( )。 A

A. 数组第 1 个元素的首地址 B.数组第 2 个元素的首地址

C. 数组所有元素的首地址 D.数组最后 1 个元素的首地址

44.若有以下数组说明，则数值最小的和最大的元素下标分别是( )。 B

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

A. 1,12 B. 0,11 C. 1,11 D. 0,12

45.若有以下说明，则数值为 4 的表达式是( )。 D

int a[12] ={1,2,3,4,5,6,7,8,9,10,11,12}; char c=’a’, d, g ;

A. a[g-c] B. a[4] C. a['d'-'c'] D. a['d'-c]

46.设有定义：char s[12] = "string" ; 则printf( "%d\n",strlen(s)); 的输出是( )。 A

A. 6 B. 7 C. 11 D. 12

47.设有定义：char s[12] = "string"; 则printf("%d\n ", sizeof(s)); 的输出是( )。 D

A. 6 B. 7 C. 11 D. 12

48.合法的数组定义是( )。 A

A. char a[ ]= "string " B. int a[5] ={0,1,2,3,4,5}

C. char a= "string " D. char a[ ]={0,1,2,3,4,5}

49.合法的数组定义是( )。 D

A. int a[3][ ]={0,1,2,3,4,5} B. int a[ ][3] ={0,1,2,3,4};

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

50.下列语句中，正确的是 ( )。D

A. char a[3][ ]={'abc', '1'} B. char a[ ][3] ={'abc', '1'};

C. char a[3][ ]={'a', "1"} D. char a[ ][3] ={ "a", "1"};

51.下列定义的字符数组中，输出 printf("%s\n", str[2]) ;的输出是( )。 C

static str[3][20] ={ "basic", "foxpro", "windows"};

A. basic B. foxpro C. windows D. 输出语句出错

52.下列各语句定义了数组，其中哪一个是不正确的( )。 C

A. char a[3][10]={"China","American","Asia"}

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

C. float x[2][ ]={1,2,4,6,8,10}

D. int m[][3]={1,2,3,4,5,6};

53. 数组定义为 int a[3][2]={1,2,3,4,5,6}，值为6的数组元素是( )。 B

A. a[3][2] B. a[2][1] C. a[1][2] D. a[2][3]

54.下面的程序中哪一行有错误( )。D

#include <stdio.h>

main()

{

float array[5]={0.0}; //第A行

int i;

for(i=0;i<5;i++)

scanf("%f",&array[i]);

for(i=1;i<5;i++)

array[0]=array[0]+array[i];//第B行

printf("%f\n",array[0]); //第C行

}

A. 第A行 B. 第B行 C. 第C行 D. 没有

55.下面哪一项是不正确的字符串赋值或赋初值的方式( )。 C

A. char \*str; str="string";

B. char str[7]={'s','t','r','i','n','g'};

C. char str1[10];str1="string";

D. char str1[]="string",str2[]="12345678";

56.若有以下说明和语句，则输出结果是哪一项( )。C

(strlen(s)为求字符串s的长度的函数)

char s[12]="a book!";

printf("%d",strlen(s));

A. 12 B. 8 C. 7 D. 11

57.若有以下说明和语句，则输出结果是( )。B

(strlen(s)为求字符串s的长度的函数)

char sp[]="\t\v\\\0will\n";

printf("%d",strlen(sp));

A. 14 B. 3 C. 9 D. 字符串中有非法字符

58.若有以下说明和语句，则输出结果是( )。 C

char str[]="\"c:\\abc.dat\"";

printf("%s",str);

A. 字符串中有非法字符

B. \"c:\\abc.dat\"

C. "c:\abc.dat"

D. "c:\\abc.dat"

**6.2 读程序**

1.有如下程序：

**#include <stdio.h>**

**main()**

**{ int i, a[10];**

**for(i=9;i>=0;i--)**

**a[i]=10-i;**

**printf(“%d%d%d”,a[2],a[5],a[8]);**

**}**

**运行结果为：**

**852**

**例题6.1一定看懂！**

首先定义整型变量i，整型数组a, a的长度为10，即a中包含10个整型元素（整型变量）

执行for循环语句

初值i=9, 使得循环条件i>=0成立，执行循环体

第1次循环

执行a[i]=10-i 等价于a[9]=10-9=1

计算表达式3，即i--，i为8，使得循环条件i>=0成立，继续执行循环体

第2次循环

执行a[i]=10-i 等价于a[8]=10-8=2

计算表达式3，即i--，i为7，使得循环条件i>=0成立，继续执行循环体

第3次循环

执行a[i]=10-i 等价于a[7]=10-7=3

计算表达式3，即i--，i为6，使得循环条件i>=0成立，继续执行循环体

第4次循环

执行a[i]=10-i 等价于a[6]=10-6=4

计算表达式3，即i--，i为5，使得循环条件i>=0成立，继续执行循环体

第5次循环

执行a[i]=10-i 等价于a[5]=10-5=5

计算表达式3，即i--，i为4，使得循环条件i>=0成立，继续执行循环体

第6次循环

执行a[i]=10-i 等价于a[4]=10-4=6

计算表达式3，即i--，i为3，使得循环条件i>=0成立，继续执行循环体

第7次循环

执行a[i]=10-i 等价于a[3]=10-3=7

计算表达式3，即i--，i为2，使得循环条件i>=0成立，继续执行循环体

第8次循环

执行a[i]=10-i 等价于a[2]=10-2=8

计算表达式3，即i--，i为1，使得循环条件i>=0成立，继续执行循环体

第9次循环

执行a[i]=10-i 等价于a[1]=10-1=9

计算表达式3，即i--，i为0，使得循环条件i>=0成立，继续执行循环体

第10次循环

执行a[i]=10-i 等价于a[0]=10-0=10

计算表达式3，即i--，i为-1，使得循环条件i>=0不成立，跳出循环体

2.有如下程序：

**#include <stdio.h>**

**main()**

**{ int i,a[6];**

**for (i=0; i<6; i++)**

**a[i]=i;**

**for (i=5; i>=0 ; i--)**

**printf("%3d",a[i]);**

**}**

**运行结果为：**

**5 4 3 2 1 0**

首先定义整型变量i，整型数组a, a的长度为6，即a中包含6个整型元素（整型变量）

执行第一个for循环语句

初值i=0, 使得循环条件i<6成立，执行循环体

第1次循环

执行a[i]= i 等价于a[0]=0

计算表达式3，即i++，i为1，使得循环条件i<6成立，继续执行循环体

第2次循环

执行a[i]= i 等价于a[1]=1

计算表达式3，即i++，i为2，使得循环条件i<6成立，继续执行循环体

第3次循环

执行a[i]= i 等价于a[2]=2

计算表达式3，即i++，i为3，使得循环条件i<6成立，继续执行循环体

第4次循环

执行a[i]= i 等价于a[3]=3

计算表达式3，即i++，i为4，使得循环条件i<6成立，继续执行循环体

第5次循环

执行a[i]= i 等价于a[4]=4

计算表达式3，即i++，i为5，使得循环条件i<6成立，继续执行循环体

第6次循环

执行a[i]= i 等价于a[5]=5

计算表达式3，即i++，i为6，使得循环条件i<6不成立，结束循环

执行第二个for循环语句

初值i=5, 使得循环条件i>=0成立，执行循环体

第1次循环

执行printf("%3d",a[i]); 即输出a[5]的值

计算表达式3，即i--，i为4，使得循环条件i>=0成立，继续执行循环体

第2次循环

执行printf("%3d",a[i]); 即输出a[4]的值

计算表达式3，即i--，i为3，使得循环条件i>=0成立，继续执行循环体

第3次循环

执行printf("%3d",a[i]); 即输出a[3]的值

计算表达式3，即i--，i为2，使得循环条件i>=0成立，继续执行循环体

第4次循环

执行printf("%3d",a[i]); 即输出a[2]的值

计算表达式3，即i--，i为1，使得循环条件i>=0成立，继续执行循环体

第5次循环

执行printf("%3d",a[i]); 即输出a[1]的值

计算表达式3，即i--，i为0，使得循环条件i>=0成立，继续执行循环体

第6次循环

执行printf("%3d",a[i]); 即输出a[0]的值

计算表达式3，即i--，i为6，使得循环条件i>=0不成立，结束循环

3.有如下程序：

**#include <stdio.h>**

**main( )**

**{ int i,k,a[10],p[3]；**

**k=5；**

**for(i=0；i<10；i++)**

**a[i]=i；**

**for(i=0；i<3；i++)**

**p[i]=a[i\*(i+1)]；**

**for(i=0；i<3；i++)**

**k+=p[i]\*2；**

**printf("%d\n",k)；**

**}**

**运行结果为：21**

首先定义整型变量i，k，整型数组a, a的长度为10，整型数组p, p的长度为3

k初值为5

第一个for循环语句为数组a进行初始化

执行完第一个for语句后，a[0]=0，a[1]=1，a[2]=2，a[3]=3，a[4]=4，a[5]=5，a[6]=6，a[7]=7，a[8]=8，a[9]=9 （循环过程略）

第二个for循环语句为数组p进行初始化

初值i=0, 使得循环条件i<3成立，执行循环体

第1次循环

执行p[i]=a[i\*(i+1)]; 即p[0]=a[0\*(0+1)]=a[0]=0

计算表达式3，即i++，i为1，使得循环条件i<3成立，继续执行循环体

第2次循环

执行p[i]=a[i\*(i+1)]; 即p[1]=a[1\*(1+1)]=a[2]=2

计算表达式3，即i++，i为2，使得循环条件i<3成立，继续执行循环体

第3次循环

执行p[i]=a[i\*(i+1)]; 即p[2]=a[2\*(2+1)]=a[6]=6

计算表达式3，即i++，i为3，使得循环条件i<3不成立，结束循环

第三个for循环语句

初值i=0, 使得循环条件i<3成立，执行循环体

第1次循环

执行k+=p[i]\*2; 即k=5+p[0]\*2=5+0=5

计算表达式3，即i++，i为1，使得循环条件i<3成立，继续执行循环体

第2次循环

执行k+=p[i]\*2; 即k=5+p[1]\*2=5+2\*2=9

计算表达式3，即i++，i为2，使得循环条件i<3成立，继续执行循环体

第1次循环

执行k+=p[i]\*2; 即k=9+p[2]\*2=9+6\*2=21

计算表达式3，即i++，i为3，使得循环条件i<3不成立，结束循环

4.有如下程序：

**#include <stdio.h>**

**int m[3][3]={{1},{2},{3}};**

**int n[3][3]={1,2 ,3};**

**main( )**

**{ printf(“%d,”, m[1][0]+n[0][0]);**

**printf(“%d\n”,m[0][1]+n[1][0]);**

**}**

**运行结果为：**

**3,0**

首先定义整型二维数组m，m为3行，3列的二维矩阵，并对其以行的形式初始化

m[0][0]=1 m[0][1]=0 m[1][2]=0

m[1][0]=2 m[1][1]=0 m[2][2]=0

m[2][0]=3 m[2][1]=0 m[2][2]=0

定义整型二维数组n，m为3行，3列的二维矩阵

n[0][0]=1 n[0][1]=2 n[1][2]=3

n[1][0]=0 n[1][1]=0 n[2][2]=0

n[2][0]=0 n[2][1]=0 n[2][2]=0

因此 m[1][0]+n[0][0]=2+1=3

m[0][1]+n[1][0]=0+0=0

5.有如下程序：

**#include <stdio.h>**

**main()**

**{ int i;**

**int x[3][3]={1,2,3,4,5,6,7,8,9};**

**for (i=1; i<3; i++)**

**printf("%d ",x[i][3-i]);**

**}**

**运行结果为：**

**6 8**

首先按存储顺序为数组x初始化

x[0][0]=1 x[0][1]=2 x[0][2]=3

x[1][0]=4 x[1][1]=5 x[1][2]=6

x[2][0]=7 x[2][1]=8 x[2][2]=9

初值i=1, 使得循环条件i<3成立，执行循环体

第1次循环

执行printf("%d ",x[i][3-i])，打印出x[i][3-i]，即x[1][2]的值

计算表达式3，即i++，i为2，使得循环条件i<3成立，继续执行循环体

第2次循环

执行printf("%d ",x[i][3-i])，打印出x[i][3-i]，即x[2][1]的值

计算表达式3，即i++，i为3，使得循环条件i<3成立，结束循环

6.有如下程序：

**#include <stdio.h>**

**main( )**

**{ int n[3][3], i, j；**

**for(i=0；i<3；i++ )**

**{for(j=0；j<3；j++ )**

**{n[i][j]=i+j；**

**printf(“%d ”, n[i][j])；**

**}**

**}**

**}**

**运行结果为：**

**0 1 2**

**1 2 3**

**2 3 4**

循环变量i为0, 循环条件 i<3成立，执行循环体

外层for第1次循环 相当于输出第1行

内层for循环j初值为0，循环条件j<3成立，执行循环体

内层for第1次循环

执行n[i][j]=i+j；即n[0][0]=0+0=0；

执行printf(“%d ”, n[i][j])；

执行内层循环表达式3，j++，j为1，j<3成立，继续执行内层循环体

内层for第2次循环

执行n[i][j]=i+j；即n[0][1]=0+1=1；

执行printf(“%d ”, n[i][j])；

执行内层循环表达式3，j++，j为2，j<3成立，继续执行内层循环体

内层for第3次循环

执行n[i][j]=i+j；即n[0][2]=0+2=2；

执行printf(“%d ”, n[i][j])；

执行内层循环表达式3，j++，j为3，j<3不成立，结束内层循环

执行printf(“\n”);

执行外层for语句的表达式3，i++，i为,1， i<3成立，继续执行外层循环体

外层for第2次循环 相当于输出第2行

内层for循环j初值为0，循环条件j<3成立，执行循环体

内层for第1次循环

执行n[i][j]=i+j；即n[1][0]=1+0=1；

执行printf(“%d ”, n[i][j])；

执行内层循环表达式3，j++，j为1，j<3成立，继续执行内层循环体

内层for第2次循环

执行n[i][j]=i+j；即n[1][1]=1+1=2；

执行printf(“%d ”, n[i][j])；

执行内层循环表达式3，j++，j为2，j<3成立，继续执行内层循环体

内层for第3次循环

执行n[i][j]=i+j；即n[1][2]=1+2=3；

执行printf(“%d ”, n[i][j])；

执行内层循环表达式3，j++，j为3，j<3不成立，结束内层循环

执行printf(“\n”);

执行外层for语句的表达式3，i++，i为,1， i<3成立，继续执行外层循环体

外层for第2次循环 相当于输出第3行

内层for循环j初值为0，循环条件j<3成立，执行循环体

内层for第1次循环

执行n[i][j]=i+j；即n[2][0]=2+0=1；

执行printf(“%d ”, n[i][j])；

执行内层循环表达式3，j++，j为1，j<3成立，继续执行内层循环体

内层for第2次循环

执行n[i][j]=i+j；即n[2][1]=2+1=2；

执行printf(“%d ”, n[i][j])；

执行内层循环表达式3，j++，j为2，j<3成立，继续执行内层循环体

内层for第3次循环

执行n[i][j]=i+j；即n[2][2]=2+2=3；

执行内层循环表达式3，j++，j为3，j<3不成立，结束内层循环

执行printf(“\n”);

执行外层for语句的表达式3，i++，i为,3， i<3不成立，结束外层循环

7.有如下程序：

**#include <stdio.h>**

**main()**

**{**

**char diamond[][5]={{‘\_’,’\_’,’\*’},{‘\_‘,’\*’,’\_’,’\*’},**

**{‘\*’,’\_’,’\_‘,’\_‘,’\*’},{‘\_’,’\*’,’\_’,’\*’},{‘\_’,’\_’,’\*’}};**

**int i,j;**

**for(i=0;i<5;i++)**

**{**

**for(j=0;j<5;j++)**

**printf(“%c”,diamond[i][j]);**

**printf(“\n”);**

**}**

**}注：“\_”代表一个空格。**

**运行结果为：**

**\***

**\* \***

**\* \***

**\* \***

**\***

8.有如下程序：

**#include <stdio.h>**

**main( )**

**{ int i, f[10];**

**f[0]=f[1]=1;**

**for(i=2;i<10;i++)**

**f[i]=f[i-2]+f[i-1];**

**for(i=0;i<10;i++)**

**{ if(i%4==0)**

**printf(“\n”);**

**printf(“%d ”,f[i]);**

**}**

**}**

**运行结果为：**

**1 1 2 3**

**5 8 13 21**

**34 55**

9.有如下程序：

**#include “stdio.h”**

**func(int b[ ])**

**{ int j;**

**for(j=0;j<4;j++)**

**b[j]=j;**

**}**

**main( )**

**{ int a[4], i;**

**func(a);**

**for(i=0; i<4; i++)**

**printf(“%2d”,a[i]);**

**}**

**运行结果为：**

**0 1 2 3**

定义函数func

函数头：未定义函数的类型，则系统默认为int型。函数func的形参为整型数组名，即只接收整型数组地址。

函数体：定义整型变量j

循环变量初值（表达式1）j=0, 使得循环条件（表达式2）j<4成立，执行循环体

第1次循环

执行b[j]=j; 即b[0]=0;

执行循环变量自增（及表达式3） j++，j为1，使得j<4成立，继续执行循环体

第2次循环

b[1]=1；

j++，j为2，使得j<4成立，继续执行循环体

第3次循环

b[2]=2；

j++，j为3，使得j<4成立，继续执行循环体

第4次循环

b[3]=3；

j++，j为4，使得j<4不成立，结束循环

main函数：

定义整型变量i和数组a，其长度为4，

func(a);表示调用函数func，并以数组名a作为调用的实参（数组名在C语言中表示数组所在内存空间的首地址，在以数组名作为实参时，形参与实参公用存储空间，因此对数组b的操作，即对数组a的操作。）

10.有如下程序：

**#include <stdio.h>**

**main（）**

**{float fun(float x[])；**

**float ave,a[3]={4.5，2，4}；**

**ave=fun（a）;**

**printf(“ave=%7.2f”,ave);**

**}**

**float fun（float x[]）**

**{int j;**

**float aver=1；**

**for (j=0;j<3;j++)**

**aver=x[j]\*aver;**

**return（aver）；**

**}**

**运行结果为：**

**ave= 36.00**

11.有如下程序：

**#include <stdio.h>**

**main()**

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

**int b[3][2],i,j;**

**for(i=0;i<=1;i++)**

**{for(j=0;j<=2;j++)**

**b[j][i]=a[i][j];**

**}**

**for(i=0;i<=2;i++)**

**{for(j=0;j<=1;j++)**

**printf("%5d",b[i][j]);**

**}**

**}**

**运行结果为：**

**1 4 2 5 3 6**

12.有如下程序：

**#include <stdio.h>**

**f(int b[],int n)**

**{int i,r;**

**r=1;**

**for (i=0;i<=n;i++)**

**r=r\*b[i];**

**return (r);**

**}**

**main()**

**{int x,a[]={1,2,3,4,5,6,7,8,9};**

**x=f(a,3);**

**printf(“%d\n”,x);**

**}**

**运行结果为：**

**24**

13.有如下程序：

**#include"stdio.h"**

**main()**

**{int j,k;**

**static int x[4][4],y[4][4];**

**for(j=0;j<4;j++)**

**for(k=j;k<4;k++)**

**x[j][k]=j+k;**

**for(j=0;j<4;j++)**

**for(k=j;k<4;k++)**

**y[k][j]=x[j][k];**

**for(j=0;j<4;j++)**

**for(k=0;k<4;k++)**

**printf("%d,",y[j][k]);**

**}**

**运行结果为：**

**0,0,0,0,1,2,0,0,2,3,4,0,3,4,5,6**

14.下面程序的功能是在a数组中查找与x值相同的元素所在位置,请填空。

　　 #include <stdio.h>

void main()

{ int a[10],i,x;

printf(“input 10 integers：”);

　　 　for(i=0;i<10;i++)

scanf(“%d”,&a[i]);

printf(“input the number you want to find x：”);

scanf(“%d”, &x );

for(i=0;i<10;i++)

if( x==a[i] )

break;

if( i<10 )

printf(“the pos of x is：%d\n”,i);

else printf(“can not find x！\n”);

}

15.以下程序的输出结果是\_\_\_\_\_\_。

main ()

{int a[3][3]={{1,2,9},{3,4,8},{5,6,7}},i,s=0;

for(i=0;i<3;i++) s+=a[i][i]+a[i][3-i-1];

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

}

答案：30

16.以下程序的输出结果是\_\_\_\_\_\_。

main()

{ char s[]="abcdef";

s[3]='\0';

printf("%s\n",s);

}

答案：abc

17.下列程序段的输出结果是\_\_\_\_\_\_。

main()

{ char b[]="Hello,you"

b[5]=0;

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

}

答案：Hello

18. 有一个3\*4矩阵，求其中的最大元素。

19.用“起泡法”对输入的10个字符排序后按从小到大的次序输出。

20.以下程序是一个函数，功能是求二阶矩阵（m行n列矩阵）的所有靠外侧的各元素值之和。（本程序中二阶矩阵用一维数组来表示。）

例如,矩阵为：

3 0 0 3

2 5 7 3

1 0 4 2

则所有靠外侧的各元素值之和为3+0+0+3+2+3+1+0+4+2=18。

add(int m,int n,int arr[])

{ int i,j,sum=0;

for(i=0;i<m;i=i+m-1)

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

sum=sum+ (7) ;

for(j=0;j<n;j=j+n-1)

for(i=1; (8) ;i++)

sum=sum+arr[i\*n+j];

return(sum);

}

**6.3 程序改错**

1.求一个数组中最大值及其下标。

**main( )**

**{ int max,j,m;**

**int a[5];**

**for(j=1;j<=5;j++)** // j=1为循环变量j赋初值为1，同时用j作为数字元素的逻辑地址下标。因此输出的时候只能从a[1]开始输出，无法输出a[0].因此应将j赋初值0，相应的循环条件改为j<5或者j<=4用于控制循环执行5次

**scanf(“%d”,a);** //若scanf语句有错误，常见考点有两个：（1）格式声明符号要与后面欲赋值的变量的类型一致，此题%d与a的类型int一致（详见p69-78）；（2）变量的前面要有地址符号&

**max=a[0];**

**for(j=1;j<=5;j++)** //修改思路与上一个for语句同

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

**{ max=a[j];**

**m=j;**

**}**

**printf(“下标：%d\n最大值:%d”, j, max)** //j为for语句的循环变量，当for语句执行完之后，j中的值为6，并非最大值下标，在执行某一次循环的比较过程中，将当时最大值的下标存在了m里

**}**

第四行改为：for(j=0;j<5;j++)

第五行改为：scanf(“%d”,&a[j]);

第七行改为：for(j=1;j<5;j++)

第八行改为：if(max<a[j])

第十三行改为：printf(“下标：%d\n最大值:%d”, m,max)

1. **CH7 函数**

**7.1 选择题**

1.（ A ）是构成C语言程序的基本单位。

A、函数 B、过程 C、子程序 D、子例程

2.C语言程序从 C 开始执行。

A) 程序中第一条可执行语句    B) 程序中第一个函数

   C) 程序中的main函数          D) 包含文件中的第一个函数

3.以下说法中正确的是（ C ）。

A、C语言程序总是从第一个定义的函数开始执行

B、在C语言程序中，要调用的函数必须在main( )函数中定义

C、C语言程序总是从main( )函数开始执行

D、C语言程序中的main( )函数必须放在程序的开始部分

4.下列关于C语言的说法错误的是（ B ） 。

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、选择、循环

D) C程序从main函数开始执行

5.C语言中函数返回值的类型是由 A 决定的.

A)函数定义时指定的类型 B) return语句中的表达式类型

C) 调用该函数时的实参的数据类型 D) 形参的数据类型

6.在C语言中，函数的数据类型是指( A )

A 函数返回值的数据类型 B. 函数形参的数据类型

C 调用该函数时的实参的数据类型 D.任意指定的数据类型

7.在函数调用时，以下说法正确的是（ B　）

　A.函数调用后必须带回返回值

　B.实际参数和形式参数可以同名

　C.函数间的数据传递不可以使用全局变量

　D.主调函数和被调函数总是在同一个文件里

8.在C语言中，表示静态存储类别的关键字是: （ C ）

A) auto B) register C) static D) extern

9.未指定存储类别的变量，其隐含的存储类别为（ A ）。

A)auto B)static C)extern D)register

10.以下叙述中正确的是\_\_B\_\_\_\_。

A、全局变量的作用域一定比局部变量的作用域范围大

B、静态(static)类别变量的生存期贯穿于整个程序的运行期间

C、函数的形参都属于全局变量

D、未在定义语句中赋初值的auto变量和static变量的初值都是随机值

11.下面程序的输出是\_\_\_C\_\_\_。

fun3(int x)

{

static int a=3;

a+=x;

return(a);

}

main()

{

int k=2, m=1, n;

n=fun3(k);

n=fun3(m);

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

}

A、3 B、4 C、6 D、9

12.以下叙述中正确的是\_\_\_A\_\_\_。

A、局部变量说明为static 存储类，其生存期将得到延长

B、全局变量说明为static 存储类，其作用域将被扩大

C、任何存储类的变量在未赋初值时，其值都是不确定的

D、形参可以使用的存储类说明符与局部变量完全相同

13.有以下程序

int fun(int x[],int n)

{

static int sum=0,i;

for(i=0;i<n;i++) sum+=x[i];

return sum;

}

main()

{

int a[]={1,2,3,4,5},b[]={6,7,8,9,},s=0;

s=fun(a,5)+fun(b,4);

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

}

程序执行后的输出结果是\_\_\_C\_\_\_。

A、45 B、50 C、60 D、55

14.以下程序执行后sum的值是\_\_\_C\_\_\_。

main()

{

int i,sum;

for(i=1;i<6;i++) sum+=i;

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

}

A、15 B、14 C、不确定 D、0

15.以下程序的输出结果是\_\_\_A\_\_\_。

main()

{

int k=4,m=1,p;

p=func(k,m); printf("%d,",p);

p=func(k,m); printf("%d\n",p);

}

func(int a,int b)

{

static int m=0,i=2;

i+=m+1; m=i+a+b;

return(m);

}

A、8,17 B、8,16 C、8,20 D、8,8

16.有以下程序

fun(int x, int y)

{

static int m=0,i=2;

i+=m+1;

m=i+x+y;

return m;

}

main()

{

int j=1,m=1,k;

k=fun(j,m); printf("%d,",k);

k=fun(j,m); printf("%d\n",k);

}

执行后的输出结果是\_\_B\_\_\_\_。

A、5,5 B、5,11 C、11,11 D、11,5

17.以下程序的输出结果是\_\_\_B\_\_\_。

int a,b;

void fun()

{

a=100;

b=200;

}

main()

{

int a=5,b=7;

fun();

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

}

A、100200 B、57 C、200100 D、75

18.有如下程序

int func(int a,int b)

{ return(a+b); }

main()

{

int x=2,y=5,z=8,r;

r=func(func(x,y),z);

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

}

该程序的输出结果是\_\_D\_\_\_\_。

A、12 B、13 C、14 D、15

19.有以下程序

void f(int v,int w)

{

int t;

t=v;v=w;w=t;

}

main()

{

int x=1,y=3,z=2;

if(x>y) f(x,y);

else if(y>z) f(y,z);

else f(x,z);

printf("%d,%d,%d\n",x,y,z);

}

执行后输出结果是\_\_\_C\_\_\_。

A、1,2,3 B、3,1,2 C、1,3,2 D、2,3,1

20.有以下程序

void f(int x,int y)

{

int t;

if(x<y) {t=x; x=y; y=t;}

}

main()

{

int a=4,b=3,c=5;

f(a,b); f(a,c); f(b,c);

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

}

执行后输出结果是\_\_\_\_\_\_。

A、3,4,5 B、5,3,4 C、5,4,3 D、4,3,5

21.有以下程序

int fun1(double a) { return a\*=a; }

int fun2(double x,double y)

{

double a=0,b=0;

a=fun1(x);

b=fun1(y);

return (int)(a+b);

}

main()

{

double w;

w=fun2(1.1,2.0);

}

程序执行后变量w中的值是\_\_\_C\_\_\_。

A、5.21 B、5 C、5.0 D、0.0

22.在调用函数时，如果实参是简单变量，它与对应形参之间的数据传递方式是\_\_\_B\_\_\_。

A、地址传递 B、单向值传递 C、由实参传给形参 D、再由形参传回实参

23.当执行下面的程序时，如果输入ABC，则输出结果是\_\_\_A\_\_\_。

#include "stdio.h"

#include "string.h"

main()

{

char ss[10]="12345";

gets(ss);

strcat(ss,"6789");

printf("%s\n",ss);

}

A、ABC6789 B、ABC67 C、12345ABC6 D、ABC456789

24.有如下程序

long fib(int n)

{

if(n>2) return(fib(n-1)+fib(n-2));

else return(2);

}

main()

{ printf("%ld\n",fib(3)); }

该程序的输出结果是\_\_\_B\_\_\_。

A、2 B、4 C、6 D、8

25.设函数fun的定义形式为

void fun(char ch,float x) {……}

则以下对函数fun的调用语句中，正确的是\_\_D\_\_\_\_。

A、fun("abc",3.0); B、t=fun('D',16.5); C、fun ('65',2.8); D、fun(32,32);

26.有以下程序

char fun(char x,char y)

{

if(x<y) return x;

return y;

}

main()

{

int a='9',b='8',c='7';

printf("%c\n",fun(fun(a,b),fun(b,c)));

}

程序的执行结果是\_\_\_D\_\_\_。

A、函数调用出错 B、8 C、9 D、7

27.若已定义的函数有返回值，则以下关于该函数调用的叙述中错误的是\_\_\_D\_\_\_。

A、函数调用可以作为独立的语句存在 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

28.有如下函数调用语句

func(rec1,rec2+rec3,(rec4,rec5));

该函数调用语句中，含有的实参个数是\_\_A\_\_\_\_。

A、3 B、4 C、5 D、有语法错

29.以下函数值的类型是\_\_\_A\_\_\_。

fun(float x)

{

float y;

y=3\*x-4;

return y;

}

A、int B、不确定 C、void D、float

30.在C语言中，形参的缺省存储类是\_\_\_A\_\_\_。

A、auto B、register C、static D、extern

31.有以下程序

int f(int a)

{ return a%2; }

main()

{

int s[8]={1,3,5,2,4,6},i,d=0;

for(i=0;f(s[i]);i++) d+=s[i];

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

}

程序运行后的输出结果是\_\_A\_\_\_\_。

A、9 B、11 C、19 D、21

32.在C语言中，函数的隐含存储类别是\_\_\_C\_\_\_。

A、auto B、static C、extern D、无存储类别

33.若程序中定义了以下函数

double myadd(double a,double b)

{ return (a+b); }

并将其放在调用语句之后，则在调用之前应该对该函数进行说明，以下选项中错误的说明是\_\_A\_\_\_\_。

A、double myadd(double a,b); B、double myadd(double,double);

C、double myadd(double b,double a); D、double myadd(double x,double y);

34.若有以下程序

#include <stdio.h>

void f(int n)

main()

{

void f(int n);

f(5);

}

void f(int n);

{ printf("%d\n",n); }

则以下叙述不正确的是\_\_\_C\_\_\_。

A、若只在主函数中对函数f进行说明，则只能在主函数中正确调用函数f

B、若在主函数前对函数f进行说明，则在主函数和其它函数中都可以正确调用函数f

C、对于以上程序，编译时系统会出提示出错信息：提示对f函数重复说明

D、函数f无返回值，所以可用void将其类型定义为无值型

35.下列叙述中正确的是\_\_\_D\_\_\_。

A、C语言编译时不检查语法 B、C语言的子程序有过程和函数两种

C、C语言的函数可以嵌套定义 D、C语言所有函数都是外部函数

36.在C语言程序中，以下说法正确的是（　）。

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都不可以嵌套

D函数的定义和函数的调用都可以嵌套

37.以下函数调用语句中含有（ ）个实参。

func((e1,e2),(e3,e4,e5));

A 2 B 3 C 5 D 语法错误

38.函数在定义时，省略函数类型说明符，则该函数值的类型为 。A

A. int B. float C. long D. double

39.以下 函数，真正地没有返回值。 B

A. int a(){int a=2;return (a);}

B. void b(){printf("c");}

C. int a(){int a=2;return a;}

D. 以上都是

40.在 C 语言中，有关函数的说法，以下正确的是 。 A

A. 函数可嵌套定义，也可嵌套调用

B. 函数可嵌套定义，但不可嵌套调用

C. 函数不可嵌套定义，但可嵌套调用

D. 函数不可嵌套定义，也不可嵌套调用

41.以下函数调用语句中，含有实参的个数为 。C

fun((2,3),(4,5+6,7));

A. 1 B. 2 C. 5 D. 6

42.函数调用可以在 。 D

A. 函数表达式中 B. 函数语句中

C. 函数参数中 D. 以上都是

43.被调函数返回给主调函数的值称为 。C

A. 形参 B. 实参 C. 返回值 D. 参数

44. ，可以不进行函数类型说明。 D

A. 被调函数的返回值是整型或字符型时

B. 被调函数的定义在主调函数定义之前时

C. 在所有函数定义前，已在函数外预先说明了被调函数类型

D. 以上都是

45.被调函数通过 语句，将值返回给主调函数。 D

A. if B. for C. while D. return

46.被调函数调用结束后，返回到 。 D

A.主调函数中该被调函数调用语句处

B.主函数中该被调函数调用语句处

C.主调函数中该被调函数调用语句的前一语句

D.主调函数中该被调函数调用语句的后一语句

47.以下对 C语言函数的有关描述中，正确的是 。A

A. 在 C 中，调用函数时，只能把实参的值传送给形参，形参的值不能传送给实参

B. C函数既可以嵌套定义又可递归调用

C. 函数必须有返回值，否则不能使用函数

D. C程序中有调用关系的所有函数必须放在同一个源程序文件中

48.C 语言中函数的隐含存储类型是 。 C

A. auto B. static C. extern D. 无存储类型

49.能把函数处理结果的两个数据返回给主调函数，在下面的方法中不正确的是 。A

A. return 这两个数 B. 形参用两个元素的数组

C. 形参用两个这种数据类型的指针 D. 用两个全局变量

50.C语言可执行程序从什么地方开始执行 ( )。 C

A. 程序中第一条可执行语句

B. 程序中第一个函数

C. 程序中的main函数

D. 包含文件中的第一个函数

51.有一个函数原型如下所示，则该函数的返回类型为( ) 。 C

abc(float x,float y);

A. void B. double C. int D. float

**7.2 读程序**

1.有如下程序：

**#include <stdio.h>**

**int Sub(int a, int b)**

**{return (a- b);}**

**main()**

**{int x, y, result = 0;**

**scanf("%d,%d", &x,&y );**

**result = Sub(x,y ) ;**

**printf("result = %d\n",result);**

**}**

**当从键盘输入:6,3运行结果为：**

**result =3**

2.有如下程序：

**#include <stdio.h>**

**int min( int x, int y )**

**{ int m；**

**if ( x> y ) m = x；**

**else m = y；**

**return(m)；**

**｝**

**main() {**

**int a=3,b=5,abmin ；**

**abmin = min(a,b)；**

**printf(“min is %d”， abmin)；**

**}**

**运行结果为：**

**min is 5**

3.有如下程序：

**#include<stdio.h>**

**func(int x) {**

**x=10;**

**printf(“%d, ”,x);**

**}**

**main( )**

**{ int x=20;**

**func(x);**

**printf(“%d”, x);**

**}**

**运行结果为：**

**10, 20**

在main函数中调用函数func，main函数将20作为实参穿给func，并转向开始执行func.

main()对应的内存 func（）对应的内存

20 x

20 x

func()执行x=10;，其内存中x变为10.

10 x

func()执行printf(“%d, ”,x); 即输出func函数对应内存中x的值，输出的是10. 至此，func函数执行结束，返回main函数。

main函数执行printf(“%d”, x);此时输出main函数对应内存中的x，即20

4.有如下程序：

**#include <stdio.h>**

**int m=4;**

**int func(int x,int y)**

**{ int m=1;**

**return(x\*y-m);**

**}**

**main()**

**{int a=2,b=3;**

**printf("%d\n",m);**

**printf("%d\n",func(a,b)/m);**

**}**

**运行结果为：**

**4**

**1**

整型变量m在函数外定义，因此m为全局变量，其作用于范围为其定义位置开始，一直到整个程序结束。因此func与main函数都可以访问m

程序首先执行main函数

执行printf("%d\n",m); 即输出m中的值4，并换行。

执行printf("%d\n",func(a,b)/m);即输出表达式func(a,b)/m的值，为了计算该表达式，

需要调用函数func。此时main将a,b中的2和3值作为实参传递给func的x和y

程序开始转向执行func函数，此时func中的x为2，y为3

执行int m=1; 此句定义了一个局部变量m并赋值为1 。m的作用域为其所在的复合

语句，即func的函数体，因此在func的函数体重，有限访问局部变量m。

执行return(x\*y-m); 即return（2\*3-1） ;返回的是整数5.

func函数返回至main函数中的被调用处

main函数中func(a,b)的值为5，func(a,b)/m=5/4=1,注意，在main函数中访问的m为全局变量m，此时main函数无法访问func中的m，因为不在func中m的作用域。

5.有如下程序：

**#include <stdio.h>**

**int fun(int a, int b)**

**{ if(a>b) return(a);**

**else return(b);**

**}**

**main()**

**{ int x=15, y=8, r;**

**r= fun(x,y);**

**printf("r=%d\n", r);**

**}**

**运行结果为：r=15**

程序首先执行main函数

执行r= fun(x,y);即将func(x,y)的值赋给r，为了计算该表达式，需要调用函数func。此时main将x,y中的15和8值作为实参传递给func的a和b

程序开始转向执行func函数，此时func中的a为15，b为8

执行if语句;判断if后面的表达式，a>b成立，因此执行相应的操作return(a)；即返回

a的值 。

func函数返回至main函数中的被调用处

main函数中func(x,y)的值为15，即将15赋给r。

执行printf("r=%d\n", r); 即输出r=15

6.有如下程序：

**#include <stdio.h>**

**int fac(int n)**

**{ int f=1,i;**

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

**f=f \* i;**

**return(f);**

**}**

**main()**

**{ int j,s;**

**scanf(“%d”,&j);**

**s=fac(j);**

**printf("%d!=%d\n",j,s);**

**}**

**如果从键盘输入3， 运行结果为： 3!=6**

程序首先执行main函数

执行r= fun(x,y);即将func(x,y)的值赋给r，为了计算该表达式，需要调用函数func。此时main将x,y中的15和8值作为实参传递给func的a和b

程序开始转向执行func函数，此时func中的a为15，b为8

执行if语句;判断if后面的表达式，a>b成立，因此执行相应的操作return(a)；即返回

a的值 。

func函数返回至main函数中的被调用处

main函数中func(x,y)的值为15，即将15赋给r。

执行printf("r=%d\n", r); 即输出r=15

7.有如下程序：

**#include <stdio.h>**

**unsigned fun6(unsigned num)**

**{ unsigned k=1;**

**do**

**{ k\*=num%10;**

**num/=10;**

**}while(num);**

**return k;**

**}**

**main()**

**{ unsigned n=26;**

**printf(“%d\n”,fun6(n));**

**}**

**运行结果为：12**

程序首先执行main函数

执行printf(“%d\n”,fun6(n)); 即输出表达式func(6)的值，为了计算该表达式，需要调用

函数func。此时main将n中的26作为实参传递给func的num

程序开始转向执行func函数，此时func中的num为26

执行do-while语句

第1次循环

执行k\*=num%10,即k=k\*(num%10)=1\*(26%10)=6

执行num/=10;即num=num/10=26/10=2

while后面循环条件为num,此时num为2，是非0值，即表示循环条件成立，

继续执行循环体。此时k为6

第2次循环

执行k\*=num%10,即k=k\*(num%10)=6\*(2%10)=12

执行num/=10;即num=num/10=2/10=0

while后面循环条件为num,此时num为0，表示循环条件不成立，

结束循环

执行return k; 即返回至main函数中的被调用处

执行main函数

继续执行printf(“%d\n”,fun6(n)); 即输出12

8.有如下程序：

**#include <stdio.h>**

**int max(int x, int y);**

**main()**

**{ int a,b,c;**

**a=7;b=8;**

**c=max(a,b);**

**printf("Max is %d",c);**

**}**

**max(int x, int y)**

**{ int z;**

**z=x>y? x : y;**

**return(z) ;**

**}**

**运行结果为：**

**Max is 8**

9.有如下程序：

int f(int x,int y)

{ return(y-x)\*x; }

main()

{ int a=3,b=4,c=5,d;

d=f(f(3,4),f(3,5));

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

}

执行后输出结果是\_\_\_\_\_\_。

答案：9

10.以下函数的功能是:求x的y次方，请填空。

double fun(double x,int y)

{ int i;

double z;

for(i=1,z=x;i<y;i++) z=z\*\_\_\_\_\_\_;

return z;

}

答案：x

11.有如下程序：

main()

{ printf(“main( ) :%d\n”,fact(5));

fact(-5);

}

fact(int value)

{ int f;

if(value<0)

{ printf(“Arg error\n");

return(-1);

}

else if(value==1||value==0) f=1;

else f=value\*fact(value-1)+1;

printf(“This called value=%d\n”,f);

return f;

}

**7.3 程序改错**

1.用一个函数求两个数之和。

**sum(x,y)** //函数定义的一般形式p173-174

**{ float z;**

**z=x+y;**

**return;** //return语句后面可以返回0、常量、变量和表达式的值。

**}**

**main()**

**{ float a,b;**

**int c;** / /若定义变量的语句有错误，常见考点有两个：（1）变量的类型，（2）在定义用于存放运算结果的变量时，一定要赋初值。一般赋值0或者循环初值。

**scanf("%f,%f",&a,&b);**

**c=sum(a,b);**

**printf("\nSum is %f",sum);**

**}**

第一行改为：float sum(float x, float y );

第四行改为：return(z);或者return z;

第八行：float c;

第十一行：printf("\nSum is %f",c);

2.求表达式的值

**#include <stdio.h>**

**#include <math.h>**

**int fun(int x, int y);**

**main()**

**{ int a,b; float f;**

**scanf(“%d,%d”,a,b);** //与改错第1题中的scanf错误相同

**if(ab>0){** // C语言中乘号不能省略，且用\*表示乘法运算

**fun(a,b);** // 调用带有返回值的函数，应将函数的返回值保存在变量里

**printf("The result is:%d\n", &f)**  //与第6题中printf错误相同

**}**

**else printf(“error!”);}**

**fun(x, y)**  // 定义函数的一般形式p173-174

**{ float result;**

**result = sqrt(a+b);**

**return;**  //return语句后面可以返回0、常量、变量和表达式的值。

**}**

第一处改为：if(a\*b>0)

第二处改为：f= fun(a,b);

第三处改为：printf("The result is:%d\n",f);

第四处改为：float fun(int x, int y)

第五处改为：f= fun(a,b);

第六处改为：result = sqrt(a\*b);

第七处改为：return result;

1. **CH8指针**

**8.1 选择题**

1.设有如下定义：

struck sk

{ int a;

float b;

}data;

int \*p;

若要使P指向data中的a域，正确的赋值语句是（ C ）。

A) p=&a; B) p=data.a; C) p=&data.a; D)\*p=data.a;

2.语句int \*p;说明了 C 。

A)p是指向一维数组的指针

B)p是指向函数的指针,该函数返回一int型数据

C)p是指向int型数据的指针 // 指针的定义

D)p是函数名,该函数返回一指向int型数据的指针

3.下列不正确的定义是（　A ）。

1. int \*p=&i,i; B.int \*p,i;

C．int i,\*p=&i; D.int i,\*p;

选项A先定义一个整型指针变量p，然后将变量i的地址赋给p。然而此时还未定义变量i因此编译器无法获得变量i的地址。(A与C对比，选项C先定义变量i，则在内存中为i分配空间，因此i在内存空间的地址就可以确定了；然后再定义p，此时可以为p赋i的地址，C正确)

4.若有说明：int n=2,\*p=&n,\*q=p,则以下非法的赋值语句是: （　D ）

A）p=q B）\*p=\*q C）n=\*q D）p=n

p,q同为整型指针变量，二者里面仅能存放整型变量的地址。

选项A，q中为地址，因此可将此地址赋给p

选项B，\*p表示p所指向对象n的内容，即一个整数；\*q表示q所指向对象的内容，由于在定义q时为其初始化，将p中n的地址给q，因此p中存放n的地址，\*q表示q所指向对象n的内容.因此\*p=\*q 相当于 n=n;

选项C，n=\*q 等价于n=n;

选项D，p中只能存放地址，不能将n中的整数值赋给p

5.有语句：int a[10],;则 B 是对指针变量p的正确定义和初始化。

A)int p=\*a; B)int \*p=a; C)int p=&a; D)int \*p=&a;

选项A，a是数组名，不是指针变量名，因此不可用\*标注数组名a

选项C，a是数组名，数组名就是地址，无需再用地址符号。而且在定义指针变量p时，应在变量名前加\*，标明p是指针变量

选项D，a是数组名，数组名就是地址，无需再用地址符号。

6.若有说明语句“int a[5],\*p=a;”,则对数组元素的正确引用是( C )。

A.a[p] B.p[a] C.\*(p+2) D.p+2

首先定义一个整型数组a，a的长度为5，然后定义一个指针变量p，并同时对p进行初始化，将数组a的地址赋给p。因此此时p中存放的数组a的首地址，即数组中第一个元素a[0]的地址。

对于数组元素下标的引用(详见p144), 一般形式 数组名[下标] 其中下标为逻辑地址下标，从0开始计数，方括号中的下标可以是变量，可以是表达式，但结果一定要是整数。

选项A，p中存放的是地址，不是整数，不能做数组元素的下标

选项B，a是数组名，数组名就是地址，不是整数，不能做数组元素的下标

选项C，(重点！！！详见p231~234) p+2表示指向同一数组中的下两个元素的地址，当前p指向a[0]，则p+2表示a[2]的地址，因此\*(p+2)表示a[2]的内容

7.有如下程序

int a[10]={1,2,3,4,5,6,7,8,9,10},\*P=a;

则数值为9的表达式是 B

A) \*P+9 B) \*(P+8) C) \*P+=9 D) P+8

(重点！！！详见p231~234)

首先定义一个整型数组a，a的长度为5，然后定义一个指针变量P，并同时对P进行初始化，将数组a的地址赋给P。因此此时P中存放的数组a的首地址，即数组中第一个元素a[0]的地址。

数组中9对应的是a[8], 选项B，P+8表示数组中后8个元素的地址，即a[8]的地址。\*(P+8)则表示该地址内所存放的内容，即a[8]的值。

选项A，\*P表示P所指向对象的内容，此时P指向a[0], \*P即a[0]的值1. \*P+9=1+9=10

选项C，\*P表示P所指向对象的内容，此时P指向a[0], \*P即a[0]的值。因此\*P+=9 即\*P =\*P+9, 等价于a[0]=a[0]+9.

选项D，P+8表示数组中后8个元素的地址，即a[8]的地址，而非a[8]中的值。

8.在C语言中，以 D 作为字符串结束标志

A)’\n’ B)’ ’ C) ’0’ D)’\0’

9.下列数据中属于“字符串常量”的是（　A ）。

A.“a” B.{ABC} C.‘abc\0’ D.‘a’

若干个字符构成字符串

在C语言中，用单引号标识字符；用双引号标识字符串

选项B，C，分别用{}和’’标识字符串

选项D，标识字符。

10.已知char x[]="hello", y[]={'h','e','a','b','e'};, 则关于两个数组长度的正确描述是 B .

A)相同 B)x大于y C)x小于y D)以上答案都不对

C语言中，字符串后面需要一个结束标志位'\0'，通常系统会自动添加。

对一维数组初始化时可采用字符串的形式（例如本题数组x），也可采用字符集合的形式（例如本题数组y）。在以字符串形式初始化时，数组x不尽要存储字符串中的字符，还要存储字符串后的结束标志位，因此数组x的长度为6；在以字符集合形式初始化时，数组y，仅存储集合中的元素，因此数组y长度为5

11.若有定义：int x,\*pb；则以下正确的表达式是（A）

A、pb=&x B、pb=x C、\*pb=&x D、\*pb=\*x

12.以下程序的输出结果是（B）

A、因变量无定义输出不定值 B、0 C、-1 D、1

#include<stdio.h>

void main()

{printf(“%d\n”,NULL);}

13.以下程序的输出结果是（B）

A、5，2，3

B、-5，-12，-7

C、-5，-12，-17

D、5，-2，-7

#include<stdio.h>

void sub(int x,int y,int \*z)

{\*z=y-x;}

void main()

{

int a,b,c;

sub(10,5,&a); sub(7,a,&b); sub(a,b,&c);

printf(“%d,%d,%d\n”,a,b,c);

}

14.以下程序的输出结果是（C）

A、4 B、6 C、8 D、10

#include<stdio.h>

void main()

{

int k=2,m=4,n=6;

int \*pk=&k,\*pm=&m,\*p;

\*(p=&n)=\*pk\*(\*pm);

printf(“%d\n”,n);

}

16.以下程序的输出结果是（D）。

A、23 B、24 C、25 D、26

#include<stdio.h>

void prtv(int \*x)

{printf(“%d\n”,++\*x);}

void main()  
{

int a=25;prtv(&a);

}

17.以下程序的输出结果是（B）。

A、运行出错 B、100 C、a的地址 D、b的地址

#include<stdio.h>

void main()

{

int \*\*k,\*a,b=100;

a=&b;k=&a;

printf(“%d\n”,\*\*k);

}

18.以下程序运行后,输出结果是（D）。

A) 8 B)7 C) 6 D) 5

#include<stdio,h>

ss(char \*s)

{ char \*p=s;

while(\*p) p++;

return(p-s);

}

main()

{ char \*a="abded";

int i;

i=ss(a);

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

}

**8.2 读程序**

1.有如下程序：

**# include < stdio .h >**

**main ( )**

**{ int x[ ] = {10, 20, 30, 40, 50 };**

**int \*p ;**

**p=x;**

**printf ( “%d”, \*(p+2 ) );**

**}**

**运行结果为：**

**30**

首先定义一个整型数组x，x的长度为5;然后定义一个指针变量p;对p进行初始化，将数组x的地址赋给p。因此此时p中存放的数组x的首地址，即数组中第一个元素x[0]的地址。

然后执行printf语句，输出表达式\*(p+2)的值。p+2表示以p当前指向的位置起始，之后第2个元素的地址，即a[2]的地址。\*(p+2)则表示该地址内所存放的内容，即a[2]的值30,因此输出30

2.有如下程序：

**#include <stdio.h>**

**main( )**

**{ char s[]=”abcdefg”;**

**char \*p;**

**p=s;**

**printf(“ch=%c\n”,\*(p+5));**

**}**

**运行结果为：**

**ch=f**

首先定义一个字符型数组s，并用字符串**abcdefg**对s进行初始化; 然后定义一个字符型指针变量p; 对p进行初始化，将数组s的地址赋给p。因此此时p中存放的数组s的首地址，即数组中第一个元素s[0]的地址。

然后执行printf语句，输出表达式\*(p+5)的值。p+5表示以p当前指向的位置起始，之后第5个元素的地址，即a[5]的地址。\*(p+5)则表示该地址内所存放的内容，即a[5]的值f, 因此输出ch=f

3.有如下程序：

**#include<stdio.h>**

**main ( )**

**{ int a[]={1, 2, 3, 4, 5} ；**

**int x, y, \*p；**

**p=a；**

**x=\*(p+2)；**

**printf("%d：%d \n", \*p, x)；**

**}**

**运行结果为:**

**1:3**

首先定义一个整型数组a，并对a进行初始化; 然后定义整型变量x,y，整型指针变量p; 再将数组a的地址赋给p。因此此时p中存放的数组a的首地址，即数组中第一个元素a[0]的地址。执行**x=\*(p+2)；**p+2表示以p当前所指向的位置起始，之后第2个元素的地址，即a[2]的地址。\*(p+2)则表示该地址内所存放的内容，即a[2]的值3，然后再把3赋给x

然后执行printf语句，先输出表达式\*p的值。此时\*p表示的是p所指向变量的内容，即a[0]的值1。再输出一个冒号。然后再输出x中的值3。

4.有如下程序：

**#include<stdio.h>**

**main()**

**{ int arr[ ]={30,25,20,15,10,5}, \*p=arr;**

**p++;**

**printf(“%d\n”,\*(p+3));**

**}**

**运行结果为：10**

首先定义一个整型数组arr，并对arr进行初始化; 然后定义整型指针变量p; 再将数组arr的地址赋给p。因此此时p中存放的数组arr的首地址，即数组中第一个元素a[0]的地址。

执行p++，即p=p+1。p+1表示以p当前所指向的位置起始，之后第1个元素的地址，即arr[1]的地址，然后再将arr[1]的地址赋给p，执行完此语句后，p不再指向arr[0]，而是指向arr[1]。

然后执行printf语句，输出表达式\*(p+3)的值。p+3表示以p当前指向的位置起始(此时p指向arr[1])，之后第3个元素的地址，即arr[4]的地址。\*(p+3)则表示该地址内所存放的内容，即arr[4]的值10, 因此输出10

5.有如下程序：

**#include <stdio.h>**

**main( )**

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

**int x, y, \*p;**

**p = &a[0];**

**x = \*(p+2);**

**y = \*(p+4);**

**printf(“\*p=%d, x=%d, y=%d\n”, \*p, x, y);**

**}**

**运行结果为：**

**\*p=1, x=3, y=5**

首先定义一个整型数组a，并对a进行初始化; 然后定义整型变量x,y，整型指针变量p; 再将数组元素a[0]的地址赋给p。

执行x=\*(p+2)；p+2表示以p当前所指向的位置起始，之后第2个元素的地址，即a[2]的地址。\*(p+2)则表示该地址内所存放的内容，即a[2]的值3，然后再把3赋给x

执行y = \*(p+4); p+4表示以p当前所指向的位置起始，之后第4个元素的地址，即a[4]的地址。\*(p+4)则表示该地址内所存放的内容，即a[4]的值5，然后再把5赋给y

执行printf语句，先输出表达式\*p的值。此时\*p表示的是p所指向变量的内容，即a[0]的值1。再输x的值3。再输出y的值5。

6.有如下程序：

**#include<stdio.h>**

**main( )**

**{ static char a[ ]=”Program”, \*ptr;**

**for(ptr=a, ptr<a+7; ptr+=2)**

**putchar(\*ptr);**

**}**

**运行结果为：**

**Porm**

首先定义一个字符型数组a，并对a进行初始化; 然后定义字符型指针变量p;

执行for语句 ptr=a为表达式1，将数字a的地址赋给ptr；表达式2（循环条件）ptr<a+7；表达式3为ptr+=2，即ptr= ptr+2；

第1次执行循环体

执行putchar(\*ptr); 即输出\*ptr所对应的字符。此时ptr指向数组中的第1个元素，即

a[0],因此\*ptr表示a[0]中的值，即’P’.

执行完循环体，转向执行表达式3，即ptr= ptr+2。ptr+2表示以ptr当前所指向的位

置起始，之后第2个元素的地址,即a[2]的地址，然后将a[2]的地址赋给ptr。a[2]的地

址等价于a+2，因此循环条件ptr<a+7成立，继续执行循环体

第2次执行循环体

执行putchar(\*ptr); 即输出\*ptr所对应的字符。此时ptr指向数组中的第3个元素，即

a[2],因此\*ptr表示a[2]中的值，即’o’.

执行完循环体，转向执行表达式3，即ptr= ptr+2。ptr+2表示以ptr当前所指向的位

置起始，之后第2个元素的地址,即a[4]的地址，然后将a[4]的地址赋给ptr。a[4]的地

址等价于a+4，因此循环条件ptr<a+7即a+4<a+7成立，继续执行循环体

第3次执行循环体

执行putchar(\*ptr); 即输出\*ptr所对应的字符。此时ptr指向数组中的第5个元素，即

a[4],因此\*ptr表示a[4]中的值，即’r’.

执行完循环体，转向执行表达式3，即ptr= ptr+2。ptr+2表示以ptr当前所指向的位

置起始，之后第2个元素的地址,即a[6]的地址，然后将a[6]的地址赋给ptr。a[6]的地址等价于a+6，因此循环条件ptr<a+7即a+6<a+7成立，继续执行循环体

第4次执行循环体

执行putchar(\*ptr); 即输出\*ptr所对应的字符。此时ptr指向数组中的第7个元素，即

a[6],因此\*ptr表示a[6]中的值，即’m’.

执行完循环体，转向执行表达式3，即ptr= ptr+2。ptr+2表示以ptr当前所指向的位

置起始，之后第2个元素的地址,即a[8]的地址，然后将a[8]的地址赋给ptr。a[6]的地址等价于a+8，因此循环条件ptr<a+7即a+8<a+7不成立，结束循环。

7.有如下程序：

**#include <stdio.h>**

**char s[]=”ABCD”;**

**main()**

**{ char \*p;**

**for(p=s;p<s+4;p++)**

**printf(“%c %s\n”,\*p,p);**

**}**

**运行结果为：**

**A ABCD**

**B BCD**

**C CD**

**D D**

首先定义一个字符型数组s，并对s进行初始化; 数组s是全局变量，其有效范围从其定义开始至整个程序结束。

执行main函数

定义一个字符型指针p。

执行for语句 p=s为表达式1，将数字s的首地址赋给p；表达式2（循环条件）p<s+4；表达式3为p++，即p= p+1；

第1次执行循环体

执行printf(“%c %s\n”,\*p,p);即以字符%c形式输出\*p所对应的字符。此时p指

向数组中的第1个元素，即s[0],因此\*p表示a[0]中的值，即’A’. 然后再以字

符串%s的形式输出以p中地址为首地址的整个字符串，即输出ABCD

执行完循环体，转向执行表达式3，即p= p+1。p+1表示以p当前所指向的

位置起始，之后1个元素的地址,即s[1]的地址，然后将a[1]的地址赋给p。

s[1]的地址等价于s+1，因此循环条件p<s+4成立，继续执行循环体

第2次执行循环体

执行printf(“%c %s\n”,\*p,p);即以字符%c形式输出\*p所对应的字符。此时p指

向数组中的第2个元素，即s[1],因此\*p表示s[1]中的值，即’B’. 然后再以字

符串%s的形式输出以p中地址为首地址的整个字符串，此时p指向s[1]，即

从s[1]开始，依次输出后面的字符串，因此又输出BCD

执行完循环体，转向执行表达式3，即p= p+1。p+1表示以p当前所指向的

位置起始，之后1个元素的地址,即s[2]的地址，然后将a[2]的地址赋给p。

s[2]的地址等价于s+2，因此循环条件p<s+4成立，继续执行循环体

第3次执行循环体

执行printf(“%c %s\n”,\*p,p);即以字符%c形式输出\*p所对应的字符。此时p指

向数组中的第3个元素，即s[2],因此\*p表示s[2]中的值，即’C’. 然后再以字

符串%s的形式输出以p中地址为首地址的整个字符串，此时p指向s[2]，即

从s[2]开始，依次输出后面的字符串，因此又输出CD

执行完循环体，转向执行表达式3，即p= p+1。p+1表示以p当前所指向的

位置起始，之后1个元素的地址,即s[2]的地址，然后将s[2]的地址赋给p。

s[2]的地址等价于s+3，因此循环条件p<s+4成立，继续执行循环体

第4次执行循环体

执行printf(“%c %s\n”,\*p,p);即以字符%c形式输出\*p所对应的字符。此时p指

向数组中的第4个元素，即s[3],因此\*p表示s[3]中的值，即’D’. 然后再以字

符串%s的形式输出以p中地址为首地址的整个字符串，即输出D

执行完循环体，转向执行表达式3，即p= p+1。p+1表示以p当前所指向的

位置起始，之后1个元素的地址,即s[3]的地址，然后将s[3]的地址赋给p。

s[3]的地址等价于s+4，因此循环条件p<s+4不成立，结束循环

8.用指针作函数参数，编程序求一维数组中的最大和最小的元素值。

#define N 10

main()

{ void maxmin(int arr[],int \*pt1,int \*pt2,int n);

int array[N]={10,7,19,29,4,0,7,35,-16,21},\*p1,\*p2,a,b;

p1=&a; p2=&b;

maxmin(array,p1,p2,N);

printf("max=%d,min=%d",a,b);

}

void maxmin(int arr[],int \*pt1,int \*pt2,int n)

{ int i;

\*pt1=\*pt2=arr[0];

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

{ if(arr[i]>\*pt1) (9) ;

if(arr[i]<\*pt2) (10) ;

}

}

1. **CH9 结构体**

**9.1 选择题**

1.若有以下说明语句：

struct student

{ int num;

char name[ ];

float score;

}stu;

则下面的叙述不正确的是: ( D )

A. struct是结构体类型的关键字

B. struct student 是用户定义的结构体类型

C. num, score都是结构体成员名

D. stu是用户定义的结构体类型名

2.若有以下说明语句：

struct date

{ int year;

int month;

int day;

}brithday;

则下面的叙述不正确的是\_\_C\_\_\_.

A) struct是声明结构体类型时用的关键字

B) struct date 是用户定义的结构体类型名

C) brithday是用户定义的结构体类型名

D) year,day 都是结构体成员名

3.以下对结构变量stul中成员age的非法引用是 B

struct student

{ int age；

int num；

}stu1,\*p；

p=&stu1；

A) stu1.age B) student.age C) p->age D) (\*p).age

4.设有以下说明语句：

typedef struct stu

{ int a;

float b;

} stutype;

则下面叙述中错误的是（ D ）。

A、struct是结构类型的关键字

B、struct stu是用户定义的结构类型

C、a和b都是结构成员名

D、stutype是用户定义的结构体变量名

5.设有以下定义，值为5的枚举常量是（ ）。

enum week{sun，mon=4，tue，wed，thu，fri，sat} w；

A tue B sat C fri D thu

6.设有如下定义:

struct sk

{int a;float b;}data,\*p;

若有p=&data;，则对data中的a域的正确引用是（B）。

A)(\*p).data.a B)(\*p).a C)p->data.a D)p.data.a

7.根据以下定义，能输出字母M的语句是（D）。

A、printf(“%c\n”,class[3].name) B、printf(“%c\n”,class[3].name[1]);

C、printf(“%c\n”,class[2].name[1]) D、printf(“%c\n”,class[2].name[0]);

struct person{char name[9];int age;};

struct person class[10]={“John”,17,”Paul”,19,”Mary”,18,”Adam”,16,};

8.以下程序的输出结果是（D）。

A、0 B、1 C、3 D、6

#include<stdio.h>

void main()

{

struct emplx{int x;int y;}enum[2]={1,3,2,7};

printf(“%d\n”,enum[0].y/enum[0].x\*enum[1].x);

}

9.若有以下说明及语句，则值为6的表达式是（D）。

A、p++->n B、p->n++ C、(\*p).n++ D、++p->n

struct st{int n;struct st \*next;};

struct st a[3],\*p;

a[0].n=5;a[0].next=&a[1];a[1].n=7;a[1].next=&a[2];

a[2].n=9;a[2].next=’\0’;p=&a[0];

10.已知字符0的ASCII码的十进制的值是48，且数组的第0个元素在低位，以下程序的输出结果是（B）。

A、39 B、9 C、38 D、8

void main()

{

union{int i[2];long k;char c[4];}r,\*s=&r;

s->i[0]=0x39;s->i[1]=0x38;

printf(“%x\n”,s->c[0]);

}

11.以下程序输出的结果是（C）。

A、32 B、16 C、8 D、24

typedef union{long x[2];int y[4];char z[8];}MYTYPE;

MYTYPE them;

void main()

{

printf(“%d\n”,sizeof(them));

}

12.设有以下语句

typedef struct S

{ int g; char h;} T;

则下面叙述中正确的是（B）。

A) 可用S定义结构体变量

B）可以用T定义结构体变量

C）S是struct类型的变量

D）T是struct S类型的变量

13.有以下说明和定义语句

struct student

{ int age; char num[8];};

struct student stu[3]={{20,"200401"},{21,"200402"},{10\9,"200403"}};

struct student \*p=stu;

以下选项中引用结构体变量成员的表达式错误的是（D）。

A) (p++)->num B) p->num

C) (\*p).num D) stu[3].age

**9.2 读程序**

1.有如下程序

**#include<stdio.h>**

**struct st**

**{ int x;**

**int y;**

**} a[2]={5, 7, 2, 9} ;**

**main()**

**{**

**printf("%d\n",a[0].y\*a [1].x);**

**}**

**运行结果是：**

**14**

首先是定义结构体st，st中共有两个整型成员x，y。

然后定义一个st类型的数组a，a的长度为2，即数组中含有两个st类型的元素，分别是a[0]和a[1]. 对a进行初始化，此题是按照储存顺序进行初始化，即将5赋给a[0]中的x（即a[0].x=5）; 将7赋给a[0]中的y（即a[0].y=7）; 将2赋给a[1]中的x（即a[1].x=2）; 将9赋给a[1]中的y（即a[1].y=9）;

执行main函数，输出表达式a[0].y\*a [1].x的值，即7\*2的值

**5 a[0].x a[0]**

**7 a[0].y**

**2 a[1].x a[1]**

1. **a[1].y**

2.有如下程序

**#include<stdio.h>**

**main( )**

**{struct stu**

**{int num;**

**char a[5];**

**float score;**

**}m={1234,”wang”,89.5};**

**printf(“%d,%s,%f”,m.num,m.a,m.score);**

**}**

**运行结果是：**

**1234,wang,89.5**

3.有如下程序

**#include<stdio.h>**

**struct cmplx**

**{ int x;**

**int y;**

**} cnum[2]={1, 3, 2, 7};**

**main( )**

**{**

**printf(“%d\n”, cnum[0].y \* cnum[1].x );**

**}**

**运行结果是：6**

**与第一题解法同**

4.有如下程序

**#include <stdio.h>**

**struct abc**

**{ int a, b, c; };**

**main()**

**{ struct abc s[2]={{1,2,3},{4,5,6}};**

**int t;**

**t=s[0].a+s[1].b;**

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

**}**

**运行结果是：6**

1. **CH10 文件**

1.系统的标准输入文件是指（ A ）。

a)键盘       b)显示器     c)软盘     d)硬盘

2.若执行fopen函数时发生错误，则函数的返回值是（ B ）。

a)地址值      b)0       c)1       d)EOF

3.若要用fopen函数打开一个新的二进制文件，该文件要既能读也能写，则文件方式字符串应是（ B ）。

a)"ab+"    b)"wb+"      c)"rb+"    d)"ab"

4．fscanf函数的正确调用形式是（ D ）。

a)fscanf(fp,格式字符串,输出表列)     b)fscanf(格式字符串，输出表列,fp);

c)fscanf(格式字符串,文件指针,输出表列); d)fscanf(文件指针，格式字符串,输入表列);

5.fgetc函数的作用是从指定文件读入一个字符，该文件的打开方式必须是（ C ）。

a)只写    b)追加   c)读或读写        d)答案b和c都正确

6．函数调用语句：fseek(fp,-20L,2);的含义是（ C ）

a)将文件位置指针移到距离文件头20个字节处

b)将文件位置指针从当前位置向后移动20个字节

c)将文件位置指针从文件末尾处后退20个字节

d)将文件位置指针移到离当前位置20个字节处

7．利用fseek函数可实现的操作（ A ）

a)fseek(文件类型指针，起始点，位移量); b)fseek(fp,位移量,起始点)；

c)fseek(位移量，起始点,fp);      d)fseek(起始点,位移量,文件类型指针);

8．在执行fopen函数时，ferror函数的初值是（ D ）

a)TURE      b)-1    c)1    d)0

9．fseek函数的正确调用形式是（ ）

A)fseek(文件指针,起始点,位移量)     B)fseek(文件指针,位移量,起始点)

C)fseek(位移量,起始点,文件指针)   D)fseek(起始点,位移量,文件指针)

10．若fp是指向某文件的指针，且已读到文件末尾，则函数feof(fp)的返回值是（ ）

A)EOF    B)-1     C)1      D)NULL

11．下列关于C语言数据文件的叙述中正确的是（ ）

A)文件由ASCII码字符序列组成，C语言只能读写文本文件

B)文件由二进制数据序列组成，C语言只能读写二进制文件

C)文件由记录序列组成，可按数据的存放形式分为二进制文件和文本文件

D)文件由数据流形式组成，可按数据的存放形式分为二进制文件和文本文件

12.函数fseek(pf, OL,SEEK\_END)中的SEEK\_END代表的起始点是（ ）

A)文件开始     B)文件末尾     C)文件当前位置    D)以上都不对

13.C语言中，能识别处理的文件为（ ）

A)文本文件和数据块文件       B)文本文件和二进制文件

C)流文件和文本文件          D)数据文件和二进制文件

14.若调用fputc函数输出字符成功，则其返回值是（ ）

A)EOF    B)1     C)0       D)输出的字符

15 已知函数的调用形式：fread(buf,size,count,fp),参数buf的含义是（ ）

A)一个整型变量，代表要读入的数据项总数 B)一个文件指针，指向要读的文件

C)一个指针，指向要读入数据的存放地址 D)一个存储区，存放要读的数据项

16 当顺利执行了文件关闭操作时，fclose函数的返回值是（ ）

A)-1    B)TRUE     C)0      D)1

17.如果需要打开一个已经存在的非空文件“Demo”进行修改下面正确的选项是（ ）

A)fp=fopen("Demo","r");     B)fp=fopen("Demo","ab+");

C)fp=fopen("Demo","w+");     D)fp=fopen("Demo","r+");

18．关于文件理解不正确的为（ ）

a)Ｃ语言把文件看作是字节的序列，即由一个个字节的数据顺序组成；

b)所谓文件一般指存储在外部介质上数据的集合；

c)系统自动地在内存区为每一个正在使用的文件开辟一个缓冲区；

d)每个打开文件都和文件结构体变量相关联，程序通过该变量访问该文件；

19．关于二进制文件和文本文件描述正确的为（ ）

a)文本文件把每一个字节放成一个ASCII代码的形式，只能存放字符或字符串数据；

b)二进制文件把内存中的数据按其在内存中的存储形式原样输出到磁盘上存放；

c)二进制文件可以节省外存空间和转换时间，不能存放字符形式的数据；

d)一般中间结果数据需要暂时保存在外存上，以后又需要输入内存的，常用文本文件保存。

20. 系统的标准输入文件操作的数据流向为（ ）

a)从键盘到内存   b)从显示器到磁盘文件  c)从硬盘到内存 d)从内存到U盘

21. 利用fopen (fname, mode)函数实现的操作不正确的为（ ）

a)正常返回被打开文件的文件指针, 若执行fopen函数时发生错误则函数的返回NULL；

b)若找不到由pname指定的相应文件，则按指定的名字建立一个新文件；

c)若找不到由pname指定的相应文件，且mode 规定按读方式打开文件则产生错误；

d)为pname指定的相应文件开辟一个缓冲区，调用操作系统提供的打开或建立新文件功能。

22．利用fwrite (buffer, sizeof(Student)，3, fp)函数描述不正确的（ ）

a) 将3个学生的数据块按二进制形式写入文件；

b) 将由buffer指定的数据缓冲区内的3\* sizeof(Student)个字节的数据写入指定文件；

c) 返回实际输出数据块的个数, 若返回0值表示输出结束或发生了错误；

d) 若由fp指定的文件不存在，则返回0值；

23．利用fread (buffer,size,count，fp)函数可实现的操作（ ）

a) 从fp指向的文件中，将count个字节的数据读到由buffer指出的数据区中；

b) 从fp指向的文件中，将size\*count个字节的数据读到由buffer指出的数据区中；

c) 以二进制形式读取文件中的数据，返回值是实际从文件读取数据块的个数count；

d) 若文件操作出现异常，则返回实际从文件读取数据块的个数；

24．检查由fp指定的文件在读写时是否出错的函数是（ ）a) feof()   b) ferror()    c) clearerr（fp） d) ferror(fp)

25.函数调用语句：fseek（fp，-10L，2）;的含义是（ ）。

A 将文件位置指针从文件末尾处向文件头的方向移动10个字节

B 将文件位置指针从当前位置向文件头的方向移动10个字节

C 将文件位置指针从当前位置向文件末尾方向移动10个字节

D 将文件位置指针移到距离文件头10个字节处

26.以下可以作为文件打开函数fopen中的第一个参数的正确格式是\_\_\_\_\_\_\_。

A.”file1.txt”    B.file1.txt     C. file1.txt,w      D.“file1.txt,w

27.若fp是指向某文件的指针，文件操作结束之后，关闭文件指针应使用下列（ ）语句

A）fp=fclose()；   B）fp=fclose；  C）fclose；   D）fclose(fp)；

28.函数rewind的作用是（ ）。

A) 使位置指针重新返回文件的开头

B) 将位置指针指向文件中所要求的特定位置

C) 使位置指针指向文件的末尾

D) 使位置指针自动移至下一个字符的位置

29.以下叙述中错误的是（ ）

A）C语言中对二进制文件的访问速度比文本文件快

B）C语言中，随机文件以二进制代码形式存储数据

C）语句 FILE  fp; 定义了一个名为fp的文件指针

D）C语言中的文本文件以ASCII码形式存储数据

30.以下与函数fseek(fp,0L,SEEK\_SET)有相同作用的是（ ）

A）feof(fp)   B）ftell(fp)  C）fgetc(fp)    D）rewind(fp)

1. **判断对错**

对的划“√”，错的划“×”

1.在Turbo C中，整型数据在内存中占2个字节。（ **×**）

2.int i=20;switch(i/10){case 2:printf("A");case 1:printf("B");}的输出结果为A。（ **×**）

3.break语句用在循环体中，可结束本层循环，continue语句用在循环体中，可结束本次循环。（ **√**）

4.函数的递归调用不过是一个函数直接或间接地调用它自身。（**√** ）

5.函数strlen("ASDFG\n")的值是7。（**×** ）

6.通过return语句，函数可以带回一个或一个以上的返回值。（ **×**）

7.结构体类型只有一种。 （ **×**）

8.char \*p="girl"；的含义是定义字符型指针变量p，p的值是字符串"girl"。（**×** ）

9.若有定义：char \*p(char a[10]);则p是函数名。（**√** ）

10.用fopen("file","r+");打开的文件"file"可以进行修改。 （ **√**）

11.参加位运算的数据可以是任何类型的数据。(**×**)

12.若有定义和语句：int a；char c;float f;scanf("%d,%c,%f",&a,&c,&f);若通过键盘输入：10,A,12.5，则a=10,c=‘A’,f=12.5。(**√**)

13.C语言把文件看作是一个字符（字节）的序列。(**√**)

14.若有宏定义：#define S(a,b) t=a;a=b;b=t由于变量t没定义，所以此宏定义是错误的。(**×** )

15.在Turbo C中，下面的定义和语句是合法的：file \*fp;fp=fopen("a.txt","r");(**×** )

16.若有定义：char s[ ]＝"china"；则Turbo C系统为数组s开辟6个字节的内存单元。(**√** )

17.若有定义和语句：int a[3][3]＝{{3,5},{8,9},{12,35}},i,sum=0；for(i=0;i<3;i++) sum+=a[i][2-i];则sum=21。(**√** )

18.若有定义和语句：struct student { int num; char name[10]; float score;} s[5]={{1,"lili",98.5},{9,"xiaohua",66}},\*p=s;printf("%d",\*p++);输出结果是1。(**×**)

19.在程序中定义了一个结构体类型后，可以多次用它来定义具有该类型的变量。(**√**)

20.在Turbo C中，此定义和语句是合法的：enum aa{ a＝5,b,c}bb;bb=(enum aa)5;( **√**)

（ × ）21. C允许对数组的大小作动态定义，即可用变量定义数组的大小。

（ × ）22.字符数组不要求它的最后一个字符为‘\0’，甚至可以不包含‘\0’。

（ × ）23. 数组定义 int a[10]; 也可以写成 int a(10); 。

（ √ ）24.在对全部数组元素赋初值时，可以不指定数组长度。

（ × ）25.定义s为5×6(5行6列)的数组可写成 float a[5,6];

（ √ ）26.数组定义 int a[10]; 表示数组名为a，此数组有10个元素。

（ × ）27.数组定义 int a[10]; 表示数组名为a，此数组有10个元素，第10个元素为a[10]。

（ √ ）28.static char c[]={“a book”}; 与static char c[]=“a book”; 等价。

（ × ）29. static char c[]={'a' ,' ','b','o','o','k','\0'};与 static char c[]={'a' ,' ' ,'b','o','o','k'}; 等价。

（ × ）30.设已有说明：static char c1[10],c2[10]; 下面程序片段是合法的。c1={“book”}; c2=c1;

（ × ）31. 函数说明指的是在程序中设定一个函数模块。

（ √ ）32. 形参只有在被调用时才分配存储空间。

（ × ）33.在C的函数中，最好使用全局变量。

（ × ）34.在调用函数时，实参传值给形参，调用结束时，形参值传给实参。

（ √ ）35.所有函数定义都是并行的，相互独立的。

（ × ）36.函数的隐含存储类型是extern。

（ × ）37.形参可以是常量、变量或表达式。

（ × ）38.函数调用可以作为一个函数的形参。

（ √ ）39. C语言规定，实参应与其对应的形参类型一致。

（ × ）40.定义函数时，形参的类型说明可以放在函数体内。

1. **编程题**

**1.输入2个整数，求两数的平方和并输出。**

#include <stdio.h>

int main(void)

{ intt a ,b,s;

printf("please input a,b:\n");

scanf("%d%d”,&a,&b);

s=a\*a+b\*b;

printf("the result is %d\n",s);

return 0;

}

**2. 输入一个圆半径r，当r>＝0时，计算并输出圆的面积和周长，否则，输出提示信息。**

#include <stdio.h>

#define PI 3.14

int main(void)

{ double r ,area , girth;

printf("please input r:\n");

scanf("%lf",&r);

if (r>=0)

{ area =PI\*r\*r;

girth =2\*PI\*r ;

printf("the area is %.2f\n", area);

printf("the girth is %.2f\n", girth);}

else

printf("Input error!\n");

return 0;

}

**3.已知函数y=f(x)，编程实现输入一个x值，输出y值。**

**2x+1 (x<0)**

**y= 0 (x=0)**

**2x-1 (x>0)**

#include <stdio.h>

void main()

{ int x,y;

scanf(“%d”,&x);

if(x<0) y=2\*x+1;

else if(x>0) y=2\*x-1;

else y=0;

printf(“%d”,y);

}

**4.从键盘上输入一个百分制成绩score，按下列原则输出其等级：score≥90，等级为A；80≤score<90，等级为B；70≤score<80，等级为C；60≤score<70，等级为D；score<60，等级为E。**

#include <stdio.h>

void main(){

int data;

char grade;

printf("Please enter the score:");

scanf("%d”, &data);

switch(data/10)

{ case 10:

case 9 : grade=’A’; break;

case 8: grade=’B’; break;

case 7: grade=’C’; break;

case 6: grade=’D’; break;

default: grade=’E’;

}

printf("the grade is %c”,grade);

}

**5. 编一程序每个月根据每个月上网时间计算上网费用，计算方法如下：**

****

**要求当输入每月上网小时数,显示该月总的上网费用(6分)**

#include <stdio.h>

void main()

{ int hour;

float fee;

printf(“please input hour:\n”);

scanf(“%d”,&hour);

if(hour<=10)

fee=30;

else if(hour>=10&&hour<=50)

fee=3\*hour;

else fee=hour\*2.5;

printf(“The total fee is %f”,fee);

}

**6. 从键盘输入10个整数，统计其中正数、负数和零的个数，并在屏幕上输出。**

#include <stdio.h>

void main( ) {

int a, i,p=0,n=0,z=0;

printf("please input number");

for(i=0;i<10;i++){

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

if (a>0) p++;

else if (a<0) n++;

else z++;

}

printf("正数：%5d, 负数：%5d,零：%5d\n",p,n,z);

}

**7.编程序实现求1-10之间的所有数的乘积并输出。**

#include <stdio.h>

void main( )

{ int i；

long sum=1;

for(i=1; i<=10; i=i+1)

sum=sum\*i;

printf(“the sum of odd is :%ld”,sum);

}

**8. 从键盘上输入10个数，求其平均值。**

#include <stdio.h>

void main(){

int a,i,sum=0;

float ave;;

for(i=0;i<10;i++){

scanf("%d",&a);

sum+=a;

}

ave=(float)sum/10;

printf("ave = %f\n", ave);

}

**9.编程序实现求1-1000之间的所有奇数的和并输出。**

#include <stdio.h>

void main( )

{ int i, sum=0;

for(i=1; i<1000; i=i+2)

sum=sum+i;

printf(“the sum of odd is :%d”,sum);

}

**10.有一个分数序列：2/1，3/2，5/3，8/5，13/8，……编程求这个序列的前20项之和。**

#include <stdio.h>

void main(){

int i,t,n=20;

float a=2,b=1,s=0;

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

{s=s+a/b;

t=a;

a=a+b;

b=t;

}

printf("sum=%6.2f",s);

}

**11．从键盘输入两个数，求出其最大值（要求使用函数完成求最大值，并在主函数中调用该函数）**

#include <stdio.h>

float max(float x,float y);

void main()

{ float a,b,m;

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

m=max(a,b);

printf("Max is %f\n",m);

}

float max(float x,float y)

{

if (x>=y)

return x;

else

return y;

}

**12. 编写程序，其中自定义一函数，用来判断一个整数是否为素数，主函数输入一个数，输出是否为素数。**

#include <math.h>

#include <stdio.h>

int IsPrimeNumber(int number)

{ int i;

if (number <= 1)

return 0;

for (i=2; i<sqrt(number); i++)

{ if ((number % i) == 0)

return 0; }

return 1;}

void main()

{ int n;

printf(“Please input n:”);

scanf(“%d”,&n);

if(IsPrimeNumber(n))

printf(“\n%d is a Prime Number”,n);

else printf(“\n%d is not a Prime Number”,n);}

**13、从键盘输入n个数存放在数组中，将最小值与第一个数交换，输出交换后的n个数。**#include <stdio.h>

int main(void){

int i,n,iIndex,temp;

int a[10];

printf("Enter n: ");

scanf("%d", &n);

printf("Enter %d integers:\n ");

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

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

iIndex=0;

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

if(a[i]<a[iIndex]) iIndex=i;

}

temp=a[0];a[0]=a[iIndex];a[iIndex]=temp;

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

printf("%5d", a[i]);

printf("\n");

return 0;

}

第二种解法 利用函数

#include<stdio.h>

int comp(int arry[], int n)

{

int i,index,temp;

printf("为数组赋值：\n");

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

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

}

for(i=1,index=0;i<=n-1;i++)

{ if(arry[i]<arry[index])

{ index=i;

}

}

temp=arry[0];arry[0]=arry[index];arry[index]=temp;

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

{ printf("%d ",arry[i]);

}

return 0;

}

main()

{ int n;

int a[10];

printf("为n赋值：\n");

scanf("%d",&n);

comp(a,n);}