国内外经典教材辅导系列·计算机类

# 谭浩强《C程序设计》

（第4版）

# 配套题库

## 【考研真题精选＋章节题库】

主编：圣才学习网
www.100xuexi.com

圣才电子书出品

## 内容简介

中国高等院校计算机基础教育课程体系规划教材《C 程序设计》（第 4 版，谭浩强主编，清华大学出版社）是我国高校广泛采用的计算机专业权威教材之一，也被众多高校（包括科研机构）指定为计算机专业考研考博专业课参考书目。

不同一般意义的传统题库，本题库是详解研究生入学考试指定考研参考书目为谭浩强《C 程序设计》（第 4 版）的专业课复习题库，包括考研真题精选、章节题库两大部分。具体来说包括以下两部分：

第一部分为考研真题精选。精选部分名校考研真题以及相关教辅资料的典型习题，并提供详尽答案解析。学员可以熟悉考试真题的特点，并测试自己的水平。

第二部分为章节题库。遵循谭浩强《C 程序设计》（第 4 版）的章目编排，共分为 11 章，同时针对该教材的重难点相应整理了典型题，并对题库中的试题进行详细解析。

圣才电子书编辑部

# 目 录

# 第一部分　考研真题精选

## 一、选择题

1．以下均是合法变量名的是（　　）。[武汉科技大学 2019 研]

A．#name　total

B．node　value_max

C．_var　long

D．stu-code　a+b

【答案】B

【解析】C 语言中变量名只能包含数字，字母和下划线，且只能以字母和下划线开始。A 项含非法字符#，错误；C 中 long 为关键字，变量不能以关键字命名；D 中含非法字符-和+。

2．以下选项中不属于 C 语言类型的是（　　）。[武汉科技大学 2019 研]

A．short int

B．unsigned long int

C．char

D．bool

【答案】D

【解析】C 语言中没有 bool 型，只有 C++才有 boolean 型，也称 bool。C 语言中一般用“0”表示“假”，用“1”表示“真”。

3．若有声明语句：int x; char y[20]; double z; 则正确的输入语句是（　　）。[武汉科技大学 2019 研]

A．scanf("%d%c%le\n",&x,&y,&z);

B．scanf("%2d%s%lf",&x,&y,&z);

C．scanf("%d%s%lf",&x,y,&z);

D．scanf("%x%s%3.2f",&x,y,&z);

【答案】C

【解析】y 为一维数组名，指向数组首元素的地址，因此不需要再使用取地址运算符&，AB 错误；D 中%3.2f 表示长度为 3，小数为 2 位，但是小数点也占一位，因此 D 错误，答案选 C。

4．若 a、b 和 t 都为 int 变量，则下面不能交换变量 a 和 b 值的是（　　）。[武汉科技大学 2019 研]

A．t=a; a=b; b=t;

B．a=t; t=b; b=a;

C．t=b; b=a; a=t;

D．a=a+b; b=a-b; a=a-b;

【答案】B

【解析】B 中首先把 t 的值赋值给了 a，则 a 的值已经被取代了，后面执行 b=a，则 ab 的值都等于 t 的值。

5．若有定义：int a=1,b=2; float x=3,w; 则合法的 switch 语句是（　　）。[武汉科技大学 2019 研]

A．

```
switch(a)
{
   case 1:w=a/b;   break;
   case 2: w=a%b;   break;
}
```

B．

```
switch(b)
{
   case 1:z=a%b;
   case 2: z=a/b;    break;
}
```

C.

```
switch(x)
{
   case 2:w=a%b;   break;
   case 1:w=a/b;   break;
}
```

D.

```
switch(a+b);
{
   case 3:
   case 2:w=a%b;   break;
}
```

【答案】A

【解析】B 中，变量 z 未定义；C 中 x 为浮点型，switch 后面的表达式不能是浮点型，只能是整型和字符型；D 中 swith 表达式后面不能加分号，答案选 A。

6．对下述程序段的描述正确的是（　　）。[武汉科技大学 2019 研]

```
scanf("%d,%d",&a,&b);
if(a>b)
   a=b;b=a;
else
   a++; b++;
printf("a=%d,b=%d",a,b);
```

A．若输入 4,5 则输出 a=5,b=6

B．若输入 5,4 则输出 a=4,b=5

C．若输入 5,4 则输出 a=5,b=5

D．有语法错误，不能通过编译

【答案】D

【解析】if（表达式）后面如果没有用花括号括起来，那么 if 的子语句只包括第一条语句，即在程序中只有 a=b 是属于 if 语句块的，if 和 else 中间隔了一条语句 b=a，编译无法通过。

7．以下正确的描述是（　　）。[武汉科技大学 2019 研]

A．从多层循环嵌套中退出时，只能使用 break 语句

B．在循环体内使用 continue 和 break 语句，作用相同

C．只能在循环体内和 switch 体内使用 break 语句

D．continue 语句的作用是结束整个循环的执行

【答案】C

【解析】从多层嵌套中退出不是只能使用 break 语句，也可以使用 return 或者程序自己执行完，A 错误；在循环体内 continue 代表不执行该次循环中的剩余未执行语句，break 代表直接跳出本层循环，BD 错误，答案选 C。

8．如果有定义：int x=0,s=0; 则下面程序段的执行结果是（　　）。[武汉科技大学 2019 研]

```
while(!x!= 0)
    s+=x++;
printf(“%d”,s);
```

A．1
B．0
C．无限循环
D．控制表达式非法，无法编译
【答案】B
【解析】while 后面的表达式中，首先执行!运算符，然后再执行!=运算符，第一次判断中，x=0 则!x!=0 满足条件，进入循环中，执行 s+=x++，x++是先运算，再自加，执行完后 s=0，x=1，再回到 while 的判断条件，判断为 false，跳出循环，输出 s 的值为 0，答案选 B。

9．下面各语句中，能正确进行字符串操作的语句是（　　）。[武汉科技大学 2019 研]
A．char a[10]={'A','B','C','D','\0'};
B．char a[10];　a="ABCDE";
C．char *p;　*p="ABCDE";
D．char *s; scanf("%s", s);
【答案】A
【解析】B 项中，字符数组的数组名指向数组的首元素地址，初始化后不可再被更改；CD 两项中的字符指针在定义时均没有进行初始化，对其赋值是非法的，答案选 A。

10．以下能对数组 value 进行正确初始化的语句是（　　）。[武汉科技大学 2019 研]
A．int value[2][]={{1,1},{2,2}};
B．int value[][3]={{1,,3},{4,5,6}};
C．int value[2][3]={1,2,3,4,5,6};
D．int value[][3]={{1},{4,6,}};
【答案】C
【解析】二维数组的定义必须指定列数，可以不用指定行数，A 错误；数组 value 为 int 型数组，不能给数组里面的元素赋空值，BD 错误，答案选 C。

11．函数 fun 和实参数组的声明形式为：void fun(char ch,float x[]); float a[5];
以下对函数的调用语句中，正确的是（　　）。[武汉科技大学 2019 研]
A．fun("a",a[]);
B．t=fun('D',a);
C．fun('65',2.8);
D．fun(32,a[5]);
【答案】B
【解析】调用函数 fun 需要传入两个实参，第一个实参的类型是 char 字符型，第二个实参的类型是 float 数组型，只有选项 B 满足条件。其中 C 项'65'本身有语法错误，D 项在传递数组时只需要数组名即可。

12．设有定义 int a[3][3];和函数调用语句 sort(a,3); 则正确的函数声明是（　　）。[武汉科技大学 2019 研]
A．void sort(int a,n);
B．void sort(int a[][],int n);
C．void sort(int a[][3],int n);
D．void sort(int a[][3],n);
【答案】C
【解析】根据函数调用语句可以知道 sort 函数有两个参数，第一个参数的类型是一个二维 int 型数组，第二个参数是一个 int 型数据，所以 A 错误；B 中因为二维数组一定要指定列数，B 错误；D 中形参 n 没有指定数据类型，错误；答案选 C。

13．有函数定义：int func(int *p)，x 和 y 是 int 型变量，则正确的调用是（　　）。[武汉科技大学 2019 研]

A．y=func(x);

B．func(x);

C．func()=x;

D．y=func(&x);

【答案】D

【解析】根据 func 函数的定义可以知道调用 func 函数需要传入一个指针，且该指针的指向类型是 int 型，只有 D 传入的是指向 int 型数据的指针，答案选 D。

14．已知书籍结构定义如下，则对结构变量 bk 的正确赋值是（　　）。[武汉科技大学 2019 研]

```
struct BOOK
{
   struct
   {
      int year,month,day;
   }publish;
}bk;
```

A．bk.year=1998;　bk.month=11;　bk.day=11;

B．publish.year=1998;　publish.month=11;　publish.day=11;

C．year=1998;　month=11;　day=11;

D．bk.publish.year=1998;　bk.publish.month=11;　bk.publish.day=11;

【答案】D

【解析】变量 bk 是结构体 BOOK 的一个结构体变量，该变量含有一个成员变量 publish，publish 也是一个结构体变量，该结构变量含三个成员变量，分别是 year、month、day，结构体变量中的成员变量不可直接访问，必须以结构体变量名.成员变量名形式访问，所以只能通过 bk.publish.year 形式访问到最内层的变量并为其赋值，答案选 D。

15．对于以下定义，能打印出字母 h 的语句是（　　）。[武汉科技大学 2019 研]

```
struct person{ char title[20];　int code;};
struct person book[5]={"Physics",17,"Math",18,"English",20,"History",18};
```

A．printf("%c",book[0].title[1]);

B．printf("%c",book[1].title[4]);

C．printf("%c",book[2].title[7]);

D．printf("%c",book[3].title[6]);

【答案】A

【解析】person 是一个自定义结构体类型，该结构体含有两个成员变量，分别是一个字符数组和一个 int 数据，BC 选项打印出来的是'\0'；D 选项打印出来的是 y，只有 A 打印出来的是 h，答案选 A。

16．与十进制 1100 等值的十六进制数是（　　）。[华南理工大学 2018 研]

A．44A

B．44C

C．54A

D．54C

【答案】B

【解析】1100 转换成二进制为 0100 0100 1100，因此转换为十六进制为 44C。

17．设 int a=3;，下列哪一个表达式的值等于 0（　　）。[华南理工大学 2018 研]

A．a&&(a>0)

B. !a||a
C. a%=a
D. a>=a
【答案】C
【解析】A 中 a!=0 且 a>0 所以表达式的值为 1；B 中||表示或，所以值也为 1；D 中表达式值也为 1；答案选 C。

18. 在 C 语言中，当函数的返回值缺省时，表示该函数返回值的类型是（　　）。[华南理工大学 2018 研]
A. char
B. float
C. long
D. int
【答案】D
【解析】在 C 语言中，当函数的返回值缺省时，函数返回值的类型默认为 int 型。

19. 十六进制形式输出整数的格式说明符是（　　）。[华南理工大学 2018 研]
A. %u
B. %ld
C. %x
D. %o
【答案】C
【解析】A 表示输出的是无符号整型；B 表示输出的是有符号长整型；D 表示输出的是八进制。

20. 若有说明：int *p,m=5,n;，以下正确的程序段是（　　）。[华南理工大学 2018 研]
A.

```
p=&n;
  scanf("%d",n);
```

B.

```
p=&n;
  scanf("%d",*p);
```

C.

```
scanf("%d",&n);
  p=n
```

D.

```
p=&n;
  *p=n;
```

【答案】D
【解析】scanf 语句中第二个参数应该是变量的地址，AB 错误；C 中 p 为指针变量，不可以直接把一个 int 型变量赋值给指针型，C 错误；答案选 D。

21. 两次运行下面的程序，如果从键盘上分别输入 6 和 4，则输出的结果是（　　）。[华南理工大学 2018 研]

```
void main(void)
{
   int x;
   scanf(“%d”,&x);
   if(x++>5) printf(“%d”,x);
   else printf(“%d\n”,x--);
}
```

A. 7 和 5
B. 6 和 3
C. 7 和 4
D. 6 和 4

【答案】A

【解析】当输入 6 时，判断 x++>5 为真，进入 if 语句块，此时 x=7，输出 7；当输入 4 时，进入 else 语句块，此时 x=5，然后因为--是先运算后自减，所以先输出 5，后 x 的值为 4，答案选 A。

22．以下叙述中不正确的是（　　）。[华南理工大学 2018 研]

A. 在不同的函数中可以使用相同名字的变量
B. 函数中的形式参数是局部变量
C. 在一个函数内定义的变量只能在本函数范围内有效
D. 在一个函数的复合语句中定义的变量在本函数范围内有效

【答案】D

【解析】在一个函数的复合语句中定义的变量只在该复合语句中有效。

23．设有下列程序

```
ff( )
{
   int c=9;
   static int a=1, b=4;
   if (b == 4)
      {a+=c; b++;}
   else
      {a+= c; b-- ;}
   printf (“a=%d,b=%d\n”, a, b);
}
main ( )
{ ff( ); ff( ); }
```

则该程序执行后，显示的结果为（　　）。[华南理工大学 2018 研]

A. a=10,b=5
a=19,b=5
B. a=10,b=4
a=19,b=5
C. a=10,b=4
a=19,b=4
D. a=10,b=5
a=19,b=4

【答案】D

【解析】第一次调用 ff()时，c=9，a=1，b=4，程序进入 if 语句块，执行完后 a=10，b=5；第二次调用 ff()，因为 a、b 是静态变量，所以 a、b 的值不会重新初始化，所以进入 else 语句块，执行完后 a=19，b=4，答案选 D。

24．以下数组定义中不正确的是（　　）。[华南理工大学 2018 研]

A．int a[2][3];

B．int b[][3] = {0};

C．int c[100][100] = {0};

D．int d[3][] = {{1}, {1, 2, 3},{1}};

【答案】D

【解析】定义二维数组时一定要指定数组的列数，可以不指定数组的行数，D 错误。

25．若有定义：int *p,*s,c;，且各变量已正确赋值，则非法的赋值表达式是（　　）。[华南理工大学 2018 研]

A．p = s

B．c = *s

C．*s = &p

D．p = &c

【答案】C

【解析】C 中 p 为指针变量，则&p 表示的是指针的地址，若要赋值，则左边变量应该是一个二级指针，而*s 代表的是 s 所指向地址的变量值，这个变量是一个 int 型，显然不正确。

26．若有以下说明：

int w[3][4]={{0,1},{2,4},{5,8}};

int (*p)[4]=w;

则数值为 4 的表达式是（　　）。[华南理工大学 2018 研]

A．*w[1]+1

B．p++,*(p+1)

C．w[2][2]

D．p[1][1]

【答案】D

【解析】A 中*w[1]表示的的是数值 2，则表达式的值为 3，错误；B 中 p+1 表示指向二位数组第二行的首元素地址，而*(p+1)代表的是第三行首元素值，B 错误；C 中表示的是数值 0，答案选 D。

27．下列函数的功能是（　　）。[华南理工大学 2018 研]

```
int fun1(char *x)
{
   char *y=x;
   while(* y++);
   return (y-x-1);
}
```

A．求字符串的长度

B．比较两个字符串的大小

C．将字符串 X 复制到字符串 Y

D．将字符串 X 连接到字符串 Y 后

【答案】A

【解析】while 后面的表达式是指针依次遍历直到碰到\0，此时 y 指向字符串最后一个元素的后一个位置，但是由于 y++，因此 y 会继续后移一位，而 x 指向字符串的头部，后面的 y-x-1 显然是用于计算字符串的长度。

28．feof 函数用来判断文件是否结束，如果文件没有结束，则返回值是（　　）。[华南理工大学 2018 研]

A．-1

B．0

C．1

D．EOF

【答案】B

【解析】此题考查 feof 函数的定义与作用。

29．设有说明：FILE *fp; char *filename ="paper";

对于 fp=fopen(filename,"rb+");语句，下面说法正确的是（　　）。[华南理工大学 2018 研]

A．打开名为 filename 的文件读写

B．打开名为 paper 的文件读写

C．打开名为 filename 的文件只读

D．打开名为 paper 的文件只读

【答案】B

【解析】filename 只是一个字符指针变量的名字，不是文件的名字 AC 错误；rb+表示打开二进制文件，且该文件可以读写，答案选 B。

30．若变量 a，b，c 的取值分别是 1，2，3，则表达式“!((b+c)>(a+4))”的值是（　　）。[北京航空航天大学 2018 研]

A．0

B．1

C．2

D．3

【答案】B

【解析】首先 b+c 等于 5，a+4 也等于 5，因此(b+c)>(a+4)为“假”，即 0，对 0 取非结果为 1，因此答案为 B。

31．以下关于循环语句的叙述中，正确的是（　　）。[北京航空航天大学 2018 研]

A．for 循环语句的三个部分必须都要有表达式

B．while 循环语句的循环体内至少要有一条语句

C．do…while 循环语句的循环体至少会被执行一次

D．continue 语句可以退出包含它的整个循环体

【答案】C

【解析】for 循环的三个表达式都可以省略，但是之间的分号不能省略，同时要有退出循环的机制，因此 A 项错误；while 循环语句的循环体内可以为空，并不违反相应语法，只不过循环什么也不执行，因此 B 项错误；continue 语句只是不执行本次循环的剩余语句，而并非退出整个循环，因此 D 项错误，答案选 C。

32．对于下列代码：

```
switch(option)
{
   case 'H':printf("Hello  ");
   case 'W':printf("Welcome ");
   case 'B':printf("Bye");
   break;
}
```

若 option 的取值为'W'，则该代码段的输出结果是（　　）。[北京航空航天大学 2018 研]

A．Welcome

B．Welcome Bye

C．Hello Welcome Bye

D．以上结果都不对

【答案】B

【解析】由于 option 为'W'，所以首先应该输出 Welcome，但是由于该语句后面没有 break 语句来终止选择语句 switch，因此会继续执行下面的语句，直到遇上 break，所以最后输出 Welcome Bye，答案为 B。

33．若有以下变量的声明语句：

int i=1,a[]={0,2,4},*b;

b=&i;

则下列选项中，其结果与表达式“*(a+1)”相等的是（　　）。[北京航空航天大学 2018 研]

A．a[0]

B．*a+i

C．*(a+b)

D．*(a+*b)

【答案】D

【解析】a 指向数组的首元素，因此*(a+1)表示取数组第二个元素的值，为 2。A 项，a[0]=0，不相等；B 项，*a 为数组第一个元素的值为 0，再加上 i=1，因此结果为 1，不相等；C 项，a 和 b 都是指针，相加没有意义，错误；D 项，*b 的值 i 的值，即 1，*(a+1)表示取数组第二个元素的值为 2，相等，因此答案选 D。

34．若已知有如下宏定义

#define　CANBERRA(x,y)　((x-y)/(x+y))

则以下表达式中，返回结果值最大的是（　　）。[北京航空航天大学 2018 研]

A．CANBERRA(3.0,2.0);

B．CANBERRA(4.0,1.0);

C．CANBERRA(1.0+2.0,0.0+2.0);

D．CANBERRA(1.0+2.0,1.0+1.0);

【答案】C

【解析】A 项中为 1.0/5.0，结果为 0.2；B 项中为 3.0/5.0，结果为 0.6；C 项中的宏替换后为(1.0+2.0-0.0+2.0)/(1.0+2.0+0+2.0)=1.0；D 项中宏替换后为(1.0+2.0-1.0+1.0)/(1.0+2.0+1.0+1.0)=0.6，因此最后答案为 C。

35．对于以下 C 程序，其正确的是（　　）。[北京航空航天大学 2018 研]

```
#include<stdio.h>
int main()
{
   char str1[]="Hello";
   char str2[]="Hello";
   if(str1==str2)
      printf("Equal\n");
   else
      printf("Unequal\n");
   return 0;
}
```

A．Unequal

B．Equal

C．该程序无法通过编译

D．该程序运行时出错

【答案】A

【解析】首先该程序符合语法规则，因此不会编译时产生错误，其次字符数组 str1 和 str2 都为指针常量，将他们直接用关系运算符进行比较肯定是不相等的，但是它们所指的字符串是相等的，因此最后输出 Unequal。

36．已知有以下 sample.c 程序的定义：

```
/*sample.c*/
#include<stdio.h>
int main(int argc,char *argv[])
{
   printf("%c",*++argv[2]);
   return 0;
}
```

将该程序编译成可执行文件 sample 后，若在命令行下输入如下命令：

sample January February March

则该命令正确的输出是（　　）。[北京航空航天大学 2018 研]

A. J

B. a

C. F

D. e

【答案】D

【解析】根据命令行的输入可知，四个字符串赋给相应的指针数组，所以 argv[2]表示第三个字符串 February，再进行++操作，指针指向字符串的第二个字符 e，最后进行取值操作，结果为 e，答案为 D。

37. 以下程序运行后的输出结果是（　　）。[中央财经大学 2018 研]

```
int main()
{
   double d;
   float f;
   long l;
   int i;
   i=f=l=d=20/3;
   printf("%d %ld %.1f %.1f\n",i,l,f,d);
   return 0;
}
```

A. 6 6 6.0 6.0

B. 6 6 6.7 6.7

C. 6 6 6.0 6.7

D. 6 6 6.7 6.0

【答案】C

【解析】赋值运算符是自右向左结合的，所以首先执行 d=20/3=6，同时 i、l、f 也全为 6，在进行输出时，f 和 d 要保留一位小数，所以答案选 C。

38. 若有定义和语句：char s[10];s="abcd";printf("%s\n",s);，则结果是（　　）。[中央财经大学 2018 研]

A. 输出 abcd@#$

B. 输出 a

C. 输出 abcd

D. 编译不通过

【答案】D

【解析】在定义一维字符数组时，s 为数组名，指向数组首元素的地址，为地址常量，不可更改，因此语句 s="abcd"是非法的，编译不会通过。

39. 以下叙述错误的是（　　）。[中央财经大学 2018 研]

A. 在程序中凡是以“#”开始的语句行都是预处理命令行

B. 预处理命令行的最后不能以分号表示结束

C. #include MAX 是合法的宏定义命令行

D. C 程序对预处理命令行的处理是在程序执行的过程中进行的

【答案】D

【解析】在 C 语言中，凡是以“#”开头的行都称为编译预处理命令行，为了区别 C 语句，后面是不加分号的。编译预处理是在编译程序对 C 源程序进行编译前执行的，而不是在程序执行过程中进行的。

40. 若有如下宏定义：

```
#define   N   2
#define   y(n)   ((N+1)*n)
```

则执行下列语句：z=4*(N+y(5));后的结果是（　　）。[中央财经大学 2018 研]

A. 语句有错误

B. z 值为 68

C. z 值为 60

D. z 值为 180

【答案】B

【解析】y(5)=15，z=4*(N+y(5))=4*17=68，答案选 B。

41. 有如下说明

```
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（　　）。[中央财经大学 2018 研]

A. *p+9

B. *(p+8)

C. *p+=9

D. p+8

【答案】B

【解析】A 中*p=1，*p+9=10，A 错误。C 中*p 得到的是一个常量，常量不可以被赋值，C 错误。D 中 p 是地址，p+8 仍然表示一个地址。因此 B 项正确，p+8 指向元素 9，进行取值得 9。

42. 若有以下说明和语句：

```
struct worker
{
   int no;
   char *name;
}work,*p=&work;
```

则以下引用方式不正确的是（　　）。[中央财经大学 2018 研]

A. work.no

B. (*p).no

C. p->no

D. work->no

【答案】D

【解析】结构体变量访问成员变量的引用方式采用“.”，而结构体指针采用“->”，因此 AC 是正确的，B 项中*p 表示结构体变量，因此可以用“.”，所以答案选择 D。

43. 以下不合法的用户标识符是（　　）。[四川大学 2017 研]

A. J2_KEY

B. Double

C. 4d

D. _8_

【答案】C

【解析】标识符只能包含数字，字母，下划线，且不能以数字开头，选项 C 错误。

44．算法具有五个特性，以下选项中不属于算法特征性的是（　　）。[四川大学 2017 研]

A．有穷性

B．简洁性

C．可行性

D．确定性

【答案】B

【解析】一个算法应该具有五个重要特征：①有穷性；②确切性；③输入性；④输出性；⑤可行性。因此简洁性并不属于算法特征。

45．已定义如下函数：

```
fun(int *p)
{return *p;}
```

该函数的返回值是（　　）。[四川大学 2017 研]

A．不确定的值

B．形参 p 中存放的值

C．形参 p 所指存储单元中的值

D．形参 p 的地址值

【答案】C

【解析】p 是一个指向 int 型的指针变量，*p 表示的是 p 所指向内存存放的变量，是一个 int 型，所以 return *p 表示返回 p 所指存储单元中的值，答案选 C。

46．在 C 语言中，形参的缺省存储类型是（　　）。[四川大学 2017 研]

A．auto

B．register

C．static

D．extern

【答案】A

【解析】形参是局部变量，缺省类型为 auto 型。

47．语句：printf("%d",(a=2)&&(b=-2));的输出结果是（　　）。[四川大学 2017 研]

A．无输出

B．结果不确定

C．-1

D．1

【答案】D

【解析】a=2 为真，b=-2 也为真，所以输出 1，答案选 D。

## 二、综合应用题

1．对于下列 for 循环语句，请将其改写为功能完全相同的 while 循环语句。[北京航空航天大学 2018 研]

```
int i,j,count=0;
for(i=0;i<100;i++)
{
   for(j=100;j>=i;j-=2)
   {
      count+=j-i;
   }
}
```

【答案】此题可以在定义时先令 i=0，j=100，在第一层 while 循环时只需判断 i<100 即可，在第二层 while 循环时只需判断 j>=i 即可，j-=2 放在内层循环体中即可，具体程序如下：

```
int i=0, j=100, count=0;
while(i<100)
{
   while(j>=i)
   {
      count+=j-i;
      j-=2;
   }
   i++;
}
```

2．输入一个字符，判别它是否为大写字母，如果是，将它变为小写字母；如果不是，不转换。然后输出最后得到的字符。请在下面空白处填上适当语句。[华南理工大学 2018 研]

```
#include<stdio.h>
int main()
{
   char ch;
   scanf("%c",____①____);
   ch=(ch>='A'&&____②____)?(ch+32):____③____;
   printf(____④____,ch);
   return ____⑤____;
}
```

【答案】

①&ch

②ch<='Z'

③ch

④"%c"

⑤0

【解析】程序进行输入时要加上地址符&；main 函数中的第三行为一个三目运算符，当 ch 在 A～Z 之间时 ch 为真，此时 ch+32 变为小写字母，否则不变；程序进行输出时应用%加上数据类型进行输出，最后用 return 0 来结束程序。

3．有一个分数序列

$$2/1,3/2,5/3,8/5,13/8,21/13,\cdots$$

求这个数列的前 20 项之和。请在下面空白处填上适当语句。[华南理工大学 2018 研]

```
int main()
{
   int i,n=20;
   double a=2,b=1,s=0,t;
   for(i=1;      ①      ;i++)
   {
      s=      ②      ;
      ____③____;
      a=a+b;
      ____④____;
   }
   printf("sum=%16.10f\n",      ⑤      );
   return 0;
}
```

【答案】

①i<=n

②s+a/b

③t=a

④b=t

⑤s

【解析】此程序循环是从=1 开始，所以要计算数列前 20 项，则循环条件应为 i<=n；s 用来累加求和，所以每次进行累加操作，即 s=s+a/b；通过分析数列可知，数列中分子是上一项的分子分母之和，而分母是上一项的分子，依照此关系可以通过中间变量 t 进行换算，最后输出所求结果 s。

4. 从键盘输入 10 个整数，编程求其中大于 3 且小于 100 的数的平均值并输出结果。[华南理工大学 2018 研]

【答案】此题可以先利用循环从屏幕读取 10 个数，存放在一个一位数组中，然后循环判断大于 3 且小于 100 的数，最后取平均值进行输出，程序如下：

```
#include<stdio.h>
void    main()
{
   int num[10],s=0,n=10;
   printf("input 10 numbers:");
   for(int i=0;i<10;i++)
   {
   scanf("%d",&num[i]);
      if(num[i]<=3||num[i]>=100)
      {
         num[i]=0;
         n--;
      }
      s+=num[i];
   }
   if(n==0)
      printf("no exist!");
   else
      printf("%d\n",s/n);
}
```

5．编程求 100～200 间的全部素数。[华南理工大学 2018 研]

【答案】素数意思是只能被 1 和本身整除，因此将 1 到本身之间的数做除数，进行求余，如果余数为 0，则不是素数，否则是素数。根据经验，假设所要判断的数为 n，则一般只需要判断 1 到根号 n 之间的数即可，具体程序如下：

```
#include<stdio.h>
#include<math.h>
void main()
{ int i,j,count;
   for(i=100;i<=200;i++)
   {
      count=0;
      for(j=2;j<sqrt(i);j++)
      {
         if(i%j==0)
         count++;
      }
      if(count==0)
      printf("%d 是素数!\n",i);
   }
}
```

6．有 4 个圆塔，圆心分别为（2，2）、（－2，2）、（－2，－2）、（2，－2），圆半径为 1，如图 4-8 所示。这 4 个塔的高度为 10m，塔以外无建筑物。今输入任一点的坐标，求该点的建筑高度（塔外的高度为零）。[四川大学 2017 研]

【答案】N-S 图如图 1-1 所示。

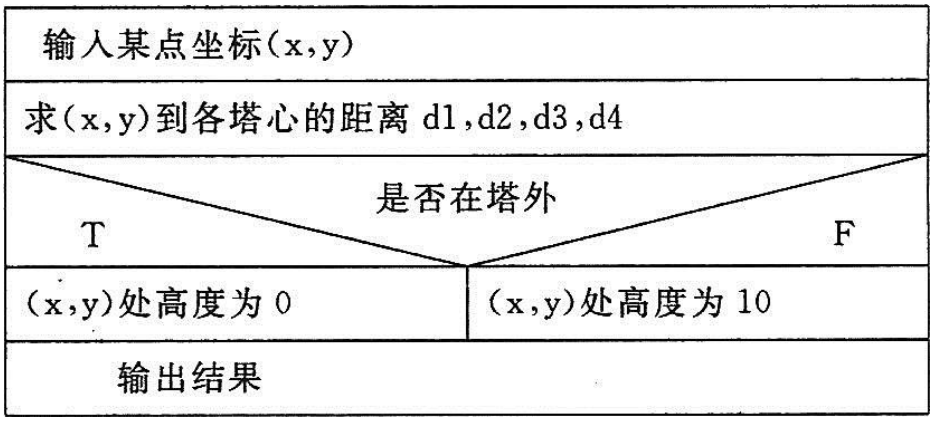



图 1-1　计算某点建筑高度的 N-S 流程图

程序如下：

```
#include<stdio.h>
int main()
{
   int h=10;
   float x1=2,y1=2,x2=-2,y2=-2,x3=-2,y3=-2,x4=2,y4=-2,x,y,d1,d2,d3,d4;
   printf("请输入一个点(x,y)： ");
   scanf("%f,%f",&x,&y);
   d1=(x-x4)*(x-x4)+(y-y4)*(y-y4);          //求该点到各中心点距离
   d2=(x-x1)*(x-x1)+(y-y1)*(y-y1);
   d3=(x-x2)*(x-x2)+(y-y2)*(y-y2);
   d4=(x-x3)*(x-x3)+(y-y3)*(y-y3);
   if(d1>1&&d2>1&&d3>1&&d4>1)   h=0;       //判断该点是否在塔外
   printf("该点高度为%d\n",h);
   return 0;
}
```

7．从键盘输入一个字符串，将其中的小写字母全部转换成大写字母，然后输出到一个磁盘文件“test”中保存。输入的字符串以“!”结束。[四川大学 2017 研]

【答案】本题先利用循环逐个读入所输入的字符，同时判断所输入字符是否为“!”，若是则停止输入，结束程序；若不是，则判断该字符是否是小写字母，若是则转换为大写字母，最后写入文件 text 中，具体程序如下：

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
   FILE *fp;
   char ch;
   fp=fopen("text.txt","w+");
   if(fp == NULL)
   {
      printf("Cannot open file strike any key exit!");
      exit(1);
   }
   printf("input the string:");
   while(1)
   {
   scanf("%c",&ch);
      if(ch=='!') break;
   if(ch>='a'&&ch<='z')
      ch-=32;
      fprintf(fp,"%c",ch);
   }
   fclose(fp);
}
```

# 第二部分　章节题库

## 第 1 章　程序设计和 C 语言

一、选择题

1．C 语言编译程序的功能是（　　）。

A．执行一个 C 语言编写的源程序

B．把 C 源程序翻译成 ASCII 码

C．把 C 源程序翻译成机器代码

D．把 C 源程序与系统提供的库函数组合成一个二进制执行文件

【答案】C

【解析】编译程序的功能是将“高级语言”翻译为“机器语言”。每条 C 语言语句，经过编译最终都将转换成二进制的机器指令。答案选择 C 选项。

2．计算机高级语言程序的运行方法有编译执行和解释执行两种，以下叙述中正确的是（　　）。

A．C 语言程序仅可以编译执行

B．C 语言程序仅可以解释执行

C．C 语言程序既可以编译执行，又可以解释执行

D．以上说法都不对

【答案】A

【解析】编译执行是指程序执行前需要一个专门的编译过程把程序编译成机器语言的文件，再次运行时不需要重新翻译，执行效率高；解释执行是指每个语句都是执行的时候才翻译，执行效率低。用 C 语言编写的程序必须经过编译器编译后，转换为二进制的机器指令来运行。答案选择 A 选项。

3．以下叙述中错误的是（　　）。

A．C 语言中的每条可执行语句和非执行语句最终都将被转换成二进制的机器指令

B．C 程序经过编译、链接步骤之后才能形成一个真正可执行的二进制机器指令文件

C．用 C 语言编写的程序称为源程序，它以 ASCII 代码形式存放在一个文本文件中

D．C 语言源程序经编译后生成后缀为.obj 的目标程序

【答案】A

【解析】A 项错误，注释语句不会被翻译成二进制的机器指令。C 源程序经过 C 编译程序编译之后生成后缀为.obj 的二进制文件（称为目标文件），然后由“链接程序”（Link）的软件把.obj 文件与各种库函数连接起来生成一个后缀为.exe 的可执行文件。答案选择 A 选项。

4．以下叙述中错误的是（　　）。

A．C 语言的可执行程序是由一系列机器指令构成的

B．用 C 语言编写的源程序不能直接在计算机上运行

C．通过编译得到的二进制目标程序需要连接才可以运行

D．在没有安装 C 语言集成开发环境的机器上不能运行 C 源程序生成的 exe 文件

【答案】D

【解析】A 项正确，C 语言的可执行程序是由一系列机器指令组成的；BC 项正确，用 C 语言编写的源程序必须经过编译，生成二进制目标代码，再经过连接才能运行；D 项错误，C 语言经过编译链接后的二进制目标代码可以脱离 C 语言集成开发环境独立运行。答案选择 D 选项。

5．以下叙述正确的是（　　）。

A．C 编译程序把文件后缀为.c 的源程序文件编译成文件后缀为.obj 的二进制文件

B．C 编译程序把文件后缀为.c 的源程序文件编译成文件后缀为.exe 的可执行文件

C．C 编译程序把文件后缀为.obj 的二进制文件编译成文件后缀为.exe 的可执行文件

D．链接程序把文件后缀为.c 的源程序文件链接成文件后缀为.exe 的可执行文件

【答案】A

【解析】C 编译程序把文件后缀为 C 的源程序文件编译成文件后缀为.obj 的二进制文件，链接将一个或多个目标文件与程序用到的库文件连接起来，形成一个可以在操作系统直接运行的执行程序.exe，故排除 B、C、D 项，答案选择 A 选项。

6．计算机能直接执行的程序是（　　）。

A．源程序

B．目标程序

C．汇编程序

D．可执行程序

【答案】D

【解析】C 语言是一种高级语言，C 语言源程序经过 C 语言编译程序编译之后，生成一个后缀为.obj 的二进制文件（称为目标程序），最后还要由“连接程序”（Link）软件，把此.obj 文件与 C 语言提供的各种库函数连接在一起，生成后缀.exe 的可执行程序。汇编程序是由汇编语言写成的程序，计算机不能直接执行。计算机能直接执行的程序是经过编译器处理转换为机器语言的程序。答案选择 D 选项。

7．针对简单程序设计，以下叙述的实施步骤正确的是（　　）。

A．确定算法和数据结构、编码、调试、整理文档

B．编码、确定算法和数据结构、调试、整理文档

C．整理文档、确定算法和数据结构、编码、调试

D．确定算法和数据结构、调试、编码、整理文档

【答案】A

【解析】简单程序设计的步骤是首先要确定算法和数据结构，然后编码、调试，最后整理相关文档。答案选择 A 选项。

8．以下叙述中正确的是（　　）。

A．程序设计的任务就是编写程序代码并上机调试

B．程序设计的任务就是确定所用数据结构

C．程序设计的任务就是确定所用算法

D．以上三种说法都不完整

【答案】D

【解析】程序设计是指设计、编程、调试程序的方法和过程，通常分为 4 个阶段：①问题建模；②算法设计；③编写代码；④编译调试。其工作内容涉及有关的基本概念、工具、方法及方法学，是目标明确的智力活动。答案选择 D 选项。

9．以下叙述中错误的是（　　）。

A．C 程序在运行过程中所有计算都以十进制方式进行

B．C 程序在运行过程中所有计算都以二进制方式进行

C．所有 C 程序都需要编译链接无误后才能运行

D．C 程序中字符变量存放的是字符的 ASCII 值

【答案】A

【解析】C 程序在运行过程中所有计算都以二进制方式进行。答案选择 A 选项。

10．C 语言源程序名的后缀是（　　）。

A．c

B．exe

C．obj

D．cp

【答案】A

【解析】由 C 语言构成的指令序列称为 C 源程序，源程序文件的后缀为“.c”。源程序经过 C 编译程序编译生成后缀为“.obj”的二进制文件（称为目标文件），然后由称为“连接程序”（Link）的软件把目标文件与 C 语

言提供的各种库函数连接起来，生成后缀为“.exe”的可执行文件。答案选择 A 选项。

11．关于程序设计基本概念，以下叙述错误的是（　　）。

A．计算机可以直接执行由任意高级语言编写的程序

B．高级语言都有与之对应的编译程序或解释程序

C．用任何一种计算机高级语言都可以把算法转换为程序

D．结构化算法可以解决任何复杂的问题

【答案】A

【解析】A 项计算机只能识别机器语言，不能直接识别由高级语言编写的程序。结构化的程序是由三种基本的结构组成的：顺序结构、选择结构和循环结构，使用这三种结构能够解决任何问题。答案选择 A 选项。

12．以下关于算法的叙述中错误的是（　　）。

A．算法可以用伪代码、流程图等多种形式来描述

B．一个正确的算法必须有输入

C．一个正确的算法必须有输出

D．用流程图可以描述的算法可以用任何一种计算机高级语言编写成程序代码

【答案】B

【解析】算法可以使用自然语言、伪代码、流程图等多种不同的方法来描述。故选项 A、D 说法正确。一个正确的算法可以有零个或者多个输入，必须有一个或者多个输出。故选项 C 说法正确，B 说法错误。答案选择 B 选项。

13．以下叙述中错误的是（　　）。

A．算法正确的程序最终一定会结束

B．算法正确的程序可以有零个输出

C．算法正确的程序可以有零个输入

D．算法正确的程序对于相同的输入一定有相同的结果

【答案】B

【解析】算法的 5 个特性：①有穷性；②确定性；③可行性；④有零个或多个输入；⑤有一个或多个输出。答案选择 B 选项。

14．以下选项中叙述正确的是（　　）。

A．C 程序中的语句要经过编译才能转换成二进制机器指令

B．算法需要包含所有三种基本结构

C．有些算法不能用三种基本结构来表达

D．简单算法的操作步骤不能超过 20 步

【答案】A

【解析】编译就是把高级语言变成计算机可以识别的二进制语言，不经过编译的源程序是不能运行的，A 项正确。算法不一定要包含所有三种基本结构，B 项错误。结构化程序主要由 3 种基本控制结构组成，循环结构、选择结构、顺序结构，它们组成的算法结构可以解决任何复杂的问题，C 项错误。算法的复杂程度不是由操作步骤多少决定的，而是由时间复杂度与空间复杂度来衡量，D 项错误。答案选择 A 选项。

15．流程图是描述算法的很好的工具，一般的流程图中由几种基本图形组成。其中判断框的图形是（　　）。

A．菱形

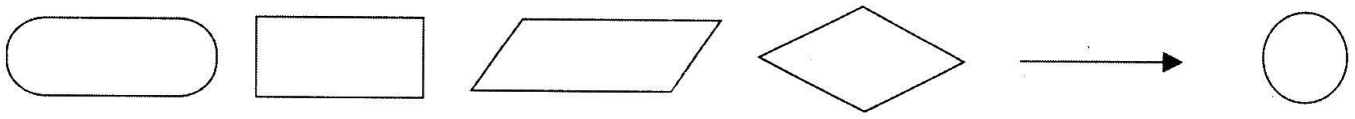
B．长方形

C．平行四边形

D．椭圆型

【答案】A

【解析】传统的流程图由下图所示的几种基本图形组成。




答案选择 A 选项。

16．构成 C 程序的三种基本结构是（　　）。

A．顺序结构、转移结构、递归结构

B．顺序结构、嵌套结构、递归结构

C．顺序结构、选择结构、循环结构

D．选择结构、循环结构、嵌套结构

【答案】C

【解析】结构化程序由三种基本结构组成：顺序结构、选择结构和循环结构。已经证明，由三种基本结构组成的算法可以解决任何复杂的问题。答案选择 C 选项。

17．C 语言主要是借助以下（　　）功能来实现程序模块化的。

A．定义函数

B．定义常量和外部变量

C．三种基本结构语句

D．丰富的数据类型

【答案】A

【解析】C 程序的模块化主要通过函数来实现。C 语言允许对函数单独进行编译，从而可以实现模块化。

18．以下选项中叙述正确的是（　　）。

A．结构化程序的三种基本结构是循环结构、选择结构、顺序结构

B．C 语言源程序不编译也能直接运行

C．使用 N-S 流程图不能描述复杂算法

D．计算机能够直接运行 C 语言源程序，不必进行任何转换

【答案】A

【解析】编译就是把高级语言变成计算机可以识别的二进制语言，不经过编译的源程序是不能运行的，B 项错误。算法可以用各种描述方法进行描述，N-S 流程图把算法的每一步都用一个矩形框来表示，把一个个矩形框按执行的次序连接起来就是一个算法描述，无论算法复杂与否都能用 N-S 流程图描述，C 项错误。C 语言源程序需要经过编译和连接生成目标文件和可执行文件后才能运行，D 项错误。答案选择 A 选项。

19．以下选项中关于程序模块化的叙述错误的是（　　）。

A．把程序分成若干相对独立的模块，可便于编码和调试

B．把程序分成若干相对独立、功能单一的模块，可便于重复使用这些模块

C．可采用自底向上、逐步细化的设计方法把若干独立模块组装成所要求的程序

D．可采用自顶向下、逐步细化的设计方法把若干独立模块组装成所要求的程序

【答案】C

【解析】把一个大程序分解成若干相对独立的子程序，大大提高了程序编制的效率。软件编制人员在进行程序设计时，首先应当集中考虑主程序中的算法，写出主程序后再动手逐步完成子程序的调用。对于这些子程序也可用调试主程序的同样方法逐步完成其下一层，就是自顶向下、逐步细化、模块化的程序设计方法。答案选择 C 选项。

20．以下关于结构化程序设计的叙述中正确的是（　　）。

A．一个结构化程序必须同时由顺序、分支、循环三种结构组成

B．结构化程序使用 goto 语句会很便捷

C．在 C 语言中，程序的模块化是利用函数实现的

D．由三种基本结构构成的程序只能解决小规模的问题

【答案】C

【解析】A 项，一个结构化程序可以由顺序、分支、循环三种结构组成，但不是必须同时都包括，可以包括其中的一个或多个；B 项，goto 语句会破坏程序的结构性、可读性，应尽量不用；D 项，三种基本结构构成的程序也可以解决大规模的问题；C 项，在 C 语言中，利用函数来实现程序的模块化。答案选择 C 选项。

21．以下叙述中错误的是（　　）。

A．使用三种基本结构构成的程序只能解决简单问题

B．结构化程序由顺序、分支、循环三种基本结构组成

C．C 语言是一种结构化程序设计语言

D．结构化程序设计提倡模块化的设计方法

【答案】A

【解析】结构化程序由顺序、分支和循环三种基本结构组成，选项 B 正确。由三种基本结构组成的算法可以解决任何复杂的问题，而不只是解决简单问题，选项 A 错误。由三种基本结构所构成的算法称为结构化算法；由三种基本结构所构成的程序称为结构化程序，C 语言是一种结构化程序设计语言。结构化程序通过函数实现模块化的设计方法。选项 C、D 正确。答案选择 A 选项。

22．以下叙述中正确的是（　　）。

A．在算法设计时，可以把复杂任务分解成一些简单的子任务

B．在 C 语言程序设计中，所有函数必须保存在一个源文件中

C．只要包含了三种基本结构的算法就是结构化程序

D．结构化程序必须包含所有的三种基本结构，缺一不可

【答案】A

【解析】A 项正确，把复杂任务分解成一些简单的子任务是程序模块化的思想；B 项错误，C 语言中，函数可以放在不同的源文件中；CD 两项错误，由三种基本结构所构成的程序称为结构化程序，三种基本结构可以任意组合。答案选择 A 选项。

23．以下选项中叙述正确的是（　　）。

A．使用三种基本结构就可以实现任何复杂算法

B．只要程序包含了三种基本结构中的任意一种，就是结构化程序

C．程序语法错误要在运行时才能发现

D．C 语言程序不需要包含 main()函数

【答案】A

【解析】结构化程序主要由循环结构、选择结构、顺序结构三种基本控制结构组成，它们组成的算法结构可以解决任何复杂的问题，A 项正确。结构化程序设计是以模块化设计为中心，将待开发的软件系统划分为若干个相互独立的模块，而不是说包含了三种基本结构就是结构化程序，B 项错误。程序语法错误是在编译过程中发现的，一般来说编译器只能检查语法和最简单的语义错误，而不检查程序的逻辑错误，C 项错误。C 程序总是从 main 函数开始执行，其他函数由 main 函数直接或间接调用执行，所以 C 语言程序必须包含 main 函数，D 项错误。答案选择 A 选项。

24．结构化程序由顺序、选择、循环三种基本结构组成，以下相关叙述错误的是（　　）。

A．三种基本结构不可以嵌套使用

B．顺序结构是按语句在程序中的先后顺序逐条执行，没有分支，没有转移

C．选择结构是根据不同的条件执行不同分支中的语句

D．循环结构是根据条件决定是否重复、重复执行多少次循环体语句

【答案】A

【解析】结构化程序主要由 3 种基本控制结构组成，顺序结构是最基本的算法结构，当执行由这些语句构成的程序时，将按这些语句在程序中的先后顺序逐条执行，没有分支，没有转移，没有步骤之间的相互约束，没有对某一步骤的多次使用，完全按照步骤的原有次序依次执行，B 选项叙述正确。选择结构根据不同的条件去执行不同分支中的语句，C 选项叙述正确。循环结构就是根据各自的条件，使同一组语句重复执行多次，D 选项叙述

正确。三种结构可以嵌套使用，A 选项叙述错误，答案选择 A 选项。

25．以下选项中叙述正确的是（　　）。

A．复杂任务可以分解成简单子任务

B．C 语言程序中的所有函数必须保存在同一个源文件中

C．全部三种基本结构都包含的才是结构化程序

D．C 语言程序可以定义多个不同内容的 main 函数

【答案】A

【解析】结构化程序设计把一个复杂的问题的求解过程分成阶段进行，即复杂任务可以分解成简单的任务，A 项正确。C 语言程序中的函数不一定要保存在同一个源文件中，外部函数可以被同程序中其他源文件中调用，B 项错误。结构化程序主要由 3 种基本控制结构组成，循环结构、选择结构、顺序结构，它们组成的算法结构可以解决任何复杂的问题。算法不一定要包含所有三种基本结构，也可以只包含一种或两种，C 项错误。C 程序由一个或若干个函数构成，程序中有且只能有一个主函数，即 main 函数，D 项错误。答案选择 A 选项。

26．以下叙述错误的是（　　）。

A．在进行模块化程序设计的时候，应首先完成每个模块的编写调试，再集中考虑主程序中的算法

B．同一程序各模块可由不同人员同时进行编写调试，可提高编写程序的效率

C．模块化的程序设计是采用自顶向下、逐步细化的原则

D．程序的每个模块都可通过三种基本结构实现

【答案】A

【解析】进行模块化设计时，首先设计框架，并定义和调试好各个模块之间的输入输出关系，完成各个模块的编写调试后再集中编译，A 项错误；各个模块可以由不同人员同时进行编写调试，提高编写程序的效率，B 项正确；模块化的程序设计采用自顶向下、逐步细化的原则，C 项正确；结构化程序使用三种基本结构可以解决任何复杂的问题，D 项叙述正确。答案选择 A 选项。

27．以下叙述正确的是（　　）。

A．只使用三种基本结构即可解决任何复杂问题

B．C 语言程序并不是必须要定义 main()函数

C．只要程序包含了任意一种基本结构，就肯定是结构化程序

D．程序中的语法错误只能在运行时才能显现

【答案】A

【解析】A 项正确，程序设计语言仅仅使用顺序、选择和循环三种基本控制结构就足以表达出各种其他形式结构的程序设计方法。B 项错误，C 语言程序必须要定义 main()函数；C 项错误，结构化程序设计是以模块化设计为中心，将待开发的软件系统划分为若干个相互独立的模块，而不是说包含了三种基本结构就是结构化程序；D 项错误，程序中的语法错误在编译时能显现。答案选择 A 选项。

# 第 2 章　算法——程序的灵魂

## 一、选择题

1．以下叙述中正确的是（　　）。

A．在 C 语言程序中，main 函数必须放在其他函数的最前面

B．每个后缀为.c 的 C 语言源程序都可以单独进行编译

C．在 C 语言程序中，只有 main 函数才可以单独进行编译

D．每个后缀为.c 的 C 语言源程序都应该包含一个 main 函数

【答案】B

【解析】main 函数可以在程序的任何位置。每一个可执行的 C 程序都必须有一个且只能有一个主函数。后缀名为.c 的 C 语言源程序都可以单独进行编译。main 函数只是让执行程序的系统知道该从哪里开始执行程序（从主函数处执行），其他有关这个程序的子函数是通过函数调用来实现其功能（不需 main 函数）。答案选择 B 选项。

2．以下叙述中错误的是（　　）。

A．C 语言编写的函数源程序，其文件名后缀可以是.c

B．C 语言编写的函数都可以作为一个独立的源程序文件

C．C 语言编写的每个函数都可以进行独立的编译并执行

D．一个 C 语言程序只能有一个主函数

【答案】C

【解析】C 源程序经过 C 编译程序编译之后生成一个后缀为.obj 的二进制文件（称为目标文件），然后由称为“连接程序”（Link）的软件，把此.obj 文件与 C 语言提供的各种库函数连接起来生成一个后缀为.EXE 的可执行文件。只有含有 main 函数的经过编译链接才能执行。答案选择 C 选项。

3．以下叙述中错误的是（　　）。

A．一个 C 程序可以包含多个不同名的函数

B．一个 C 程序只能有一个主函数

C．C 程序在书写时，有严格的缩进要求，否则不能编译通过

D．C 程序的主函数必须用 main 作为函数名

【答案】C

【解析】一个 C 程序有且只有一个主函数 main。一个 C 程序可以包含多个不同名字的子函数。C 程序在书写时没有严格的缩进要求。答案选择 C 选项。

4．以下叙述中正确的是（　　）。

A．C 语言规定必须用 main 作为主函数名，程序将从此开始执行

B．可以在程序中由用户指定任意一个函数作为主函数，程序将从此开始执行

C．C 语言程序将从源程序中第一个函数开始执行

D．main 的各种大小写拼写形式都可以作为主函数名，如：MAIN，Main 等

【答案】A

【解析】用户不能指定某函数为主函数，C 语言规定，程序从 main 函数开始执行，从 main 函数退出，C 语言函数名区别大小写。答案选择 A 选项。

5．下列叙述中错误的是（　　）。

A．C 程序可以由一个或多个函数组成

B．C 程序可以由多个程序文件组成

C．一个 C 语言程序只能实现一种算法

D．一个 C 函数可以单独作为一个 C 程序文件存在

【答案】C

【解析】一个 C 程序可以有一个或多个程序文件，也可以有一个或多个函数，所以一个 C 语言程序可以实现多种算法，答案选择 C 选项。

6．下列叙述中错误的是（　　）。

A．C 程序在运行过程中所有的计算都以二进制方式进行

B．C 程序在运行过程中所有的计算都以十进制方式进行

C．所有的 C 程序都需要在连接无误后才能运行

D．C 程序中整型变量只能存放整数，实型变量只能存放浮点数

【答案】B

【解析】在 C 程序运行过程中，编译器的作用是将程序转换为目标代码，目标代码都是二进制的。答案选择 B 选项。

7．以下叙述正确的是（　　）。

A．C 语言程序是由过程和函数组成的

B．C 语言函数可以嵌套调用，例如：fun(fun(x))

C．C 语言函数不可以单独编译

D．C 语言中除了 main 函数，其他函数不可作为单独文件形式存在

【答案】B

【解析】一个函数的返回值可以作为参数然后传给另一个函数，因此函数是可以嵌套调用的。A 项错误，C 语言程序只有函数构成，没有过程；C 项错误，编译系统的任务在于检查语法错误，只要符合语法规则的 C 程序都可以通过编译，就算是单独的函数也可以；D 项错误，在 C 语言中除 main()函数以外的其他函数可以和 main()函数在同一个 C 文件中，也可以单独处于其他的 C 文件，只要在使用到这些函数的 main()函数的 C 文件中用预编译指令“#include”包含进来即可。答案选择 B 选项。

8．对于一个正常运行的 C 程序，以下叙述中正确的是（　　）。

A．程序的执行总是从程序的第一个函数开始，在 main 函数结束

B．程序的执行总是从 main 函数开始

C．程序的执行总是从 main 函数开始，在程序的最后一个函数中结束

D．程序的执行总是从程序的第一个函数开始，在程序的最后一个函数中结束

【答案】B

【解析】一个正常运行的 C 程序总是从 main 函数开始执行，最后返回到 main 函数结束。答案选择 B 选项。

9．下列叙述中正确的是（　　）。

A．每个 C 程序文件中都必须要有一个 main 函数

B．在 C 程序中 main 函数的位置是固定的

C．C 程序中所有函数之间都可以相互调用

D．在 C 程序的函数中不能定义另一个函数

【答案】D

【解析】在 C 程序中，main 函数的位置可以任意，而且不管 main 函数位置怎么变化，程序都会以 main 函数作为入口，选项 B 错误；每个 C 程序（而不是每个 C 程序文件）必须有且只能有一个 main 函数，选项 A 错误；main 函数不能被其他函数调用，选项 C 错误；函数的定义不能放在另一个函数体内，但是声明可以，答案选择 D 选项。

10．以下叙述正确的是（　　）。

A．C 程序总是以 main()作为程序执行的起始行

B．main()函数若不带参数，其后面的一对圆括号可省略

C．函数体内的定义语句和可执行语句允许任意穿插出现

D．C 语言中的语句之间必须用分号作为分隔符

【答案】A

【解析】main 函数后面的括号告诉编译器这是一个函数，不可以省略，排除 B 选项；在复合语句中，不仅可以有执行语句，还可以有定义语句，定义语句应该出现在执行语句的前面，故排除 C 选项；C 语言中的某些语句可以不用分号，例如 if 语句，宏定义，故 D 选项错误。答案选择 A 选项。

11．以下关于 C 语言的叙述中正确的是（　　）。

A．C 语言中的注释不可以夹在变量名或关键字的中间

B．C 语言中的变量可以在使用之前的任何位置进行定义

C．在 C 语言算术表达式的书写中，运算符两侧的运算数类型必须一致

D．C 语言的数值常量中夹带空格不影响常量值的正确表示

【答案】A

【解析】A 项正确，C 语言中，程序中的注释可以出现在程序中任何合适的地方，但是，不能写在变量名或关键字的中间，一旦写在其中间，将会失去变量名或关键字的意义，导致报错；B 项错误，条件没有说全，应该是在有效的范围内，变量可以在任何位置定义，例如注释中定义无效。C 项错误，在 C 语言的算术运算符中，取余运算符“%”的两个运算分量必须是整数，但对于其他运算符，如“+”，“-”来说，两侧的运算符类型也可以不一样，例如左侧为一个字符类型，右侧为一个整数类型，系统在执行程序时会自动将字符类型转换为 ASCII 值进行运算；D 项错误，C 语言的数值表示时各个数位必须紧靠在一起，否则编译系统只会识别紧靠运算符的一部分数值，另一部分数值会发生语法错误。答案选择 A 选项。

12．以下叙述中正确的是（　　）。

A．C 程序的基本组成单位是语句

B．C 程序中的每一行只能写一条语句

C．简单 C 语句必须以分号结束

D．C 语言必须在一行内写完

【答案】C

【解析】C 程序的基本组成单位是函数，A 项错误；C 程序以分号作为每个语句结尾，一行能写多条语句，也可以将一条语句分几行书写，B、D 两项错误；C 语言中语句分为简单语句和复合语句。简单语句以分号作为结束。其中简单语句里面又有赋值语句、声明语句、结构化语句、函数调用语句和空语句。复合语句指用花括号{}将简单语句甚至另一些复合包起来，所以就以}作为语句结束的标记。答案选择 C 选项。

13．以下选项中叙述正确的是（　　）。

A．函数体必须由{开始

B．C 程序必须由 main 语句开始

C．C 程序中的注释可以嵌套

D．C 程序中的注释必须在一行完成

【答案】A

【解析】函数体是函数首部下面的花括号内的部分，所以函数体必须由{开始，A 选项正确。一个源程序文件可以包括预处理命令、全局声明、函数定义，程序总是从 main 函数开始执行的，不是 main 语句，B 选项错误。函数可以嵌套，注释不能嵌套，C 选项错误。C 程序中允许两种注释，以//开头的单行注释；以/*开始，以*/结束的块式注释，D 选项错误。答案选择 A 选项。

14．以下叙述中正确的是（　　）。

A．C 程序中的注释只能出现在程序的开始位置和语句的后面

B．C 程序书写格式严格，要求一行内只能写一个语句

C．C 程序书写格式自由，一个语句可以写在多行上

D．用 C 语言编写的程序只能放在一个程序文件中

【答案】C

【解析】C 程序的注释可以出现在 C 程序的任何位置，注释符号：“//”或“/*…*/”，选项 A 错误。C 程序中，一行内可写多个语句，每条语句用分号“;”结束，选项 B 错误，选项 C 正确。用 C 语言编写的程序可以放在多个程序文件中，用#include 命令行实现文件包含功能，选项 D 错误。答案选择 C 选项。

15．以下叙述中错误的是（　　）。

A．书写风格良好的程序执行效率高

B．书写风格良好的程序易读性好

C．C 程序可以在一行上写多条语句

D．C 程序允许将一条语句分写在多行上

【答案】A

【解析】书写风格与程序执行效率无关，程序执行效率与程序的数据结构有关，由算法的时间复杂度和空间复杂度决定，但书写风格会深刻地影响软件的质量和可维护性，良好的程序设计风格可以使程序结构清晰合理。良好的书写习惯一般一行写一条语句，这样方便阅读，但是一行写多条语句或者将一条语句分写在多行上是符合 C 程序编写规则的。答案选择 A 选项。

19．以下四个程序中，完全正确的是（　　）。

A.

```
#include <stdio.h>
main()
{
   /*/programming/*/
   printf("programming!\n");
}
```

B.

```
#include <stdio.h>
main()
{
   /*programming*/
   printf("programming!\n");
}
```

C.

```
#include <stdio.h>
main()
{
   /*/*programming*/*/
   printf("programming!\n");
}
```

D.

```
include <stdio.h>
main()
{
   /*programming*/
   printf("programming!\n");
}
```

【答案】B

【解析】A 项中，“main()”函数后面不能加分号；C 语言中注释语句的注释方法是：/*注释内容*/或//注释一行，且“/*”和“*/”不能嵌套使用，C 项错误；D 选项中预编译命令“include <stdio.h>”前缺少“#”号。答案选择 B 选项。

17．有以下程序

```
#include <stdio.h>
main()
{
   int a=0,b=0;
   /*给 a 赋值 a=10;
   b=20;给 b 赋值*/
   printf("a+b=%d\n",a+b);/*输出计算结果*/
}
```

程序运行后的输出结果是（　　）。

A. a+b=0

B. a+b=30

C. a+b=10

D. 出错

【答案】A

【解析】注释/*和*/之间的代码不参与编译，所以 a、b 的值仍为 0。答案选择 A 选项。

18. 关于 C 语言的变量名，以下叙述正确的是（　　）。

A. 变量名不可以与关键字同名

B. 变量名不可以与预定义标识符同名

C. 变量名必须以字母开头

D. 变量名是没有长度限制的

【答案】A

【解析】合法的标识符由字母（大、小写均可）、数字和下划线组成，并且必须以字母或下划线开头。关键字是指被 C 语言保留的，不能用作其他用途的标识符，它们在程序中都代表着固定的含义，用户不可重新定义，A 项正确、BC 两项错误。变量名没有长度限制，但不可超过编译器可以辨识的范围，D 项错误。答案选择 A 选项。

19. 以下选项中叙述正确的是（　　）。

A. C 语言的标识符可分为关键字、预定义标识符和用户标识符三类

B. C 语言的标识符可分为语句、变量和关键字三类

C. C 语言的标识符可分为函数名、变量和预定义标识符三类

D. C 语言的标识符可分为运算符、用户标识符和关键字三类

【答案】A

【解析】C 语言的标识符可分为关键字、预定义标识符和用户标识符三类，答案选择 A 选项。

20. C 语言中的标识符分为关键字、预定义标识符和用户标识符，以下叙述中正确的是（　　）。

A. 预定义标识符（如库函数中的函数名）可用作用户标识符，但失去原有含义

B. 用户标识符可以由字母和数字任意顺序组成

C. 在标识符中大写字母和小写字母被认为是相同的字符

D. 关键字可用作用户标识符，但失去原有含义

【答案】A

【解析】C 语言允许把预定义标识符重新定义另作他用，但这将失去预先定义的原意。B 项，标识符的第一个字符必须为字母或下划线；C 项，标识符区分大小写；D 项，关键字是指被 C 语言保留的，不能用作其他用途的标识符。答案选择 A 选项。

21. 关于 C 语言标识符，以下叙述错误的是（　　）。

A. 标识符可全部由数字组成

B. 标识符可全部由下划线组成

C. 标识符可全部由小写字母组成

D．标识符可全部由大写字母组成

【答案】A

【解析】C 语言标识符只能由字母、数字、下划线构成，且只能以字母、下划线开头，故答案选择 A 选项。

22．按照 C 语言规定的用户标识符命名规则，不能出现在标识符中的是（　　）。

A．大写字母

B．连接符

C．数字字符

D．下划线

【答案】B

【解析】C 语言中标识符只能由下划线、字母和数字组成，且不能以数字开头。答案选择 B 选项。

23．以下 C 语言用户标示符中，不合法的是（　　）。

A．_1

B．AaBc

C．a_b

D．a--b

【答案】D

【解析】C 语言中的标识符只能由字母、数字和下画线构成，且第一个字符必须是字母或下画线，同时不能与 C 语言中的关键字相同。D 项还有非法字符“-”。答案选择 D 选项。

24．以下选项中，能用作用户标识符的（　　）。

A．void

B．8_8

C．_0_

D．unsigned

【答案】C

【解析】标识符是由若干个字符组成的字符序列，用来命名程序的一些实体。C 语言定义标识符应遵循以下六种规则：①标识符由字母、数字或下划线组成；②第一个字符必须是字母或下划线；③标识符最多由 274 个字符组成；④在标识符中严格区分大小写字母；⑤关键字不能作为自定义的标识符在程序中使用。A、D 项皆为 C 语言的关键字，B 项第一个字符为数字，错误。答案选择 C 选项。

25．以下选项中可用作 c 语言中合法用户标识符的是（　　）。

A．_123

B．void

C．-abc

D．2a

【答案】A

【解析】合法标识符的命名规则是：标识符可以由字母、数字和下划线组成，并且第一个字符必须为字母或下划线，其中，关键字在程序中都代表着固定的含义，不能另作他用。B 项中 void 是关键字，不合法。C 项和 D 项没有以字母或下划线开头，不合法。答案选择 A 选项。

26．以下选项中合法的标识符是（　　）。

A．1_1

B．1-1

C．_11

D．1_ _

【答案】C

【解析】C 语言的标识符命名规则为：①只能由字母、数字和下划线 3 种字符组成；②标识符首位必须是字母或下划线；③不能与 C 语言中的关键字或保留字相同。AD 两项，标识符首位不能为数字；B 项，标识符首位

不能为数字且“-”为不合法的字符。答案选择 C 选项。

27．以下选项中不合法的标识符是（　　）。

A．print

B．FOR

C．&a

D．_00

【答案】C

【解析】标识符是由若干个字符组成的字符序列，用来命名程序的一些实体。语法规则为：①标识符由字母、数字或下划线组成；②第一个字符必须是字母或下划线；③标识符最多由 274 个字符组成；④在标识符中严格区分大小写字母；⑤关键字不能作为自定义的标识符在程序中使用。C 中有非法字符&。答案选择 C 选项。

28．下列定义变量的语句中错误的是（　　）。

A．double int_;

B．float US$;

C．char For;

D．int _int;

【答案】B

【解析】标识符由字母、数字、下划线组成。$是非法字符，不能出现在标识符中。答案选择 B 选项。

29．以下叙述中错误的是（　　）。

A．非零的数值型常量有正值和负值的区分

B．常量是在程序运行过程中值不能被改变的量

C．定义符号常量必须用类型名来设定常量的类型

D．用符号名表示的常量叫符号常量

【答案】C

【解析】在 C 语言程序中，可以用一个符号名来代表一个常量，称为符号常量，符号常量在定义是不需要类型，其本身就能表达其类型。答案选择 C 选项。

30．以下选项中关于 C 语言常量的叙述错误的是（　　）。

A．所谓常量，是指在程序运行过程中，其值不能被改变的量

B．常量分为整型常量、实型常量、字符常量和字符串常量

C．常量可分为数值型和非数值型常量

D．经常被使用的变量可以定义为常量

【答案】D

【解析】常量是指在程序运行过程中其值不能被改变的量。在 C 语言中，有整型常量、实型常量、字符常量和字符串常量等类型。整型常量和实型常量又称数值型常量，它们有正值和负值的区分。所谓变量是指在程序运行过程中其值可以改变的量。C 语言规定，程序中所有变量都必须先定义后使用。变量和常量有明显的区分。D 项，常量指在程序运行过程中，其值不能被改变的量，而变量是指在程序运行过程中，其值能被改变的量，因此变量不可以定义成常量。答案选择 D 选项。

31．以下选项中不能用作 C 程序合法常量的是（　　）。

A．1,234

B．"\123"

C．123

D．"\x7D"

【答案】A

【解析】常量分为数值常量和字符串常量，A 项错误。BD 两项为字符常量；C 项为数值常量。答案选择 A 选项。

32．以下选项中能表示合法常量的是（　　）。

A．整数：1,200

B．实数：1.5E2.0

C．字符斜杠：'\'

D．字符串："\007"

【答案】D

【解析】A 项错误，表达错误；B 项错误，E 后面应为整数，不能是小数；C 项错误；字符斜杠的表示方法为'\\'，因为第一个\表示是转义字符；D 项正确，表示空字符串，第一个字符为"\0"，正确。答案选择 D 选项。

33．以下选项中不能作为 C 语言合法常量的是（　　）。

A．'cd'

B．0.1e+6

C．"a"

D．'\011'

【答案】A

【解析】常量包括整型常量、实型常量、字符常量和字符串常量等。单引号表示字符常量，但不能包含字符串。表达字符串常量时需用双引号。A 项，在 C 语言中，字符常量是用单引号括起来的一个字符，'cd'包含了 2 个字符；B 项，0.1e+6 是实型常量的指数形式，代表 $0.1\times10^6$；C 项，"\a"是合法的字符串常量，\a 是一个非打印的转义字符表示响铃；D 项，'\011'是一个字符常量，\011 是一个用 3 位八进制表示的转移字符。答案选择 A 选项。

34．以下选项中，合法的 C 语言常量是（　　）。

A．1.234

B．'C++'

C．"\2.0

D．2Kb

【答案】A

【解析】C 语言中的常量：①整型常量，用不带小数点的数字表示；②实型常量，用带小数点的数字表示；③字符型常量，用带有单引号的一个字符表示；④字符串常量，用一对双引号括起来的一串字符。1.234 为实型常量，A 项正确；'C++'不合法，若改成"C++"则为字符串常量，B 项错误；"\2.0 不合法，不是任何类型常量，C 项错误；2Kb 不合法，若加上双引号"2Kb"为字符串常量，D 项错误。答案选择 A 选项。

35．以下选项中，能用作数据常量的是（　　）。

A．115L

B．0118

C．1-5e1.5

D．o115

【答案】A

【解析】C 语言中实型常量有两种表示：小数形式和指数形式。在指数形式中，字母 e（或 E）之前必须要有数字，且 e 或 E 后面的指数必须为整数，故 C 项错误；八进制整数常量以数字 0 开始，而不是 o，故 D 项错误。在八进制数中的有效数字为 0～7，故 B 项错误；L 是长整型数据标识，115L 为长整型常数即 long int，A 项正确。答案选择 A 选项。

36．以下选项中，合法的数值型常量是（　　）。

A．3.2

B．'X'

C．099

D．0xEH

【答案】A

【解析】A 项正确，3.2 是合法的实型常量；B 项错误，'X'为字符型常量不是数值型常量；C 项错误，以 0

开头，表示八进制，但八进制的每位由 0～7 中的一个组成；D 项错误，0x 和 H 冲突，都是表示十六进制。答案选择 A 选项。

37．以下选项中，合法的一组 C 语言数值常量是（　　）。

A．12.　0Xa23　4.5e0

B．028　.5e-3　-0xf

C．.177　4e1.5　0abc

D．0x8A　10,000　3.e5

【答案】A

【解析】A 项，C 语言中小数必须要有小数点，但是小数部分可以省略，12.是合法的常量；C 语言中十六进制数以 0X 或 0x 开头，0Xa23 是指十六进制数 a23；在 C 语言中，“e”或“E”后跟一个整数来表示以 10 为底的幂数，4.5e0 表示 $4.5*10^0$。B 项，028 表示的是八进制，以 0 开头，0 后面的数字只能在 0～7 之间。C 项，4e1.5，e 后面只能跟整数，0abc 表示八进制，0 后面的数字只能在 0～7 之间；D 项，10,000 中不能有逗号。答案选择 A 选项。

38．以下选项中正确的定义语句是（　　）。

A．double a;b;

B．double a=b=7;

C．double a=7,b=7;

D．double,a,b;

【答案】C

【解析】同一类型变量的定义时，不同变量之间需要用“,”分隔，选项 A 错误；定义变量时初始化赋值不能用等号连接，选项 B 错误；变量类型说明后面不能用逗号，而是用空格分离，选项 D 错误。答案选择 C 选项。

39．以下定义语句中正确的是（　　）。

A．int a=b=0;

B．char A=65+1,b='b';

C．float a=1,*b=&a,*c=&b;

D．double a=0.0;b=1.1;

【答案】B

【解析】A 项错误，变量定义的时候不能用连续用等号，等号在定义是初始化的一种；C 项错误，b 是指针变量，*c=&b 表示将一个二级指针赋值给一个一级指针，应该为*c=b 或者**c=&b；D 项，变量前为分号“;”表示前面的语句定义完毕，变量 b 的定义没有指明变量类型。答案选择 B 选项。

40．以下关于 C 语言数据类型使用的叙述中错误的是（　　）。

A．若只处理“真”和“假”两种逻辑值，应使用逻辑类型

B．若要保存带有多位小数的数据，可使用双精度类型

C．若要处理如“人员信息”等含有不同类型的相关数据，应自定义结构体类型

D．整数类型表示的自然数是准确无误差的

【答案】A

【解析】A 项错误，C 语言中没有逻辑类型，在 C++中才引入的；B 项正确，float 类型称为单精度类型，double 类型称为双精度类型，一般系统中，为 float 类型的变量分配 4 个字节的存储单元，为 double 类型的变量分配 8 个字节的存储单元。C 项正确，struct 结构体，可以用来描述包含多种基本类型的复杂对象。D 项正确，整数的表示是不存在误差的。答案选择 A 选项。

41．设有两行定义语句：

int scanf;

float case;

则以下叙述正确的是（　　）。

A．两行定义语句都不合法

B. 两行定义语句都合法
C. 第 1 行语句不合法
D. 第 2 行语句不合法
【答案】D
【解析】预定义标识符是系统已经有过定义的标识符，用户可以重新定义，可以作为变量名。scanf 为库函数名，属于预定义标识符，可以被用户重定义，第一行语句合法。C 语言关键字是被保留的，不能用作其他用途的一些标识符，它们在程序中都代表着固定的含义，用户不可重新定义。case 是选择结构 switch 语句中的关键字，不可被用户重定义，第二行语句不合法。答案选择 D 选项。

42. 阅读以下程序：

```
#include <stdio.h>
main()
{
   int case;
   float printF;
   printf("请输入 2 个数:");
   scanf("%d %f",&case,&printF);
   printf("%d %f\n",case,printF);
}
```

该程序编译时产生错误，其出错原因是（　　）。
A. 定义语句出错，case 是关键字，不能用作用户自定义标识符
B. 定义语句出错，printF 不能用作用户自定义标识符
C. 定义语句无错，scanf 不能作为输入函数使用
D. 定义语句无错，printf 不能输出 case 的值
【答案】A
【解析】在 C 语言中，关键字又称保留字，它是系统预先定义的，具有特定含义的标识符，故不允许用户重新定义。case 为 C 语言中的关键字，因此用户不能再定义标识符为 case 的变量。答案选择 A 选项。

43. C 语言中，最基本的数据类型包括（　　）。
A. 整型、实型、逻辑型
B. 整型、字符型、数组
C. 整型、实型、字符型
D. 整型、实型、结构体
【答案】C
【解析】C 语言中，最基本的数据类型包括整型、实型、字符型，答案选择 C 选项。

44. C 语言整数不包括（　　）。
A. 带小数点的整数
B. 正整数
C. 负整数
D. 无符号整数
【答案】A
【解析】C 语言整数包括正整数，负整数，无符号整数，不含带小数点的整数。答案选择 A 选项。

45. C 源程序中不能表示的数制是（　　）。
A. 二进制
B. 八进制
C. 十进制
D. 十六进制

【答案】A

【解析】C 语言中整型常量可以用十进制、八进制数、十六进制数来表示。虽然计算机只能识别二进制数，但二进制不能用源程序表示。答案选择 A 选项。

46．有以下程序：

```
#include<stdio.h>
main()
{
   int x=011;
   printf("%d\n",++x);
}
```

程序运行后的输出结果是（　　）。

A．12

B．11

C．10

D．9

【答案】C

【解析】x=011 表示按照八进制赋值，则十进制数为 9，所以输出++x 的结果为 10。答案选择 C 选项。

47．有以下程序，其中 k 的初值为八进制数

```
#include<stdio.h>
main()
{
   int k=011;
   printf("%d\n",k++);
}
```

程序运行后的输出结果是（　　）。

A．12

B．11

C．10

D．9

【答案】D

【解析】整型变量 k 的值“011”是用八进制表示的，即十进制的“9”，而输出格式为%d，即十进制格式，所以输出为“9”，然后 k 自增 1。答案选择 D 选项。

48．有如下程序：

```
#include<stdio.h>
main()
{
   int x=072;
   printf("%d\n",x+1);
}
```

程序运行后的输出结果是（　　）。

A．59

B．73

C．115

D．72

【答案】A

【解析】整型常量有 3 种，①十进制整常量，没有前缀，输出格式控制符为%d；②八进制整常量，以 0 作为前缀，输出格式控制符为%o；③十六进制整常量，以 0X 或 0x 作为前缀，输出格式控制符为%x。八进制数 072 表示成十进制数为 58，即 x=072=58，以十进制格式输出 x+1=59，答案选择 A 选项。

49．有如下程序：

```
#include<stdio.h>
main()
{
   int x=0x13;
   printf("%d\n",x+1);
}
```

程序运行后的输出结果是（　　）。

A．12

B．14

C．20

D．13

【答案】C

【解析】整型常量中，十进制整常量没有前缀，输出格式控制符为%d；八进制整常量以 0 作为前缀，输出格式控制符为%o；十六进制整常量以 0X 或 0x 作为前缀，输出格式控制符为%x。十六进制数 0x13 表示成十进制数为 19，以十进制格式输出 x+1=19+1=20，答案选择 C 选项。

50．有以下程序：

```
#include<stdio.h>
main()
{
   int x=0x13;
   printf("INT:%d\n",x+1);
}
```

程序运行后的输出结果是（　　）。

A．INT:14

B．INT:13

C．INT:12

D．INT:20

【答案】D

【解析】0x13 是十六进制数，%d 需要输出十进制数，换算成十进制：$16\times1+16^0\times3=16+3=19$，输出 INT=x+1=20，因此答案选择 D 选项。

51．若函数中有定义语句：

int k;

则（　　）。

A．系统将自动给 k 赋初值 0

B．这时 k 中的值无定义

C．系统将自动给 k 赋初值-1

D．这时 k 中无任何值

【答案】B

【解析】int k;这条语句是定义一个整型变量 k，这是动态定义，编译程序仅为 k 开辟存储单元，而没有在存储单元中存放任何初值，此时变量中的值时无意义的。若是静态定义，则会自动初始化，其默认值为 0。答案选

择 B 选项。

52．有以下程序

```
#include<stdio.h>
main()
{
   int s,t,A=10;
   double B=6;
   s=sizeof(A);
   t=sizeof(B);
   printf("%d,%d\n",s,t);
}
```

在 VC++2010 平台上编译运行，程序运行后的输出结果是（　　）。

A．2,4

B．4,4

C．4,8

D．10,6

【答案】C

【解析】sizeof 的作用就是返回一个对象或者类型所占的内存字节数。在 VC++2010 中整型占 4 个字节，双精度实型占 8 个字节。答案选择 C 选项。

53．关于 C 语言中数的表示，以下叙述中正确的是（　　）。

A．只有整型数在允许范围内能精确无误地表示，实型数会有误差

B．只要在允许范围内整型和实型都能精确表示

C．只有实型数在允许范围内能精确无误地表示，整型数会有误差

D．只有八进制表示的数才不会有误差

【答案】A

【解析】实型数据在内存中存储的二进制位数是有限的，而一个十进制实数转化为二进制实数时，其有效数字位数有可能会超过尾数的存储长度，从而导致有效数字丢失而产生误差。在整型数允许范围之内，二进制可以表示任意一个整数。答案选择 A 选项。

54．下列形式中不合法的常量是（　　）。

A．2.E8

B．-.28

C．-028

D．2e-8

【答案】C

【解析】-028 表示的是八进制的整型常量，但八进制的数字只能用 0～7 表示。AD 两项为指数形式的实数表示，在 e 或 E 的前面必须要有数字，且 e 或 E 后面的指数必须为整数；B 项，为整数常量。答案选择 C 选项。

55．以下选项中表示一个合法的常量是（说明：符号 u 表示空格）（　　）。

A．9u9u9

B．0Xab

C．123E0.2

D．2.7e

【答案】B

【解析】十六进制数用 0x 或 0X 开头。0Xab 是指十六进制数，是合法的常量，B 项正确。A 项多位数字之间不能用空格间隔，A 项错误；当用指数形式表示浮点数据时，E 的前后都要有数据，并且 E 的后面数要为整数，C、D 错误。答案选择 B 选项。

56．以下选项中可用作 C 程序合法实数的是（　　）。

A．.1e0

B．3.0e0.2

C．E9

D．9.12E

【答案】A

【解析】C 程序的合法实数有小数和指数两种表示形式。其中，对于用指数形式表示的实数来说，字母 e 或 E 之前必须要有数字，且字母 e 或 E 后面的指数必须为整数。B 项，e 后的指数不能为小数形式；C 项，E 前必须要有数字；D 项，E 后缺少整数形式的指数。答案选择 A 选项。

57．以下选项中，能用作数据常量的是（　　）。

A．o115

B．0118

C．1-5e1.5

D．115L

【答案】D

【解析】C 语言中实型常量有两种表示：小数形式和指数形式。在指数形式中，字母 e（或 E）之前必须要有数字，且 e 或 E 后面的指数必须为整数，故 C 项错误；八进制整数常量以数字 0 开始，而不是 o，故 A 项错误。在八进制数中的有效数字为 0～7，故 B 项错误；L 是长整型数据标识，115L 为长整型常数即 long int，D 项正确。答案选择 D 选项。

58．以下选项中，合法的 C 语言实数是（　　）。

A．3.1e0.4

B．.2e0

C．E13

D．7.12E

【答案】B

【解析】实型常量指数形式由十进制数加阶码标志“e”或“E”以及阶码（只能为整数，可以带符号）组成。.2e0 为指数形式实数，B 项正确。3.1e0.4，阶数不是整数，A 项错误。E13 阶码标志前缺少十进制数，C 项错误。7.12E 缺少阶码，D 项错误。答案选择 B 选项。

59．以下不合法的数值常量是（　　）。

A．8.0E0.5

B．1e1

C．011

D．0xabcd

【答案】A

【解析】实型常量指数形式中，字母 e（或 E）前后必须都要有数字，且 e（或 E）后面的指数必须为整数。答案选择 A 选项。

60．以下选项中，合法的数值型常量是（　　）。

A．3.1415

B．"A"

C．092

D．0xDH

【答案】A

【解析】数值型常量包括整型常量和实型常量。整型常量有三种：①十进制常量，用 0～9 表示，不能以 0 开头；②八进制常量，用 0～7 表示，必须用 0 开头；③十六进制常量，用 0～9 和 A～F（a～f）表示，必须以 0x 或 0X 开头。实型常量：用带小数点的数字表示，其值有两种表达形式，分别为十进制小数形式和指数形式。

3.1415 为实型常量，A 项正确。"A"为长度为 1 的字符串常量，B 项错误。092 按照格式来说是八进制整型常量，但是八进制不会出现 9，C 项错误。0xDH 按照格式说是十六进制整型常量，但是不应该出现 H，D 项错误。答案选择 A 选项。

61．在 C 语言中，以下选项中不能正确表示 10×1000 之值的是（　　）。

A．1.0E4.0

B．1.E4

C．10E3

D．1.0e4

【答案】A

【解析】实型常量用带小数点的数字表示，其值有两种表达形式，分别为十进制小数形式和指数形式。指数形式由十进制数加阶码标志“e”或“E”以及阶码（只能为整数，可以带符号）组成，而 A 项中阶码 4.0 是浮点数，所以 A 项不能正确表示 10×1000 之值，B、C、D 项都表示正确并且值为 10000，答案选择 A 选项。

62．若有定义语句：

int a=10;double b=3.14;

则表达式'A'+a+b 值的类型是（　　）。

A．char

B．int

C．double

D．float

【答案】C

【解析】在多目运算符相关联的运算数中，如果类型不一致，系统将自动进行类型转换，使两边的类型达到一致后，再进行运算，转换的规则是由“低到高”。'A'是字符型，a 是整型，b 是 double 型。字符型数据占 1 个字节，整型占 2 个字节，doub1e 型占 8 个字节，故三者相加后为 double 型，答案选择 C 选项。

63．若有定义：

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

则执行表达式(a=b+c)||(++b)后，a、b、c 的值依次为（　　）。

A．1,2,3

B．5,3,2

C．5,2,3

D．5,3,3

【答案】C

【解析】||表示或运算，当第一个表达式为真时，第二个表达式不执行。根据运算符的优先级规则，先计算(a=b+c)，将 b+c 的值赋值给 a，则 a=5，而||右边的括号不会被执行，所以 b=2，c=3。答案选择 C 选项。

64．C 语言程序中，运算对象必须是整型数的运算符是（　　）。

A．&&

B．/

C．%

D．*

【答案】C

【解析】模运算中不能出现浮点数，运算对象只能是整数。答案选择 C 选项。

65．以下不能用于实型数据的运算符是（　　）。

A．%

B．/

C．*

D．+

【答案】A

【解析】“%”符号两边必须是整数，答案选择 A 选项。

66．设 x，y，z 均为实型变量，代数式 $\frac{x}{yz}$ 在 C 语言中的正确写法是（　　）。

A．x/y*z

B．x%y%z

C．x/y/z

D．x*z/y

【答案】C

【解析】%是取余运算符，不符合。运算符*、/的结合顺序是从左到右，所以 x 先除以 y，再除以 z。答案选择 C 选项。

67．以下不能正确表示代数式 $\frac{2ab}{cd}$ 的 C 语言表达式是（　　）。

A．2*a*b/c/d

B．a*b/c/d*2

C．a/c/d*b*2

D．2*a*b/c*d

【答案】D

【解析】*与/优先级相同，采用左结合的方式。D 项若改为：2*a*b/(c*d)则为正确。答案选择 D 选项。

68．若在程序中变量均已定义成 int 类型，且已赋大于 1 的值，则下列选项中能正确表示代数式 1/abc 的表达式是（　　）。

A．1.0/a/b/c

B．1/(a*b*c)

C．1.0/a*b*c

D．1/a/b/(double)c

【答案】A

【解析】abc 均大于 1，所以表达式 1/abc 小于 1，需要用浮点类型表示。若要计算表达式值，需要使其自动转化成浮点类型，A 项正确。B 项变量与常量均为整型，不会自动转换为浮点类型，B 项错误。C 项表示表达式 bc/a，错误。D 项，算数运算法结合性自左向右，先计算 1/a，结果为 0，之后的计算无论是否转换数据类型结果均为 0，D 项错误。答案选择 A 选项。

69．设变量 x 为 long int 型并已正确赋值，以下表达式中能将 x 的百位上的数字提取出来的是（　　）。

A．x/10%100

B．x%10/100

C．x%100/10

D．x/100%10

【答案】D

【解析】x/100 结果的个位数上的数字是原来 x 的百位数上的数字，则要得到这个数字只需要再%10 即可。答案选择 D 选项。

70．以下可以将任意一个两位整数 n 的个位数与十位数对换的表达式为（　　）。

A．(n-n/10×10)×10+n/10

B．n-n/10×10+n/10

C．n/10+(n-n/10)

D．(n-n/10)×10+(n/10)

【答案】A

【解析】A 项 n/10 得到十位数字，n-n/10 得到个位数字，将个位数乘以 10 加上十位数字，(n-n/10×10)×10+n/10 实现将任意一个两位整数个位数与十位数对换，A 项正确。B 项 n-n/10×10 得到个位数字，n/10 为十位数字，n-n/10×10+n/10 实现两位整数个位数与十位数之和，B 项错误。C 项 n/10+(n-n/10)=n，不能实现题目中要求，C 项错误。D 项 n/10 是十位数字，n-n/10 为原两位整数减去十位数字，不是个位数字，D 项错误。答案选择 A 选项。

71．设有以下程序段：

```
int y;
y=rand()%30+1;
```

则变量 y 的取值范围是（　　）。

A．0≤y≤30

B．0<y≤30

C．1<y<30

D．1<y≤30

【答案】B

【解析】rand 函数产生随机整数，任何整数对 30 求余得到的整数范围为 0～29，则 y 的取值范围为 1≤y≤30 或者 0<y≤30，y 是整数，0<y≤30。答案选择 B 选项。

72．若有定义语句：int x=12,y=8,z;，在其后执行语句 z=0.9+x/y;，则 z 的值为（　　）。

A．1.9

B．1

C．2

D．2.4

【答案】B

【解析】由于 x，y，z 都是整型数据，所以 x 除以 y 的值为整型数值 1，之后 1 和 0.9 相加得到 1.9，再转换为整型数 1 赋给整型变量 z。答案选择 B 选项。

73．表达式：(int)((double)9/2)-9%2 的值是（　　）。

A．0

B．3

C．4

D．5

【答案】B

【解析】运算符“/”“%”的优先级高于“-”，所以先进行除法和求余运算，再进行减法运算。强制类型转换表达式的形式：（类型名）（表达式）。“9/2”结果为 4，转换成 double 类型再转换成 int 类型结果依然为 4，9%2 的结果为 1，最后结果为 4-1=3。答案选择 B 选项。

74．设有定义：int x=7,y=12;，则以下表达式值为 3 的是（　　）。

A．(y%=x)-(x%=5)

B．y%=(x%=5)

C．y%=x-x%5

D．y%=(x-x%5)

【答案】A

【解析】A 项正确，a%=b 表示 a=a%(b)，A 项可改写成 y=y%x，x=x%5，再计算 y-x 计算的结果为 3，满足题意；B 项为 0，C 项为 2，D 项等同于 C 项。答案选择 A 选项。

75．表达式 3.6-5/2+1.2+5%2 的值是（　　）。

A．4.8

B．3.8

C．3.3

D．4.3

【答案】B

【解析】3.6-5/2+1.2+5%2 = 3.6-2+1.2+1=3.8。需要注意的是，两个整数相除仍然得到整数，小数部分被省略，5/2=2。答案选择 B 选项。

76．表达式 a+=a-=a=9 的值是（　　）。

A．9

B．-9

C．18

D．0

【答案】D

【解析】a+=a-=a=9 计算顺序为 a+=(a-=(a=9))，首先 a=9，a-=9，即 a=a-9=9-9=0；最后 a+=0，即 a=a+0=0+0=0。答案选择 D 选项。

77．以下能正确表述算式 sin(2πr+30°)的 C 语言表达式是（　　）。

A．sin(2*3.14*r+3.14*30/180.0)

B．sin(2*π*r+30)

C．sin(2*3.14*r+30)

D．sin(2*3.14*r+30*3.14/360.0)

【答案】A

【解析】A 项正确。sin 是库函数，其参数中的角度要求用弧度制表示。C 语言中 π 不是已定义的常量，需要用户自定义或者直接使用 3.14 代替 π。B 项未替换 π 也没有将 30 度换算成弧度，错误。C 项没有将 30 度换算成弧度，错误。D 项弧度换算错误，π 对应于 180°，应该除以 180.0 而不是 360.0。答案选择 A 选项。

78．有以下定义

int a;

long b;

double x,y;

则以下选项中正确的表达式是（　　）。

A．a%(int)(x-y)

B．a==x!=y

C．(a*y)%b

D．y=x+y=a

【答案】A

【解析】%运算是取两整数相除后余数的运算符，它只适用于整数的运算。A 项正确，x-y 结果为 double 型，但是通过强制类型转换将其转换为 int 型；B 项错误，a==x!=y 中==和!=是同一个优先级的，先运行 a==x，而因为 double 是占 8 位的，不能自动转换，必须要强制类型转换，a==(double)x!=y 才是正确的；C 项错误，(a*y)%b 中的(a*y)为 double 型；D 项错误，x+y 不能作为左值。答案选择 A 选项。

79．设有定义：

int x=2;

以下表达式中，值不为 6 的是（　　）。

A．x*=x+1

B．x++,2*x

C．x*=(1+x)

D．2*x,x+=2

【答案】D

【解析】A 项，因为赋值运算优先级最低，故先执行 x+1，赋值符号右侧为 3，然后再计算 x*=3，所以 x=2*3=6；B 项，x++得 3 然后 2*x 得 6；C 项与 A 项原理一致。D 项，2*x 虽然结果为 4，但没有赋值，此时 x 的值仍为 2，所以最终为 4。答案选择 D 选项。

80．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．x+1=y

B．++x,y=x--

C．x=x+10=x+y

D．double(x)/10

【答案】B

【解析】B 项正确，++x 是前缀表达式，y=x--是复合语句，先进行 x--，然后把自减后的值赋给 y。A 项错误，x+1 是右值，不能被赋值；C 项错误，x+10 是右值，不能被赋值；D 项错误，应改成(double)x/10，double (x)/10 是表示声明了一个 double 变量，它名字是(x)/10，显然不符合 C 语言语法。答案选择 B 选项。

81．已知大写字母 A 的 ASCII 码是 65，小写字母 a 的 ASCII 码是 97。以下不能将变量 c 中的大写字母转换为对应小写字母的语句是（　　）。

A．c=('A'+c)%26-'a'

B．c=c+32

C．c=c-'A'+'a'

D．c=(c-'A')%26+'a'

【答案】A

【解析】A 项，模 26 运算后得到 0~25 之间的数，再与'a'相减结果是负数；B 项，ASCII 码表中，同一字母的小写码比大写码数值上大 32；C 项，'a'-'A'=32；D 项，c-'A'取值在 0～26，等价于 C 项。答案选择 A 选项。

82．以下选项中，当 x 为大于 1 的奇数时，值为 0 的表达式是（　　）。

A．x%2==1

B．x/2

C．x%2!=0

D．x%2==0

【答案】D

【解析】当 x 为大于 1 的奇数，x%2==1，则表达式 x%2==0 为假（即值为 0），答案选择 D 选项。

83．以下选项中，值为 1 的表达式是（　　）。

A．'1'-0

B．1-'0'

C．1-'\0'

D．'\0'-'0'

【答案】C

【解析】ASCII 码表，'0'～48，'1'～49，'\0'～0，答案选择 C 选项。

84．有以下程序：

```
#include<stdio.h>
main()
{
   int a;
   a=(int)1.99*3;
   printf("%d\n",a);
}
```

程序的运行结果是（　　）。

A．3

B．5

C．6

D．结果不确定

【答案】A

【解析】()的优先级大于*，故先将 1.99 强制转化为 1，执行语句 a=(int)1.99*3；计算得 a=1×3=3。答案选择 A 选项。

85．有以下程序：

```
#include <stdio.h>
main()
{
   int A=0,B=0,C=0;
   C=(A-=A-5);
   (A=B,B+=4);
   printf("%d,%d,%d\n",A,B,C);
}
```

程序运行后输出的结果是（　　）。

A．0,4,5

B．4,4,5

C．4,4,4

D．0,0,0

【答案】A

【解析】C 语言中可以将多条语句放在同一行，用“;”隔开。赋值运算和逗号运算都是从左到右结合。首先执行 A-=A-5，即 A=A-(A-5)，得 A 的值为 5，然后执行 C=(A-=A-5)，即将 A 的值 5 赋给变量 C，使得 C 的值也为 5。然后执行下句逗号表达式中的 A=B，把 B 的值 0 赋给 A，此时 A 的值为 0，然后执行 B=B+4，使得 B 的值为 4，最后打印输出。答案选择 A 选项。

86．有以下程序

```
#include<stdio.h>
main()
{
   int a,b,k,m,*p1,*p2;
   k=1,m=8;
   p1=&k,p2=&m;
   a=/*p1-m;
   b=*p1+*p2+6;
   printf("%d",a);
   printf("%d\n",b);
}
```

编译时编译器提示错误信息，你认为出错的语句是（　　）。

A．a=/*p1-m

B．b=*p1+*p2+6;

C．k=1,m=8;

D．p1=&k,p2=&m;

【答案】A

【解析】a=/*p1-m;语句不符合语法规范，可写作 a/=(*p1-m);，表示 a=a/(*p1-m);。答案选择 A 选项。

87．有以下程序：

```
#include<stdio.h>
int sub(double a,double b)
{
   return(int)(a-b);
}
main()
{
   printf("%d\n",sub(3.8,2.1));
}
```

程序运行后的输出结果是（　　）。

A．2.0

B．1.7

C．2

D．1

【答案】D

【解析】在类型转换过程中，如果较高类型转换成较低类型，直接忽略多余位数。程序执行过程为：调用函数 sub(3.8,2.1)，3.8-2.1=1.7（double 类型），(int)强制转换将 1.7 转换成 int 类型 1，然后返回 1 并输出。答案选择 D 选项。

88．有以下程序：

```
#include<stdio.h>
int sub(double a,double b)
{
   return(int)(a-b-1.3);
}
main()
{
   printf("%d\n",sub(3.2,4.1));
}
```

程序运行后的输出结果是（　　）。

A．-2

B．1.7

C．-3

D．2.0

【答案】A

【解析】sub 函数输出(int)(a-b-1.3)将结果强制转化为整型。(a-b-1.3)=-2.2，(int)(-2.2)=-2，sub(3.2,4.1)=-2，答案选择 A 选项。

89．有以下程序：

```
#include<stdio.h>
float fun(double a)
{
   double x;
   x=a-(int)a;
   return x;
}
main()
{
   double a=3.1415;
   printf("%f\n",fun(a));
}
```

程序的运行结果是（　　）。

A. 3.000000

B. 3.141500

C. 0.141500

D. 0.000000

【答案】C

【解析】在 fun 函数中，x=a-(int)a 表示 x 取值为 a 的小数部分，因此，输入 a 为 3.1415，输出 x=0.141500。答案选择 C 选项。

90. 有以下程序：

```
#include <stdio.h>
#include <math.h>
main()
{
   int a = 1,b=4,c = 2;
   double x = 10.5,y = 4.0,z;
   z = (a + b)/c + sqrt(y)*1.2/c+x;
   printf("%f\n",z);
}
```

程序运行后的输出结果是（　　）。

A. 13.700000

B. 14.000000

C. 15.400000

D. 14.900000

【答案】A

【解析】sqrt 为平方根计算函数，a、b、c 三个变量都是整型变量，(a+b)/c 结果也取整型得 2，所以有 z=(1+4)/2+2*1.2/2+10.5=13.7。%f 格式输出后为 13.700000。答案选择 A 选项。

91. 若变量已正确定义并赋值，则错误的赋值语句是（　　）。

A. a=a+1;

B. a=sizeof(double);

C. a=d||c;

D. a+1=a;

【答案】D

【解析】赋值号的左边必须是一个代表某个存储单元的变量名，赋值号的右边必须是 C 语言中合法的表达式。赋值运算的功能是先求出右边表达式的值，然后把此值赋给赋值号左边的变量。答案选择 D 选项。

92．设 a、b、c 是整型变量且均已赋值，则以下选项中错误的赋值语句是（　　）。

A．a=(b=3)=1;

B．a=(b=2)+c;

C．a=b=c+10;

D．a=1+(b=c=2);

【答案】A

【解析】赋值运算结合性为由右向左结合，赋值运算符左值为变量，右值为变量或常量或表达式，且左右两边数据类型相同才能实现赋值。A 项中，赋值运算(b=3)=1，左值为表达式，不是变量不能被赋值，A 项错误。B 项运算过程为，先赋值 b=2，再计算 b+c，将结果赋给 a，是正确的赋值语句。C 项运算过程为，先计算 c+10 将结果赋给 b，再将 b 赋给 a，是正确的赋值语句。D 项运算过程为，先将 2 赋给 c，再将 c 赋给 b，然后计算 1+b，将结果赋给 a，是正确的赋值语句。答案选择 A 选项。

93．设有定义：

int a=0,b=1,c=1;

以下选项中，表达式值与其他三个不同的是（　　）。

A．b=a==c

B．a=b=c

C．a=c==b

D．c=a!=c

【答案】A

【解析】赋值运算结合性为由右向左结合，成功实现赋值后以左值为返回值。逻辑表达式成立则返回 1，不成立返回 0。A 项，逻辑表达式 a==c 不成立，则 b=0，表达式值为 0；B 项，将 c 赋值给 b，b=1，再将 b 赋给 a，a=1，表达式值为 1；C 项，逻辑表达式 c==b 成立，则 a=1，表达式值为 1；D 项，逻辑表达式 a!=c 成立，则 c=1，表达式值为 1。A 项与其他项不同。答案选择 A 选项。

94．若有定义：

int a,b,c;

以下选项中的赋值语句正确的是（　　）。

A．a=(b=c)+1;

B．(a=b)=c=1;

C．a=(b==c)=1;

D．a+b=c;

【答案】A

【解析】赋值运算结合性为由右向左结合，赋值运算符左值为变量，右值为变量或常量，且左右两边数据类型相同才能实现赋值。A 项中，将 c 赋值给 b，然后将 b 加 1 赋值给 a，是正确的赋值语句，正确。B 项中，将 1 赋值给 c，赋值运算(a=b)=c 中左值为表达式，不是变量不能被赋值，错误。C 项中赋值运算(b==c)=1 左值为逻辑表达式，不是变量不能被赋值，错误。D 项左值 a+b 为表达式，不是变量不能被赋值，D 项错误。答案选择 A 选项。

95．设 a、b、c 是整型变量，且已正确赋初值，以下选项中错误的赋值语句是（　　）。

A．a=1%(b=c=2);

B．a=(b=3)*c;

C．a=b=c/10;

D．a=2=(b=9)=1;

【答案】D

【解析】D 项，“()”具有最高优先级，b 先赋值 9，再根据赋值运算符号自右向左进行赋值，再给 b 赋值为 1，再计算 2=b，赋值运算符左边必须是一个变量，但数值 2 是常量，语法错误。答案选择 D 选项。

96．若已有定义语句：int a,b,c;，且变量已正确赋初值，则以下选项中正确的赋值表达式是（　　）。

A．a=(b=c)+8;

B．(a=b)=c=9;

C．a=(b==c)='A';

D．a+b=c+1;

【答案】A

【解析】A 项正确，将 c 赋值给 b 后又加上 8，然后再赋值给 a；B、D 项错误，赋值运算符左边必须是一个变量；C 项错误，(b==c)是个表达式，语法错误。答案选择 A 选项。

97．若想给已定义为 int 型的变量 a、b、c、d 赋整数 1，以下选项中错误的语句是（　　）。

A．a=b,b=c,c=d,d=1;

B．a=b=c=d=1;

C．a=1,b=a,c=b,d=c;

D．d=1,c=d,b=c,a=b;

【答案】A

【解析】由于 C 语言逗号表达式的的执行顺序是从左向右，赋值运算符是自右向左，A 选项先执行 a=b 时，b 为空值，故不正确；所以在连续赋值时一定要保证赋值运算符的右项有确定的值。答案选择 A 选项。

98．若有定义

double a=22;int i=0,k=18;

则不符合 C 语言规定的赋值语句是（　　）。

A．a=a++,i++

B．i=(a+k)<=(i+k)

C．i=a%11

D．i=!a

【答案】C

【解析】C 项错误，在 C 语言的算术运算符中，取余运算符“%”的左右两侧的两个运算分量必须是整数。A 项，赋值运算符的优先级高于“,”运算法，先进行赋值；B 项，将“=”右边的逻辑表达式的计算结果赋给变量 i；D 项，首先对变量 a 求逻辑非操作，然后将结果赋值给变量 i。答案选择 C 选项。

99．设变量已正确定义并赋值，以下正确的表达式是（　　）。

A．x=y*5=x+z

B．int(15.8%5)

C．x=y+z+5,++y

D．x=25%5.0

【答案】C

【解析】赋值运算左边必须是单一变量名。A 项错误，“y*5=x+z”部分是非法赋值。BD 两项错误，求余运算中的操作对象只能是整型。C 项，为逗号表达式。答案选择 C 选项。

100．若变量均已正确定义并赋值，以下合法的 C 语言赋值语句是（　　）。

A．x=y==5;

B．x=n%2.5;

C．x+n=i;

D．x=5=4+1;

【答案】A

【解析】A 项正确，y==5 返回 0 或者 1，然后赋值给 x。B 项错误，浮点数不能参与模运算；C 项错误，赋值运算符左边只能是单一变量，x+n 是右值，不能给它赋值；D 项错误，5 是常量，不能被赋值。答案选择 A 选项。

101．以下选项中合法的 C 语言赋值语句是（　　）。

A．++i;

B．a=b=34

C．a=3,b=9

D．k=int(a+b);

【答案】A

【解析】B 项和 C 项缺少分号，不合法。D 项类型转换不合法，应为 k=(int)(a+b)。答案选择 A 选项。

102．设有定义：

int x=11,y=12,z=0;，

以下表达式值不等于 12 的是（　　）。

A．(z,x,y)

B．(z=x,y)

C．z=(x,y)

D．z=(x==y)

【答案】D

【解析】逗号表达式的计算过程是从左到右逐个求每个表达式的值，取最右边一个表达式的值作为该逗号表达式的值。赋值运算结合性为由右向左结合，赋值运算符左值为变量，右值为变量或常量或表达式，且左右两边数据类型相同才能实现赋值。成功实现赋值后以左值为返回值。逻辑表达式成立则返回 1，不成立返回 0。D 选项逻辑表达式 x==y 不成立，则 z=0，表达式值为 0。B 选项逗号表达式 x，y 取 y 值为表达式值，然后赋值给 z=12，表达式值为 12。C 选项逗号表达式(x,y)取 y 值为表达式值，然后赋值给 z=12，表达式值为 12。A 选项逗号表达式(z,x,y)取 y 值为表达式值 12。答案选择 D 选项。

103．若有定义语句：

int x=10;

则表达式 x-=x+x 的值为（　　）。

A．-20

B．-10

C．0

D．10

【答案】B

【解析】单目加运算符优先级高于赋值运算符，所以先做 x+x 结果为 20，再做 x-20，结果为-10，然后赋值给 x。所以答案选择 B 选项。

104．若有定义语句：

int a=12;

则执行语句：a+=a-=a*a;后，a 的值是（　　）。

A．-264

B．552

C．144

D．264

【答案】A

【解析】赋值运算结合性为自右向左，a+=a-a*a 相当于 a=a-a*a，a=a+a；自右向左计算过程为 a=a-a*a=-132，a=a+a=-264。答案选择 A 选项。

105．有以下定义

int a;

long b;

double x,y;

则以下选项中正确的表达式是（　　）。

A．(a*y)%b

B．a==x<>y

C．a%(int)(x-y)

D．y=x+y=a

【答案】C

【解析】%运算是取两整数相除后余数的运算符，它只适用于整数的运算。A 项错误，(a*y)%b 中的(a*y)为 double 型；B 项错误，C 语言中没有<>运算符；C 项正确，x-y 结果为 double 型，但是通过强制类型转换将其转换为 int 型； D 项错误，x+y 不能作为左值。答案选择 C 选项。

106．有以下程序

```
#include <stdio.h>
#include <math.h>
main()
{
   int a = 3;
   printf("%d\n",(a+=a-=a*a));
}
```

程序运行后的输出结果是（　　）。

A．-12

B．9

C．0

D．3

【答案】A

【解析】C 语言中，表达式从右向左计算。a+=a-=a*a;可以写成 a-=a*a; a+=a;，而 a-=a*a 等价于 a = a-(a*a) = 3-(3*3) = -6；a+=a 等价于 a = a+a =(-6)+(-6) = -12。答案选择 A 选项。

107．设有定义：

int k=0;

以下选项的四个表达式中与其他三个表达式的值不相同的是（　　）。

A．++k

B．k+=1

C．k++

D．k+1

【答案】C

【解析】后缀表达式，先赋值，后自增。k++表达式值为 k 的值 0；其余三项为 k+1 的值，即 1。答案选择 C 选项。

108．设变量均已正确定义并且赋值，以下与其他三组输出结构不同的一组语句是（　　）。

A．x++;printf("%d\n"，x);

B．n=++x;printf("%d\n",n);

C．++x;printf("%d\n",x);

D．n=x++;printf("%d\n",n);

【答案】D

【解析】“++”和“--”运算，当以前缀形式出现时，则先进行加一或减一操作，再取值，当以后缀形式出现时，则先取值，再进行加一或减一操作。++x 表示先将 x 值加 1 后再用，x++表示先使用 x 值，用后再加 1，所以本题中 ABC 选项都会输出 x+1 的值。答案选择 D 选项。

109．有以下程序：

```
#include<stdio.h>
main()
{
   int x=010,y=10;
   printf("%d,%d\n",++x,y--);
}
```

程序运行后的输出结果是（　　）。

A. 10,9

B. 11,10

C. 010,9

D. 9,10

【答案】D

【解析】整型常量有 3 种表示方法，分别是十进制数表示法、八进制数表示法和十六进制数表示法，八进制整常量以 0 作为前缀。自增和自减运算符的两种用法：前置运算，运算符放在变量之前，规则是先使变量的值增（或减）1，然后以变化后表达式的值参与其他运算；后置运算，运算符放在变量之后，规则是变量先参与其他运算，然后再使变量的值增（或减）1。x=010，即十进制的 8，y=10，++x 先自加后取值，输出 9，y--先取值输出 10，再自减 y=9，答案选择 D 选项。

110．有以下程序：

```
#include<stdio.h>
main()
{
   int i,j,k,a=5,b=6;
   i=(a==b)?++a:--b;
   j=a++;
   k=b;
   printf("%d,%d,%d\n",i,j,k);
}
```

程序的运行结果是（　　）。

A. 7,6,5

B. 5,5,5

C. 7,5,5

D. 5,6,5

【答案】B

【解析】条件表达式 i=(a==b)?++a:--b;中先执行 a==b，值为假，根据三元运算符语法规则，执行--b，此时 b 为 5，赋给 i，i=5。j=a++，将 a=5 先赋给 j，再进行 a++，j=5，a=6，k=b=5，故最后输出的是 5,5,5。答案选择 B 选项。

111．有以下程序：

```
#include<stdio.h>
main()
{
   int a;
   scanf("%d",&a);
   if(a++<9)printf("%d\n",a);
   else printf("%d\n",a--);
}
```

程序运行时键盘输入 9<回车>，则输出的结果是（　　）。

A．10

B．11

C．9

D．8

【答案】A

【解析】“++”和“--”运算，当以前缀形式出现时，则先进行加一或减一操作，再取值，当以后缀形式出现时，则先取值，再进行加一或减一操作。判断条件中 if(a++<9)是先用后加，即 a 的值为 9，所以条件不成立，但是 a 已经进行了自增 1 操作，a 的值此时为 10。执行 else 语句时，因为打印 a--，是先用后减，所以先输出 10，然后 a 的值变为 9。答案选择 A 选项。

112．有以下程序：

```
#include <stdio.h>
main()
{
    int a=1,b=0;
    if(--a) b++;
    else if(a==0) b+=2;
    else b+=3;
    printf("%d\n",b);
}
```

程序运行后的输出结果是（　　）。

A．0

B．1

C．2

D．3

【答案】C

【解析】“++”和“--”运算，当以前缀形式出现时，则先进行加一或减一操作，再进行其他运算，当以后缀形式出现时，则先进行其他运算，再进行加一或减一操作。a 初始定义为 1，b 为 0，执行--a，a 的值变为 0，--a 的值也为 0，即 if 判断为假，执行 b+=2，输出 b 的值为 2。答案选择 C 选项。

113．有以下程序：

```
#include <stdio.h>
main()
{
    int a=7;
    while(a--);
    printf("%d\n", a);
}
```

程序运行后的输出结果是（　　）。

A．-1

B．0

C．1

D．7

【答案】A

【解析】“++”和“--”运算，当以前缀形式出现时，则先进行加一或减一操作，再取值，当以后缀形式出现时，则先取值，再进行加一或减一操作。程序中执行 a--，直到 while 判断为 0 时才跳出循环，执行下条语句，即 a 为 0 时再执行 a--，此时跳出 while 循环，最终输出的结果为-1。答案选择 A 选项。

114．有以下程序：

```
#include <stdio.h>
main()
{
   int a=1,b=1;
   while(a--)
      b--;
   printf("%d,%d\n", a,b);
}
```

程序的运行结果是（　　）。

A．-1,0

B．0,0

C．-1,-1

D．1,1

【答案】A

【解析】while 循环的判定条件为“a--”，即 a 先作为循环条件判定，然后再自减 1。第一次循环判定条件为真，执行完毕后 a=0，b=0，第二次循环判定条件为假，循环不成立，所以只执行判定表达式"a--"，所以最终 a=-1，b=0，答案选择 A 选项。

115．有以下程序

```
#include <stdio.h>
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（　　）。

A．2,3,1

B．2,3,2

C．2,3,3

D．2,2,1

【答案】A

【解析】注意区分，z=x++是先将 x 的值赋给 z，在令 x 自增；z=++x 是先将 x 自增，再将自增后的值赋给 z；而无论是++x 还是 x++，都会完成 x 自增的运算。对于表达式“z=x++,y++,++y;”，因为赋值运算符的优先级高于逗号运算符的优先级，所以可以将上式改成“(z=x++),(y++),(++y)”。然后从左向右先计算表达式 z=x++，后缀自增运算先进行其他运算，再执行自增运算，所以 z 的值为 1，x 的值为 2，再计算逗号表达式第二个表达式 y++，此时 y 的值为 1，y++的值为 2，最后计算第三个表达式++y，y 的值为 3。答案选择 A 选项。

116．以下程序段中的变量已定义为 int 类型，则

```
sum=pAd=5;
pAd=sum++,++pAd,pAd++;
printf("%d\n",pAd);
```

程序段的输出结果是（　　）。

A．6

B．4

C．5
D．7
【答案】D
【解析】自增和自减运算符的两种用法：前置运算，运算符放在变量之前，规则是先使变量的值增（或减）1，然后以变化后表达式的值参与其他运算；后置运算，运算符放在变量之后，规则是变量先参与其他运算，然后再使变量的值增（或减）1。执行 pAd=sum++，sum++是后置自增，执行完后，pAd=5，sum=6；++pAd 和 pAd++语句中没有其他运算，即效果相同，pAd 分别加 1，两句执行完后，pAd = 7。答案选择 D 选项。

117．有以下程序

```
#include<stdio.h>
main()
{
   int sum,pad,pAd;
   sum=pad=5;
   pAd=++sum,pAd++,++pad;
   printf("%d\n",pad);
}
```

程序的输出结果是（　　）。
A．8
B．5
C．7
D．6
【答案】D
【解析】C 语言中的标识符区分大小写，pad 只实现了一次自增操作，结果输出 6。答案选择 D 选项。

118．有以下程序

```
#include <stdio.h>
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（　　）。
A．32
B．24
C．5
D．40
【答案】A
【解析】自增运算符“++”分为前缀和后缀两种形式。两种形式的作用效果是一样的，都是使运算分量的值加 1，但是它们的表达式的值不一样，前缀形式表达式的值为运算分量加 1 之后的值，而后缀形式表达式的值为运算分量加 1 之前的值。题目中使用了一个 for 循环，循环变量 i 从 1 递增到 40。在循环体中有两条嵌套的 if 语句，首先判断 i++%5==0，即判断 i++的值（i 加 1 之前的值）是否能被 5 整除（判断后 i 被加 1），然后再判断++i 的值（i 加 1 之后的值）是否能被 8 整除（判断后 i 被加 1），若两个条件都满足了，就输出 i 的值，只有 i=30

时，满足 i++%5==0，此时 i=31，++i%8==0 成立，此时 i=32。答案选择 A 选项。

119．有如下程序：

```
#include<stdio.h>
main()
{
   int a =0,b=1;
   if(++a==b++)
      printf("T");
   else
      printf("F");
   printf("a=%d,b=%d\n",a,b);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．Ta=0,b=1

B．Fa=1,b=2

C．Ta=1,b=2

D．Fa=0,b=2

【答案】C

【解析】程序执行过程为：判断++a==b++是否成立，++a 前置运算先加 1，则运算符==左边表达式值为 1，a=1，b++后置运算先取值，则==右边表达式值我 1，b=2，即是判断 1==1，成立，输出 T，输出 a=1，b=2。答案选择 C 选项。

120．有如下程序：

```
#include<stdio.h>
main()
{
   int a =0,b=1;
   if(a++&&b++)
      printf("T");
   else
      printf("F");
   printf("a=%d,b=%d\n",a,b);
}
```

程序运行后的输出结果是（　　）。

A．Ta=1,b=2

B．Fa=0,b=2

C．Fa=1,b=1

D．Ta=0,b=1

【答案】C

【解析】程序中判断 if 条件是否成立，a++先取值为 0，则(a++&&b++)为 0，不且不计算 b++，而后 a 自增得 a=1，if 条件不成立，执行 else 下的语句，输出 F。最后执行输出语句；按照格式输出 a=1，b=1。答案选择 C 选项。

二、填空题

给定程序中，函数 fun 的功能是：找出 100 至 x（x≤999）之间各位上的数字之和为 15 的所有整数然后输出；符合条件的整数个数作为函数值返回。

例如，当 x 值为 500 时，各位数字之和为 15 的整数有：159、168、177、186、195、249、258、267、276、285、294、339、348、357、366、375、384、393、429、438、447、456、465、474、483、492，共有 26 个。

请在程序的下划线处填入正确的内容并把下划线删除，使程序得出正确的结果。

注意：源程序存放在考生文件夹下的 BLANK1.C 中。

不得增行或删行，也不得更改程序的结构！

试题程序如下：

```
#include <stdio.h>
int fun(int x)
{
   int n, s1, s2, s3, t;
   /**********found**********/
   n=①_____;
   t=100;
   /**********found**********/
   while(t<=②_____)
   {
      s1=t%10;
      s2=(t/10)%10;
      s3=t/100;
      if(s1+s2+s3==15)
      {
         printf("%d ",t);
         n++;
      }
      /**********found**********/
      ③_____;
   }
   return n;
}
void main()
{
   int x=-1;
   while(x>999||x<0)
   {
      printf("Please input(0<x<=999): ");
      scanf("%d",&x);
   }
   printf("\nThe result is: %d\n",fun(x));
}
```

【答案】

①0

②x

③t++

【解析】

填空 1：变量 n 用于存放符合条件的整数的个数，应赋初值为 0。

填空 2：根据题目要求，确定循环变量 t 的取值范围 t<=x。

填空 3：循环变量 t 自增 1 操作。

三、改错题

给定程序 MODI1.C 中函数 fun 的功能是计算“n!”。

例如，n 为 5，则输出 120.000000。

请改正程序中的错误，使程序能输出正确的结果。

注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构！

试题程序如下：

```
#include <stdio.h>
double fun(int n)
{
   double result=1.0;
   /*********found*********/
   if n==0
      return 1.0;
   while(n>1&&n<170)
      /*********found*********/
      result *=n--
   return result;
}
main()
{
   int n;
   printf("Input N:");
   scanf("%d",&n);
   printf("\n\n%d!=%lf\n\n",n,fun(n));
}
```

【答案】

（1）错误：if n==0

正确：if(n==0)

（2）错误：result *=n--

正确：result*=n--;或{result*=n;n--;}

【解析】

错误 1：这里是一个简单的格式错误，if 条件应该加括号。

错误 2：格式错误，C 语言中以分号表示一条语句的结束。故在 result*=n--后添加分号。

四、设计题

请根据以下各小题的要求设计 C 应用程序（包括界面和代码）。

请编写函数 fun()，该函数的功能是：计算并输出给定整数 n 的所有因子（不包括 1 和自身）之和。规定 n 的值不大于 1000。例如，在主函数中从键盘给 n 输入的值为 856，则输出为：sum=763。

注意：部分源程序给出如下。

请勿改动主函数 main()和其他函数中的任何内容，仅在 fun()函数的花括号中填入所编写的若干语句。

试题程序如下：

```
#include <stdio.h>
int fun(int n)
{

}
void main()
{
    int n,sum;
    printf("Input n: ");
    scanf("%d",&n);
    sum=fun(n);
    printf("sum=%d\n",sum);
}
```

答：

```
int fun(int n)
{
    int s=0,i;
    for(i=2;i<=n-1;i++)
        if(n%i==0)
            s+=i;
    return s;
}
```

【解析】本题的设计思路是：①遍历从 2 到 n-1 的所有整数；②用条件语句找出能被 n 整除的整数 i，并累加求和；③用 return 语句返回因子的和。

# 第 3 章　最简单的 C 程序设计——顺序结构

一、选择题

1．以下叙述中正确的是（　　）。

A．赋值语句是一种执行语句，必须放在函数的可执行部分

B．scanf 和 printf 是 C 语言提供的输入和输出语句

C．由 printf 输出的数据都隐含左对齐

D．由 printf 输出的数据的实际精度是由格式控制中的域宽和小数的域宽来完全决定的

【答案】A

【解析】A 项正确，赋值语句是一种可执行语句，应当出现在函数的可执行部分。但需要注意，不要把变量定义时的赋初值和赋值语句混为一谈。B 项错误，C 语言本身没有提供输入输出功能，scanf 和 printf 属于标准库函数；C 项错误，数据都隐含右对齐，如果想左对齐，可以在格式控制中的“%”和宽度之间加一个“-”号来实现；D 项错误，若给出的总宽度 n1 小于 n2 加上整数位数和小数点（e 或 E 格式还要加上指数的 5 位），则自动突破 n1 的限制。答案选择 A 选项。

2．以下叙述中正确的是（　　）。

A．在 scanf 函数的格式串中，必须有与输入项一一对应的格式转换说明符

B．只能在 printf 函数中指定输入数据的宽度，而不能在 scanf 函数中指定输入数据占的宽度

C．scanf 函数中的字符串是提示程序员的，输入数据时不必管它

D．复合语句也被称为语句块，它至少要包含两条语句

【答案】A

【解析】在 printf 和 scanf 函数中都可以指定数据的宽度，B 项错误；scanf 中的字符串在输入时可以使用其他非空字符，如逗号，但在输入时必须输入这些字符以保证匹配，C 项错误；复合语句可以由任意多条语句构成，也可以一条没有，D 项错误。答案选择 A 选项。

3．以下能正确输出字符 a 的语句是（　　）。

A．printf("%s","a");

B．printf("%s",'a');

C．printf("%c","a");

D．printf("%d",'a');

【答案】A

【解析】“格式控制串”用来指定每个输出项的输出格式，%s 对应字符串，%c 对应字符，%d 对应整型。双引号里面的内容为字符串"a"，单引号里面的内容为字符'a'。答案选择 A 选项。

4．以下不能输出字符 A 的语句是（　　）。（注：字符 A 的 ASCII 码值为 65，字符 a 的 ASCII 码值为 97。）

A．printf("%c\n",'a'-32);

B．printf("%d\n",'A');

C．printf("%c\n",65);

D．printf("%c\n",'B'-1);

【答案】B

【解析】A 项，字符'a'的 ASCII 码值减去 32 为'A'的 ASCII 码值，执行字符输出，即为'A'；B 项，执行整型输出，结果为 65；C 项，字符型输出'A'；D 项，字符型输出'A'。答案选择 B 选项。

5．设有定义：double x=2.12;，以下不能完整输出变量 x 值的语句是（　　）。

A．printf("x=%5.0f\n",x);

B．printf("x=%f\n",x);

C．printf("x=%1f\n",x);

D．printf("x=%0.5f\n",x);

【答案】A

【解析】printf 函数控制字符%f 输出 float 类型，%1f 输出 double 类型。格式控制%m.nf，表示数据输出总的

宽度为 m 位，其中小数部分占 n 位。当数据的小数位多于指定宽度 n 时，截去右边多余的小数，并对截去的第一位小数做四舍五入处理；而当数据的小数位少于指定宽度 n 时，在小数的右边补零；当 m 小于有效位数时，整数部分输出所有有效数字并且自动对齐，小数部分按照 n 指定位数输出。A 项按照 float 格式输出数据，宽度为 5 位，保留小数 0 位，输出为 2，不能完整输出 x。B 项按照 float 格式输出数据，输出为 2.120000。C 项按照 double 格式输出数据，输出为 2.120000。D 项按照 float 格式输出数据，保留小数位数为 5，输出为 2.12000。答案选择 A 选项。

6．有以下程序：

```
#include<stdio.h>
main()
{
   int k=10;
   printf("%4d,%o,%x\n",k,k,k);
}
```

程序的运行结果是（　　）。（u 代表一个空格）

A．10,12,a

B．uu10,012,a

C．010,12,a

D．uu10,12,a

【答案】D

【解析】%4d 表示输出占 4 个字符的十进制，故先输出 2 个空格，然后输出 10；%o 表示输出八进制，所以输出 10 的八进制为 12；%x 表示输出十六进制，即 a。答案选择 D 选项。

7．有以下程序：

```
#include <stdio.h>
main()
{
   int k=-17;
   printf("%d,%o,%x\n",k,1-k,1-k);
}
```

程序的运行结果是（　　）。

A．-17,22,12

B．-17,12,22

C．-17,-22,-l2

D．17,22,12

【答案】A

【解析】整型常量有 3 种表示方法，①十进制整常量，没有前缀，输出格式控制符为%d；②八进制整常量，以 0 作为前缀，输出格式控制符为%o；③十六进制整常量，以 0x 或 0x 作为前缀，输出格式控制符为%x。1-k=18，整型常量 18 用八进制表示为 22，十六进制表示为 12，答案选择 A 选项。

8．有以下程序段：

```
char ch;
int k;
ch='a';
k =12;
printf("%c,%d,",ch,ch,k);
printf("k=%d\n",k);
```

已知字符 a 的 ASCII 码十进制值为 97，则执行上述程序段后输出的结果是（　　）。

A．因变量类型与格式描述符的类型不匹配，输出无定值
B．输出项与格式描述符个数不符，输出为零值或不定值
C．a,97,12k=12
D．a,97,k=12
【答案】D
【解析】字符变量的值是该字符的 ASCII 码值，可以参与整型变量所允许的任何运算。“ch='a'”，%c 表示以字符格式输出 ch 的值，所以输出为 a；%d 表示以十进制代码的形式输出 ch 的值，为 97；k 没有对应的输出格式，不输出。在第二个语句中，首先输出“k=”，然后以十进制代码输出 k 的值，为 12。答案选择 D 选项。

9．有以下程序：

```
#include <stdio.h>
main()
{
   int a=1,b=0;
   printf("%d,",b=a+b);
   printf("%d\n",a=2*b);
}
```

程序运行后的输出结果是（　　）。
A．0,0
B．1,0
C．3,2
D．1,2
【答案】D
【解析】main 函数先为 a、b 赋值，然后做运算 a+b 结果赋值给 b，此时 b 为 1，并将 b 打印出来。接着做运算 2*b 结果为 2*1=2 赋值给 a，将 a 打印出来，所以最终的输出结果为 1,2。答案选择 D 选项。

10．程序段：

```
int x=12;
double y=3.141593;
printf("%d%8.6f",x,y);
```

输出结果是（　　）。
A．123.141593
B．12 3.141593
C．12，3.141593
D．123.1415930
【答案】A
【解析】输出的 x 与 y 间没有空格，“%8.6f”代表总共 8 位宽度，包括小数点，小数点后有 6 位小数。答案选择 A 选项。

11．有以下程序

```
#include <stdio.h>
main()
{
   int a=2,c=5;
   printf("a=%%d,b=%%d\n",a,c);
}
```

程序的输出结果是（　　）。
A．a=2,b=5

B. a=%2,b=%5

C. a=%d,b=%d

D. a=%%d,b=%%d

【答案】C

【解析】%在 C 语言中有两个作用，一是用作取余数运算符，另一个用作转义符。%%d 相当于将第二个%转义了，所以输出为%d。故答案选择 C 选项。

12. 有以下程序：

```
#include <stdio.h>
main()
{
   char a,b,c,d;
   scanf("%c%c",&a,&b);
   c=getchar();
   d=getchar();
   printf("%c%c%c%c\n",a,b,c,d);
}
```

当执行程序时，按下列方式输入数据（从第一列开始，<CR>代表回车，注意：回车是一个字符）

12<CR>

34<CR>

则输出结果是（　　）。

A. 1234

B. 12

C. 12<CR>3

D. 12<CR>34

【答案】C

【解析】scanf()函数的一般调用形式为：scanf（格式控制，输入地址列表）。其中，格式控制是用双引号括起来的字符串，包括格式字符和普通字符，格式是由“%”和格式字符组成。getchar()函数的功能是从标准输入设备上读入一个字符。根据程序中的格式控制可知，接收输入时分别把 1 赋给了 a，2 赋给了 b，然后 getchar()函数提取一个换行符赋给 c，再提取一个字符 3 赋给了 d。所以程序的输出结果为：12<CR>3。答案选择 C 选项。

13. 设有：char s[5],c;，则调用函数 scanf 能正确给 s 和 c 读入数据的是（　　）。

A. scanf("%s%c",s,c);

B. scanf("%d%c",&s,&c);

C. scanf("%d%c",s,&c);

D. scanf("%s%c",s,&c);

【答案】D

【解析】s[5]是一个字符数组，也可以理解为字符串，格式控制为%s，c 为字符，格式控制为%c。scanf 输入时参数是地址，数组名就是地址，所以给 s 读入数据参数就是 s 首地址，而字符 c 的参数需要取 c 的地址，即&c。答案选择 D 选项。

14. 设有定义：

int a,b;float x,y;

则以下选项中对语句所作的注释叙述错误的是（　　）。

A. scanf("%d%d%f",&a,&b);/*多余的格式符%f 完全不起作用*/

B. scanf("%d%d",&a,&b,&x);/*多余的输入项不能获得输入数据*/

C. scanf("%d%f%d",&a,&b,&x);/*输入项与格式符类型不匹配，变量 b 和 x 得不到正确的输入数据*/

D. scanf("Input:%d%d",&a,&b);/*格式串中允许加入格式符以外的字符串*/

【答案】A

【解析】A 项中%f 是起作用的，程序从键盘正确的读入前两个数据并且保存在指定的地址，读入第三个数据后，将其放入缓冲区，然后寻找应该存放的地址，因为没有找到，程序会发生错误而中断，注释错误。B 项由于 scanf 只接收两个数据，所以变量 x 得不到赋值，注释正确。C 项 scanf 会按照 float 类型读取输入的第二个数据并且保存为 int 类型，由于两种类型的存储形式与所占内存单元均不同，b 得不到正确的赋值，注释正确。D 项在使用 scanf 函数时，如果除了格式说明字符和附加格式字符外，还有其他字符，则在输入数据时要求按一一对应的位置原样输入这些字符，注释正确。答案选择 A 选项。

15．若有定义和语句：

int a,b;

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

以下选项中的输入数据，不能把值 3 赋给变量 a、值 5 赋给变量 b 的是（　　）（说明：符号 u 表示空格）。

A．3,5,

B．3,5,4

C．3,u5

D．3,5

【答案】C

【解析】输入数据时，必须与格式控制中的格式一样，需要在数据后面紧跟一个逗号，否则不能正确读入数据，C 项，3 后面有一个多余的空格，不符合给定的格式。答案选择 C 选项。

16．若有定义：

int a,b;

通过语句：

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

能把整数 3 赋给变量 a，5 赋给变量 b 的输入数据是（　　）。

A．3 5

B．3,5

C．3;5

D．35

【答案】C

【解析】在采用 scanf 这个函数输入数据时，要严格遵守其输入的规则定义。本题中，由表达式"%d;%d"可知，函数定义的规则是在两个整数之间加分号输入。答案选择 C 选项。

17．设有以下语句

char ch1,ch2;

scanf("%c%c",&ch1,&ch2);

若要为变量 ch1 和 ch2 分别输入字符 A 和 B,正确的输入形式应该是（　　）。

A．A 和 B 之间用逗号间隔

B．A 和 B 之间不能有任何间隔符

C．A 和 B 之间可以用回车间隔

D．A 和 B 之间用空格间隔

【答案】B

【解析】在 scanf 输入整数或实数这类数值型数据时，输入的数据之间必须用空格、回车符、制表符（Tab 键）等间隔符隔开，间隔符个数不限。但在输入字符型时，要求输入数据时按照一一对应的位置原样输入这些字符，即不能加逗号、回车和空格，因为这些也算是字符。%c 比较特殊，它是输入单个字符，此处输入格式为“%c%c”，所以输入字符 A 和 B 时，不能间隔。答案选择 B 选项。

18．有以下程序段：

char c1,c2,c3;

scanf("%c%c%c",&c1,&c2,&c3);

若要给 c1、c2、c3 分别输入字母 A、B、C，则以下对输入形式的描述正确的是（　　）。

A．字母 A、B、C 之间可以用空格分隔

B．字母 A、B、C 之间不能有分隔符

C．字母 A、B、C 之间可以用回车符分隔

D．字母 A、B、C 之间可以用 Tab 键分隔

【答案】B

【解析】在使用 scanf 函数时要注意，在用“%c”格式输入字符时，分隔符（空格符、制表符（Tab 键）、回车符）和转义字符都将作为有效字符进行输入。题目中要求以字符形式输入三个数据，空格、回车符、Tab 键均会被视为有效字符赋给对应变量，所以字母 A、B、C 之间不能有分隔符，答案选择 B 选项。

19．若有定义：

```
int a;
float b;
double c;
```

程序运行时输入：a=1,b=2,c=3<回车>，能把值 1 输入给变量 a、值 2 输入给变量 b、值 3 输入给变量 c 的输入语句是（　　）。

A．scanf("a=%d,b=%f,c=%lf",&a,&b,&c);

B．scanf("%d%f%lf",&a,&b,&c);

C．scanf("a=%d,b=%lf,c=%lf",&a,&b,&c);

D．scanf("a=%d,b=%f,c=%f",&a,&b,&c);

【答案】A

【解析】格式字符 d，输入十进制整数；格式字符 f，输入浮点数；格式字符 lf，输入双精度浮点数。除了格式说明字符和附加格式字符外，如果还有其他字符，则在输入数据时要求按照这些字符在一一对应的位置原样输入。程序运行时输入：a=1，b=2，c=3，则 scanf 函数格式控制串应该是“a=%d,b=%f,c=%lf”。答案选择 A 选项。

20．若有定义语句

```
double x,y,*px,*py;
```

执行了 px=&x;py=&y;之后，输入语句正确的是（　　）。

A．scanf("%f%f",x,y);

B．scanf("%f%f"&x,&y);

C．scanf("%1f%1e",px,py);

D．scanf("%1f%1f",x,y);

【答案】C

【解析】输入函数 scanf 的标准格式是：scanf(格式控制,地址列表)，AD 两项中地址列表格式不正确，应为&x 和&y。格式控制和地址列表间应该用逗号隔开，B 项也错误。%f 用来输入 float 类型变量，%1f 用来输入 double 类型变量，%1e 表示用科学计数法输入 double。答案选择 C 选项。

21．若有定义语句

```
int a,b,c,*p=&c;
```

接着执行以下选项中的语句，则能正确执行的语句是（　　）。

A．scanf("%d%d%d",a,b,c);

B．scanf("%d",p);

C．scanf("%d",a,b,c);

D．scanf("%d",&p);

【答案】B

【解析】scanf 中地址列表需要取地址，即&a 形式，故 AC 项错误；p 为指针类型，本身就是 c 的地址，因此不用再取地址。答案选择 B 选项。

22．有以下程序段：

```
char name[20];
```

int num;

scanf("name=%s num=%d",name,&num);

当执行上述程序段，并从键盘输入：name=Lili num=1001<回车>后，name 的值为（　　）。

A．Lili

B．name=Lili

C．Lili num=

D．name=Lili num=1001

【答案】A

【解析】在 C 语言中输入多个字符串时，系统会把空格字符作为输入的字符串之间的分隔符。本题中，当从键盘输入 name=Lili num=1001 时，Lili 赋值给 name，1001 赋值给 num。答案选择 A 选项。

23．设有定义：

double x[10],*p=x;

以下能给数组 x 下标为 6 的元素读入数据的正确语句是（　　）。

A．scanf("%f",&x[6]);

B．scanf("%1f",*(x+6));

C．scanf("%1f",p+6);

D．scanf("%1f",p[6]);

【答案】C

【解析】scanf 函数的一般形式为：scanf(格式控制字符串,地址列表);，其中的地址列表应当是存放输入数据变量的地址。A 项，“%f”格式符对应的是 float 类型的变量；BD 两项，*(x+6)和 p[6]都表示下标为 6 的元素的值而非其地址；C 项，p+6 表示数组 x 下标为 6 的元素的地址。答案选择 C 选项。

24．有以下程序：

```
main()
{
   int a1,a2;
   char c1,c2;
   scanf("%d%c%d%c",&a1,&c1,&a2,&c2);
   printf("%d,%c,%d,%c",a1,c1,a2,c2);
}
```

若想通过键盘输入，使得 a1 的值为 12，a2 的值为 34，c1 的值为 a，c2 的值为 b，程序输出结果是：12，a，34，b，则正确的输入格式是（以下_代表空格，<CR>代表回车）（　　）。

A．12a34b<CR>

B．12_a_34_b<CR>

C．12,a,34,b<CR>

D．12_a34_b<CR>

【答案】A

【解析】在输入字符型时，要求输入数据时按照一一对应的位置原样输入这些字符，即不能加逗号、回车和空格，因为这些也算是字符。答案选择 A 选项。

25．若变量已正确定义为 int 型，要通过语句 scanf("%d,%d,%d",&a,&b,&c);给 a 赋值 1、给 b 赋值 2、给 C 赋值 3，以下输入形式中错误的是（u 代表一个空格符）（　　）。

A．uuu1,2,3<回车>

B．1u2u3<回车>

C．1,uuu2,uuu3<回车>

D．1,2,3<回车>

【答案】B

【解析】在输入整数或实数这类数值型数据时，输入的数据之间必须用空格、回车符、制表符（Tab 键）等

间隔符隔开，间隔符个数不限。在题目中，scanf 函数使用通配符逗号，则在输入数据时也要使用通配符逗号，且逗号要紧跟着数据后面。B 项，没有输入非格式符“,”。答案选择 B 选项。

26．设变量均已正确定义，若要通过“scanf("%d%c%d%c",&a1,&c1,&a2,&c2);”语句为变量 a1 和 a2 赋数值 10 和 20，为变量 c1 和 c2 赋字符 X 和 Y。以下所示的输入形式中正确的是（注：u 代表空格字符，<CR>代表回车）（　　）。

A．10X<CR>20Y<CR>

B．10uX20uY<CR>

C．10uX<CR>20uY<CR>

D．10uXu20uY<CR>

【答案】A

【解析】因为空格和回车也是字符变量，所以在输入 X 和 Y 时，它们的前面不能有空格和回车，否则将取到空格或者回车，而非 X 和 Y；int 类型不会取到空格和回车，所以 10 和 20 前面可以有空格和回车。答案选择 A 选项。

27．设变量均已正确定义，若要通过“scanf("%d%c%d%c",&a1,&c1,&a2,&c2);”语句为变量 a1 和 a2 赋数值 10 和 20，为变量 c1 和 c2 赋字符 X 和 Y。以下所示的输入形式中正确的是（　　）（u 代表空格字符）。

A．10X20Y

B．10uX20uY

C．10uX20uY

D．10uXu20uY

【答案】A

【解析】scanf 函数是 C 语言提供的标准输入函数，作用是接收在终端设备（或系统隐含指定的输入设备）上输入的数据。scanf 函数的一般形式为：scanft(格式控制,输入项表);。本题中的格式控制字符串是“%d%c%d%c”，其中%d 表示要输入的是整数；%c 则表示输入的是字符，且各控制符之间无任何分隔字符，故要求输入的数据之间也不能分开。而 BCD 三项在 10 和 X 之间均插入空格，均不正确。答案选择 A 选项。

28．若有定义

int a;

float b;

double c;

程序运行时输入：

3 4 5<回车>

能把值 3 输入给变量 a、4 输入给变量 b、5 输入给变量 C 的语句是（　　）。

A．scanf("%lf%lf%lf",&a,&b,&c);

B．scanf("%d%lf%lf",&a,&b,&c);

C．scanf("%d%f%f",&a,&b,&c);

D．scanf("%d%f%lf",&a,&b,&c);

【答案】D

【解析】%d 输入带符号的十进制整型数，%f 以带小数点的数学形式或指数形式输入浮点数（单精度数用%f，双精度数用%lf）。答案选择 D 选项。

29．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）

123<回车>

45678<回车>

则输出结果是（　　）。

A．1245

B．1256

C．1278

D．1267

【答案】A

【解析】scanf 是格式化输入函数；getehar 函数从键盘缓冲区读入下一个字符；putchar 输出一个字符；printf 函数是格式化输出函数。在题目中，程序执行到 scanf 函数时，会暂停等待用户输入 4 个字符，按题意输入 123<回车>后，字符'1'～'3'被分别赋值到 c1～c3 中，而 c4 会得到一个换行符'\n'。然后执行第 1 个 getchar 函数，由于前面的 scanf 函数读完了缓冲区中的所有字符，所以此时程序又会暂停等待用户输入，按题意输入 45678<回车>后，缓冲区第一个字符'4'赋值 c5，第二个字符'5'赋值给 c6。答案选择 A 选项。

30．有以下程序段

```
#include<stdio.h>
main()
{
   int j;
   float y;
   char name[50];
   scanf("%2d%f%s",&j,&y,name);
}
```

当执行上述程序段，从键盘上输入 55566 7777123 后，y 的值为（　　）。

A．566.0

B．55566.0

C．7777.0

D．566777.0

【答案】A

【解析】本题考查的是格式输入函数，即按用户指定的格式从键盘上把数据输入到指定的变量之中，其中的格式命令可以说明最大域宽。在百分号%与格式码之间的整数用于限制从对应域读入的最大字符数。因此 j 的值为 55，y 的值为 566.0，字符数组 name 的值为 7777123。答案选择 A 选项。

31．若有定义：

int a;float b;

执行 scanf("%2d%f",&a,&b);语句时，若从键盘输入 876 543.0，则 a 和 b 的值分别是（　　）。

A．87 和 6.0
B．876 和 543.0
C．87 和 543.0
D．76 和 543.0
【答案】A
【解析】scanf 函数的一般形式为：scanf(格式控制,地址表列);。其中，“格式控制”是用双引号括起来的字符串，也称“转换控制字符串”，它包括两种信息：①格式说明，由“%”和格式字符组成；②普通字符，即需要原样输入的字符。“地址表列”是需要接收输入数据的一系列变量的地址。本题中的“格式控制”是“%2d%f”，其中%2d 的意思是要输入一个整数，但该整数最宽只占 2 个字符，而%f 是要输入一个浮点数。而题目要求输入的是 876 和 543.0，所以 scanf 函数将 87 赋给 a，6 赋给 b。答案选择 A 选项。

32．有以下结构体说明、变量定义和赋值语句

```
struct STD
{
   char name[10];
   int age;
   char sex;
}s[5],*ps;
ps = &s[0];
```

则以下 scanf 函数调用语句有错误的是（　　）。
A．scanf("%s",s[0].name);
B．scanf("%d",&s[0].age);
C．scanf("%c",&(ps->sex));
D．scanf("%d",ps->age);
【答案】D
【解析】A 项正确，s[0].name 是取 s[0]中的 name 成员，name 是 char 类型数组，自身就是首元素地址；B 项正确，[]和.操作符优先级高于&，等价于&(s[0].age)；C 项正确，ps->sex 是取 s[0]的 sex 成员；D 项错误，ps->age 是取 s[0]中的 age 成员，scanf 函数中需要传入变量的地址。答案选择 D 选项。

33．下列叙述中正确的是（　　）。
A．可以用关系运算符比较字符串的大小
B．空字符串不占用内存，其内存空间大小是 0
C．两个连续的单引号是合法的字符常量
D．两个连续的双引号是合法的字符串常量
【答案】D
【解析】A 项错误，关系运算符不能比较字符串大小，可以用函数库中的字符串比较函数来比较字符串的大小；B 项错误，空字符串占用一个字节的内存；C 项错误，字符常量是用单引号把一个字符括起来；D 项正确，两个连续的双引号是一个字符串常量，称为空串。答案选择 D 选项。

34．下列叙述中正确的是（　　）。
A．两个连续的双引号（""）是合法的字符串常量
B．两个连续的单引号（''）是合法的字符常量
C．可以对字符串进行关系运算
D．空字符串不占用内存，其内存空间大小是 0
【答案】A
【解析】A 项正确，两个连续的双引号""也是一个字符串常量，称为“空串”。B 项错误，两个连续的单引号表示空字符，空字符它不占内存，故其不能称之为字符常量，常量是要有地址的；C 项错误，C 语言中不能对字符串直接使用关系运算符进行运算，但是可以使用比较函数 strcmp 进行比较；D 项错误，空串要占一个字节的存储空间来存放'\0'。答案选择 A 选项。

35．若有定义语句

char c='\101';

则变量 c 在内存中占（　　）。

A．2 个字节

B．1 个字节

C．3 个字节

D．4 个字节

【答案】B

【解析】字符型常量在内存中占一个字节。'\101'表示 8 进制数 101，即 10 进制中的 65，而'65'=A，所以，c 表示的字符是 A，一个字符在内存中占一个字节，答案选择 B 选项。

36．以下选项中非法的字符常量是（　　）。

A．'\101'

B．'\65'

C．'\xff'

D．'\019'

【答案】D

【解析】字符常量有两种表示方法：一种是用该字符的图形符号；二是用字符的 ASCII 码表示，即用反斜符（\）开头，后跟字符的 ASCII 码，这种方法也称为转义序列表示法，具体方法有两种，一种是用字符的八进制 ASCII 码，表示为\odd，这里 ddd 是八进制值（o 可以省略）。另一种使用字符的十六进制 ASCII 码值，表示为\0xhh 或 0Xhh，这里 hh 是两位十六进制值。D 项，'\019'使用的是八进制表示，八进制表示时，每一位的可用数值范围是 0～7 不应该出现 9，所以错误。答案选择 D 选项。

37．以下选项中不属于字符常量的是（　　）。

A．'C'

B．"C"

C．'\Xcc

D．'\072'

【答案】B

【解析】B 项，C 语言中用双引号表示字符串，在分配存储空间时需要包含"\0"作为结束标志。CD 两项，分别表示十六进制、八进制格式 ASCII 码值对应的字符常量。答案选择 B 选项。

38．以下不合法的字符常量是（　　）。

A．'\018'

B．'\"'

C．'\\'

D．'\xcc'

【答案】A

【解析】在 C 语言中，'\'表示转义符，可以对特殊符号进行转义，此时特殊符号不再有其他含义，仅仅代表普通的字符，\后面的字符取值范围应该在 ASCII 码表范围。例如'\"'表示对双引号的转义，'\\'表示对转义符的转义，'\n'表示换行，'\t'表示 Tab 键。A 项，'\0dd'中 dd 是八进制数，它表示八进制 ASCII 码对应的字符，八进制表示中 8 不是合法的八进制数字。B 项，表示双引号这个特殊字符；C 项，表示转义符这个特殊字符；D 项，'\xhh'表示两位十六进制 ASCII 码对应的字符，十六进制数 cc 即十进制的 192，'\xcc'表示 ASCII 码为 192 对应的字符。答案选择 A 选项。

39．以下选项中非法的 C 语言字符常量是（　　）。

A．'aa'

B．'\b'

C．'\007'

D．'\xaa'

【答案】A

【解析】A 项，'aa'表示字符串，应该用双引号括起来。B 项，转义符'\b'表示后退一格；C 项，'\ddd'表示八进制 ACSII 码对应的字符；D 项，'\xhh'表示两位的十六进制 ASCII 码对应的字符。答案选择 A 选项。

40．以下选项中非法的 C 语言字符常量是（　　）。

A．'\x9d'

B．'9'

C．'\x09'

D．'\09'

【答案】D

【解析】D 项，'\0dd'形式表示 dd 是两位八进制数，只能出现 0～7 之间的数字。A 项，'\xhh'形式表示 hh 是两位十六进制数；B 项，表示字符 9；C 项，表示十六进制数 09。答案选择 D 选项。

41．以下合法的转义字符是（　　）。

A．'\0X41'

B．'\0x41'

C．'\X41'

D．'\x41'

【答案】D

【解析】转义字符以反斜杠'\'开头，后面跟一个字符或一个八进制或十六进制数表示。十六进制转义字符是由反斜杠'\'和字母 x 及随后的 1～2 个十六进制数字构成的字符序列。答案选择 D 选项。

42．以下不是合法 C 语言转义字符的是（　　）。

A．'\c'

B．'\a'

C．'\b'

D．'\r'

【答案】A

【解析】C 语言中，'\a'表示响铃，'\b'表示退格，'\r'表示回车不换行，答案选择 A 选项。

43．有以下定义语句，编译时会出现编译错误的是（　　）。

A．char a='a';

B．char a='\n';

C．char a='aa';

D．char a='\x2d';

【答案】C

【解析】本题中 a 为一个字符型变量，只能为其赋值一个字符常量，A 项编译可以通过。C 项中'aa'不是字符常量，而是一个字符串，所以会编译错误。BD 两项为转义字符，编译可以通过。答案选择 C 选项。

44．已知字母 A 的 ASCII 码值为 65，若变量 kk 为 char 型，以下不能正确判断出 kk 中的值为大写字母的表达式是（　　）。

A．kk>='A'&&kk<='Z'

B．!(kk>='A'||kk<='Z')

C．(kk+32)>='a'&&(kk+32)<='z'

D．isalpha(kk)&&(kk<91)

【答案】B

【解析】B 项，表达式等价为 kk<'A'&&kk>'Z'，无法判断 kk 中的值是否为大写字母，逻辑错误。大写字母的 ASCII 码值是 65～90，小写字母的 ASCII 码值是 97～122，isalpha(kk)是判断 kk 是否是字母的函数，是字母并且字母 ASCII 值小于 91 的一定为大写字母。答案选择 B 选项。

45．已知大写字母 A 的 ASCII 码是 65，小写字母 a 的 ASCII 码是 97。以下不能将变量 c 中的大写字母转换为对应小写字母的语句是（　　）。

A．c=('A'+c)%26-'a'

B．c=c+32

C．c=c-'A'+'a'

D．c=(c-'A')%26+'a'

【答案】A

【解析】A 项，模 26 运算后得到 0~25 之间的数，再与'a'相减结果是负数；B 项，ASCII 码表中，同一字母的小写码比大写码数值上大 32；C 项，'a'-'A'=32；D 项，c-'A'取值在 0～26，等价于 C 项。答案选择 A 选项。

46．有以下程序（字母 A 的 ASCII 代码为 65）：

```
#include <stdio.h>
main()
{
   char c1='A', c2='Y';
   printf("%d,%d\n",c1,c2);
}
```

程序运行后的输出结果是（　　）。

A．输出格式不合法，输出出错信息

B．A,Y

C．65,90

D．65,89

【答案】D

【解析】字符可以用整型来输出，输出的是对应的 ASCII 值。'A'的 ASCII 值为 65，'Y'的 ASCII 值为 89，所以输出结果为 65,89。答案选择 D 选项。

47．有以下程序：

```
#include<stdio.h>
main()
{
   char ch='Z';
   ch=(ch-'A'+1)%26+'A';
   putchar(ch);
}
```

程序的运行结果是（　　）。

A．Z

B．Y

C．B

D．A

【答案】D

【解析】'Z'的 ASCII 码是 90，'A'的 ASCII 码是 65，所以 ch=(ch-'A'+1)%26+'A'=26%26+65=65，输出 65 对应的字符'A'。答案选择 D 选项。

48．若有以下程序

```
#include <stdio.h>
main()
{
   char c1, c2;
   c1='C'+'8'-'3';
   c2='9'-'0';
   printf("%c %d\n",c1,c2);
}
```

则程序的输出结果是（　　）。

A. H 9

B. 表达式不合法输出无定值

C. F '9'

D. H '9'

【答案】A

【解析】当字符参与数学运算时，替换成其在 ACSII 码表中对应的数值，则 c1='C'+'8'-'3'=67+56-51=72='H'，c2='9'-'0'=57-48=9；在 printf 函数中，c1 以字符的形式输出，得到结果 H，c2 以整数的形式输出，得到 ASCII 码值 9。答案选择 A 选项。

49．有如下程序：

```
#include <stdio.h>
main()
{
   if('\0'== 0)putchar('X');
   if('0'== 0)putchar('Y');
   if('a'>'b')putchar('Z');
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A. X

B. XYZ

C. YZ

D. Y

【答案】A

【解析】考查 ASCII 字符，字符'\0'、'0'、'a'、'b'的 ASCII 码值分别为 0、48、97、98。putchar 函数是向标准输出设备上输出一个字符。程序执行过程为：判断'\0'==0 成立，输出 X；判断'0'==0 不成立，不输出；判断'a'>'b'不成立，不输出。答案选择 A 选项。

50．有如下程序：

```
#include <stdio.h>
main()
{
   char ch='A';
   while(ch<'D')
   {
      printf("%d",ch-'A');
      ch++;
   }
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A. ABC

B. 012

C. abc

D. 123

【答案】B

【解析】while 循环语句执行时，首先判断表达式，成立（非 0）则执行循环体，不成立（0）则退出循环。字符之间做加减法，是用其 ASCII 码进行加减，所以程序执行过程为：定义字符变量 ch='A'，判断'A'<'D'成立，'A'-'A'=0，输出 0，ch++后，值为'B'；判断'B'<'D'成立，'B'-'A'=1，输出 1，ch++后，值为'C'；判断'C'<'D'成立，'C'-'A'=2，输出 2，ch++后，值为'D'；判断'D'<'D'不成立，退出循环。程序运行后的输出结果是 012，答案选择 B 选项。

51．有如下程序：

```
#include <stdio.h>
main()
{
   char ch='M';
   while(ch!='K')
   {
      ch--;
      putchar(ch);
   }
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A. MN

B. LK

C. OP

D. MM

【答案】B

【解析】程序执行过程为：定义字符变量 ch='M'，判断 ch!='K'成立，ch--后值为'L'，输出 L；判断 ch!='K'成立，ch--后值为'K'，输出 K；判断 ch!='K'不成立，退出循环。程序运行后的输出结果为 LK。答案选择 B 选项。

52．有以下程序：

```
#include <stdio.h>
void fun(char *c)
{
```

```
    while(*c)
    {
      if(*c>='a'&&*c<='z')*c=*c-('a'-'A');
      c++;
    }
}
main()
{
    char s[81];
    gets(s);
    fun(s);
    puts(s);
}
```

当执行程序时从键盘上输入 Hello Beijing<回车>，则程序的输出结果是（　　）。

A. hello beijing

B. Hello Beijing

C. HELLO BEIJING

D. HELLO Beijing

【答案】C

【解析】函数 fun 的功能是把字符串中的小写字母全部转换成大写字母。答案选择 C 选项。

53. 有以下程序：

```
#include <stdio.h>
main()
{
    char b,c;
    int i;
    b='a';
    c ='A';
    for(i=0;i<6;i++)
    {
      if(i%2) putchar(i+b);
      else putchar(i+c);
    }
    printf("\n");
}
```

程序运行后的输出结果是（　　）。

A. ABCDEF

B. AbCdEf

C. aBcDeF

D. abcdef

【答案】B

【解析】本题中，当 i 为偶数时，即 0、2、4，执行 putchar(i+c)会依次输出 ACE；当 i 为奇数时，即 1、3、5，执行 putchar(i+b)会依次输出 bdf，所以最终输出 AbCdEf。答案选择 B 选项。

54. 有以下程序：

```
#include <stdio.h>
void fun(char *s)
{
   while(*s)
   {
      if(*s%2==0)printf("%c",*s);
      s++;
   }
}
main()
{
   char a[]=("good");
   fun(a);
   printf("\n");
}
```

注意：字母 a 的 ASCII 码值为 97，程序运行后的输出结果是（　　）。

A．d

B．go

C．god

D．good

【答案】A

【解析】字符串"good"中 g 的 ASCII 码值为 103，o 的 ASCII 码值为 111，d 的 ASCII 码值为 100。在 fun 函数中，if(*s%2==0)语句选择 ASCII 码值为偶数的字母输出。在 good 中，只有 d 的 ASCII 码值为偶数，所以结果为 d。答案选择 A 选项。

55．以下不是 C 语言字符型或字符串常量的是（　　）。

A．"It’s"

B．"0"

C．'a=0'

D．'\010'

【答案】C

【解析】字符常量是用单引号把一个字符括起来，转义字符常量以一个反斜线开头后跟一个特定的字符或者对应的 ASCII 值表示。字符串常量是由双引号括起来的一串字符。C 项既不是字符型常量，也不是字符串常量。AB 两项，均是字符串常量；D 项，是字符型常量。答案选择 C 选项。

56．C 语言中 char 类型数据占字节数为（　　）。

A．3

B．4

C．1

D．2

【答案】C

【解析】char 为关键字。字符变量在内存中占一个字节，可以存放 ASCII 字符集中的任何字符。答案选择 C 选项。

57．若有说明语句：

```
char c='\72';
```

则变量 c 中存放的是（　　）。

A．2 个字符

B．1 个字符

C．3 个字符

D．说明语句不合法

【答案】B

【解析】用一对单引号括起来的单个字符为字符常量，以“\”开头的转义字符也是字符常量。“\”后可以为单个字符，也可以为八进制或十六进制数字，故变量 c 中存放的是一个字符。答案选择 B 选项。

58．已知字符 A 的 ASCII 代码值是 65，字符变量 c1 的值是 A，c2 的值是 D。则执行语句

printf("%d,%d",c1,c2-2);

的输出结果是（　　）。

A．A,68

B．65,66

C．A,B

D．65,68

【答案】B

【解析】在 C 语言中，字符型数据在内存中的存储形式是 ASCII 码值。当需要以整型格式输出字符时，输出的也是 ASCII 码值。字符'A'和'D'的 ASCII 码值分别为 65 和 68，c2-2 对应的 ASCII 码值为 66。答案选择 B 选项。

59．有以下程序（说明：字母 A 的 ASCII 码值是 65）：

```
#include <stdio.h>
void fun(char *s)
{
   while(*s)
   {
      if(*s%2)printf("%c",*s);
      s++;
   }
}
main()
{
   char a[]="BYTE";
   fun(a);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．BY

B．BT

C．YT

D．YE

【答案】D

【解析】函数只会输出 ASCII 码值为奇数的字母，执行 fun(a)时，依次取出 a[]中的字母，因为 B 的 ASCII 码值为 66，所以不会输出 B，字母 Y 的 ASCII 码值为 89，字母 T 的 ASCII 码值为 84，字母 E 的 ASCII 码值为 69，所以最后输出字母 YE。答案选择 D 选项。

60．有以下程序：

```
#include <stdio.h>
main()
{
   char c1,c2;
   c1='A'+'8'-'4';
   c2='A'+'8'-'5';
   printf("%c,%d\n",c1,c2);
}
```

已知字母 A 的 ASCII 码值为 65，程序运行后的输出结果是（　　）。

A. E,68

B. D,69

C. E,D

D. 输出无定值

【答案】A

【解析】C 语言中，字符型可以作为整型用，可以出现在任何需要整型的表达式中。该题中的 main 函数将字符 A 经过运算分别赋值给 c1、c2，然后将 c1 按字母格式输出，c2 按整型格式输出。答案选择 A 选项。

61. 有以下程序：

```
#include <stdio.h>
main()
{
   char ch='B';
   while(ch<'E')
   {
      printf("%d",ch-'A');
      ch++;
   }
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A. 123

B. ABC

C. abc

D. 012

【答案】A

【解析】当满足 ch<'E'，输出 ch-'A'，即字符的 ASCII 码相减进行输出，%d 表示输出十进制整数。循环开始 ch='B'，ch-'A'=1，以此类推，答案选择 A 选项。

62. 以下不能输出小写字母 a 的选项是（　　）。

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

B. printf("%c\n','A'+32);

C. putchar(97);

D. putchar('a');

【答案】A

【解析】printf 函数格式控制符%c 以字符形式输出数据。putchar 函数将括号中参数以字符形式输出。A 项“a”为字符串，不是单个字符，输出格式不正确，不能输出 a。B 项字符'a'的 ASCII 码为 97，字符'A'的 ASCII 码为 65，'A'+32 即为'a'，能输出 a。C、D 项 putchar 函数参数均为字符'a'，均可正确输出 a。答案选择 A 选项。

63．有以下程序：

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

执行时如输入为：abcdefg##<回车>，则输出结果是（　　）。

A．abcdefg

B．bcdefgh$

C．bcdefgh$$

D．bcdefgh

【答案】D

【解析】for 循环每次将函数 getchar()的输入值赋给变量 c，如果不等于'#'，则执行 putchar(++c)，即将当前变量 c 的 ASCII 码加 1 后，再输出改变后的变量 c 的值。当变量 c 的值等于'#'，则终止循环，所以输出应该是 bcdefgh。答案选择 D 选项。

64．有如下程序：

```
#include <stdio.h>
main()
{
   int i;
   for(i=0;i<5;i++)
      putchar('Z'-i);
}
```

程序运行后的输出结果是（　　）。

A．'X''Y''Z''W''V'

B．VWXYZ

C．ZYXWV

D．'ABCDE'

【答案】C

【解析】putchar 表示输出单个字符，执行'Z'-i 表示用'Z'对应的 ASCII 码减去 i，putchar('Z'-i)表示输出对应 ASCII 码值为('Z'-i)的字符，在 for 循环中，i=0，1，2，3，4 时，执行循环体，输出结果为 ZYXWV。答案选择 C 选项。

65．有如下程序：

```
#include <stdio.h>
main()
{
   int i;
   for (i=0;i<5;i++)
      putchar('9'-i);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．54321

B．98765

C．'9''8''7''6''5'

D．'43210'

【答案】B

【解析】本题执行过程为：i=0，输出字符 9，在 i<5 的情况下，依次输出字符 8、7、6、5。在 i=5 时，退出 for 循环。最后显示在命令窗口结果为 98765。答案选择 B 选项。

66．有如下程序：

```
#include <stdio.h>
main()
{
   char a='3',b='A';
   int i;
   for(i=0;i<6;i++)
   {
      if(i%3)putchar(a+i);
      else putchar(b+i);
   }
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．A45D78

B．ABC678

C．34CD78

D．34AB78

【答案】A

【解析】本题执行过程为：i=0，i%3=0，if 条件不成立执行 else 函数体，输出字符 A；i=1，i%3=1，if 条件成立输出字符 4；i=2，i%3=2，if 条件成立输出字符 5；i=3，if 条件不成立，输出字符 D；i=4 和 i=5 时，分别输出字符 7 与 8；i=6 退出 for 循环。所以程序运行后输出结果为 A45D78，答案选择 A 选项。

67．有以下程序：

```
#include <stdio.h>
main()
{
   char b,c;
   int i;
   b='a';
   c ='A';
   for(i=0;i<6;i++)
   {
      if(i%2) putchar(i+b);
      else putchar(i+c);
   }
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．ABCDEF

B．AbCdEf

C．aBcDeF

D．abcdef

【答案】B

【解析】本题中，当 i 为偶数时，即 0、2、4，执行 putchar(i+c)会依次输出 ACE；当 i 为奇数时，即 1、3、5，执行 putchar(i+b)会依次输出 bdf，所以最终输出 AbCdEf。答案选择 B 选项。

68．下面选项中关于位运算的叙述正确的是（　　）。

A．位运算符都需要两个操作数

B．位运算的对象只能是整型或字符型数据

C．左移运算的结果总是原操作数据 2 倍

D．右移运算时，高位总是补 0

【答案】B

【解析】B 项正确，C 语言中，位运算的对象只能是整型或字符型数据，不能是其他类型的数据。A 项错误，位运算符中取反操作符只需要一个操作符；C 项错误，左移时，若左端移出的部分不包含有效二进制数 1，则每左移一位，相当于移位对象乘以 2，如果端移出的部分包含有效二进制数 1，结论不成立；D 项错误，右移运算时，对于无符号整数和正整数，高位补 0；对于负整数，高位补 1。答案选择 B 选项。

69．下面关于位运算符的叙述，正确的是（　　）。

A．&表示“按位与”的运算

B．#表示“按位异或”的运算

C．||表示“按位或”的运算

D．~表示“按位异或”的运算

【答案】A

【解析】C 语言提供的六种位运算符：“~”按位求反，“<<”左移，“>>”右移，“&”按位与，“^”按位异或，“|”按位或。答案选择 A 选项。

70．以下不属于 C 语言位运算符的是（　　）。

A．!

B．|

C．^

D．~

【答案】A

【解析】C 语言常用的位运算符有：“~”按位求反，“<<”左移，“>>”右移，“&”按位与，“^”按位异或，“|”按位或。A 项“!”为逻辑运算符“非”。答案选择 A 选项。

71．以下选项中错误的是（　　）。

A．a&=b 与 a=a & b 等价

B．a^=b 与 a=a^b 等价

C．a|=b 与 a=a|b 等价

D．a!=b 与 a=a!b 等价

【答案】D

【解析】D 项，a!=b 表示 a 不等于 b 时，表达式结果为 1，否则为 0；而 a=a!b 是语法错误，“!”是非运算，且是单目运算符，要求只有一个操作数，故两者不等价。答案选择 D 选项。

72．变量 a 中的数据用二进制表示的形式是 01011101，变量 b 中的数据用二进制表示的形式是 11110000。若要求将 a 的高 4 位取反，低 4 位不变，所要执行的运算是（　　）。

A．a^b

B．a|b

C．a&b

D．a<<4

【答案】A

【解析】A 项，01011101 ^ 1111000 结果为 10101101，“^”表示异或运算，参与运算的两位二进制相异则运算结果为 1，相同则运算结果为 0；B 项，01011101 | 11110000 结果为 11111101，“|”表示或运算；C 项，01011101 & 11110000 结果为 01010000，“&”表示按位与；D 项，010101101 << 4 结果为 11010000，“<<”表示左移。答案选择 A 选项。

73．若有定义语句

int b=2;

则表达式(b<<2)/(3||b)的值是（　　）。

A．4

B．8

C．0

D．2

【答案】B

【解析】b=2，左移两位相当于乘以 4，2*4=8。3||b 的值为真即为 1，表达式的值是 8/1=8。答案选择 B 选项。

74．设有定义：

int a=64,b=8;

则表达式(a&b) || (a&&b)和(a|b) && (a||b)的值分别为（　　）。

A．1 和 1

B．1 和 0

C．0 和 1

D．0 和 0

【答案】A

【解析】逻辑或运算符“||”，计算左表达式值，如果为真，则不计算右表达式，而整个表达式为真，若左表达式为假，再计算右表达式然后做或运算。逻辑与运算符“&&”，计算左表达式值，如果为假则不计算右表达式，而整个表达式为假，若左表达式为真，再计算右表达式然后做与运算。a=64=10000008，b=8=1000B，a&b=0 为假，a&&b 为真，(a&b) || (a&&b)为真，值为 1。a|b=1001000B 为真，a||b 为真，则(a|b) && (a||b)为真，值为 1。答案选择 A 选项。

75．有以下程序：

```
#include <stdio.h>
main()
{
    int a=3;
    int b=3;
    printf("%d\n", a&b);
}
```

程序运行后的输出结果是（　　）。

A．6

B．1

C．0

D．3

【答案】D

【解析】“&”按位与运算符。a=3=11B，b=3=11B，按位与结果为 11B=3，答案选择 D 选项。

76．有以下程序：

```
#include <stdio.h>
main()
{
   int c,d;
   c=13&5;
   d=10&&5;
   printf("%d,%d\n",c,d);
}
```

程序的运行结果是（　　）。

A. 5,1

B. 1,1

C. 18,1

D. 13,1

【答案】A

【解析】“&”按位与运算，13=1101B，5=101B，执行语句 c=13&5；结果为 101B=5，即 c=5。“&&”逻辑与运算，10 与 5 均为真，结果为真，即 d=1。答案选择 A 选项。

77. 有以下程序

```
#include <stdio.h>
main()
{
   int a=12,c;
   c=(a<<2)<<1;
   printf("%d\n",c);
}
```

程序运行后的输出结果是（　　）。

A. 3

B. 50

C. 2

D. 96

【答案】D

【解析】执行 c=(a<<2)<<1;语句时，首先是 a 左移两位，等效于将 a 乘以 4，得到 48，然后将 48 左移一位，相当于乘以 2，所以结果为 c=96。答案选择 D 选项。

78. 有以下程序：

```
#include <stdio.h>
main()
{
   int a=2,b;
   b=a<<2;
   printf("%d\n",b);
}
```

程序运行后的输出结果是（　　）。

A. 2

B. 4

C. 6

D. 8

【答案】D

【解析】a 的初始值为 2，程序中将 a 左移 2 位的结果赋值给 b，即相当于十进制乘以 4，结果为 8。答案选择 D 选项。

79．有以下程序

```
#include <stdio.h>
void main()
{
   unsigned char a=8, c;
   c = a>>3;
   printf("%d\n",c);
}
```

程序运行后的输出结果是（　　）。

A．32

B．16

C．1

D．0

【答案】C

【解析】字符常量与其 ASCII 码值一一对应，在计算的时候，可以使用该字符的 ASCII 码参与运算，输出时会根据格式控制符输出对应的字符或者 ASCII 码。右移运算符“>>”的运算规则是把左边运算数的各二进制位全部右移 n 位，n 取决于“>>”右边的数值。字符 a 的 ASCII 码值为 8，对应二进制为 1000，所以 a>>3=1，输出的字符 c 的 ASCII 码值为 1，答案选择 C 选项。

80．有以下程序：

```
#include<stdio.h>
main()
{
   short c=124;
   c=c_____;
   printf("%d\n",c);
}
```

若要使程序的运行结果为 248，应在下画线处填入的是（　　）。

A．>>2

B．|248

C．&0248

D．<<1

【答案】D

【解析】要输入 248，是 124 的两倍，124 为 0111 1100，248 为 1111 1000，因此只要左移一位即可。答案选择 D 选项。

81．有以下程序

```
#include <stdio.h>
main()
{
   int a=2,b=2,c=2;
   printf("%d\n",a/b&c);
}
```

程序运行后的输出结果是（　　）。

A．0

B．1

C．2

D．3

【答案】A

【解析】运算符“/”的优先级高于“&”，a/b&c = 2/2&2 = 1&2 = 0000 0001 & 0000 0010 = 0000 0000 = 0。答案选择 A 选项。

82．有以下程序

```
#include <stdio.h>
main()
{
   int a=1,b=2,c=3,x;
   x=(a^b)&c;
   printf("%d\n",x);
}
```

程序的运行结果是（　　）。

A．2

B．1

C．3

D．0

【答案】C

【解析】a^b = 0000 0001b ^ 0000 0010b = 0000 0011b = 3，3&c = 0000 0011b & 0000 0011b = 3。答案选择 C 选项。

83．有以下程序：

```
#include <stdio.h>
main()
{
   int c,d;
   c=10^3;
   d=10+3;
   printf("%d,%d\n",c,d);
}
```

程序运行后的输出结果是（　　）。

A．103,13

B．13,13

C．10,13

D．9,13

【答案】D

【解析】C 语言中“^”的意思是按位异或，10^3 表示先把 10 和 3 换算成二进制再进行异或，即 1010 ^ 0011 = 1001b，转成十进制为 9。答案选择 D 选项。

84．若有以下程序段：

```
int r=8;
printf("%d\n",r>>1);
```

输出结果是（　　）。

A．16
B．8
C．4
D．2
【答案】C
【解析】C 语言中，“>>”右移运算符是将变量转换成二进制，然后右移相应位数，将移出的位信息舍去，并在高位补 0，将所得的结果再赋值给变量。本题十进制数 8 转换为二进制数为 00001000，右移一位得到 00000100，再转换成十进制数就是 4。所以答案选择 C 选项。

85．有以下程序：

```
#include<stdio.h>
main()
{
   int a=5,b=1,t;
   t=(a<<2)|b;
   printf("%d\n",t);
}
```

程序运行后的输出结果是（　　）。
A．21
B．11
C．6
D．1
【答案】A
【解析】方法有如下两种：①位运算最常规的方式是转换为二进制，然后再运算。5 的二进制是 101，在<<2 后为 10100，然后和 00001 进行或运算后等于 10101，其十进制为 21；②a<<2 即为 a*4，结果为 20，|1 相当于保证这个数是奇数，若不是则加 1，所以为 21。答案选择 A 选项。

86．有以下程序：

```
#include<stdio.h>
main()
{
   char a=4;
   printf("%d\n",a=a<<1);
}
```

程序运行的结果是（　　）。
A．40
B．16
C．8
D．4
【答案】C
【解析】移位运算符的优先级比赋值运算符优先级高，故语句“a=a<<1”先进行 a 的二进制移位运算，左移一位相当于原数乘以 2，a<<1 结果为 4*2=8，没有超过 char 类型的范围，将 8 赋值给 a，并输出 a 的值。答案选择 C 选项。

87．设有以下语句
int a=1,b=2,c;
c=a^(b<<2);
执行后，c 的值为（　　）。

A．9
B．7
C．8
D．6
【答案】A
【解析】b<<2 后得到 8，异或操作，两个操作数相同得 0，不同得 1。a 的最右 4 位是 0001，b 的最右 4 位是 1000，其余位全部是 0。0001 ^ 1000 = (1001)2 = 9，答案选择 A 选项。

88．有以下程序

```
#include <stdio.h>
main()
{
   unsigned char a=2,b=4,c=5,d;
   d=a|b;
   d&=c;
   printf("%d\n",d);
}
```

程序运行后的输出结果是（　　）。
A．4
B．3
C．5
D．6
【答案】A
【解析】d = a|b = 00000010 | 00000100 = 00000110，d = d&c = 00000110 & 00000101 = (00000100)2=4。答案选择 A 选项。

89．有以下程序

```
#include <stdio.h>
main()
{
   int i=1;
   i=i^i;
   printf("%d\n",i);
}
```

程序运行后的输出结果是（　　）。
A．7
B．-1
C．1
D．0
【答案】D
【解析】异或操作，两个操作数相同得 0，不同得 1，因此自身与自身异或，结果是 0。答案选择 D 选项。

90．若变量已正确定义，则以下语句的输出结果是（　　）。

```
s=32;
s^=32;
printf("%d",s);
```

A．0
B．-1

C．1

D．32

【答案】A

【解析】异或操作，两个操作数相同得 0，不同得 1，因此自身与自身异或，结果是 0。答案选择 A 选项。

91．有如下程序：

```
#include <stdio.h>
main()
{
   int a=8,b;
   b=(a>>2)%2;
   printf("%d,%d\n",a,b);
}
```

程序运行后的输出结果是（　　）。

A．8,0

B．4,0

C．4,1

D．8,1

【答案】A

【解析】“>>”右移运算符，按位右移运算规则是将一个操作数先转换成二进制数，然后将二进制数各位右移若干位，移出的低位舍弃，并在高位补位。若为无符号数，右移时左边高位移入 0。a = 8 = 01000B，执行 a>>2 后，a = 00010B = 2，b = 2%2 = 0，输出 a 和 b 分别为 8 和 0，答案选择 A 选项。

92．有如下程序：

```
#include <stdio.h>
main()
{
   int a=9,b;
   b=(a>>3)%4;
   printf("%d,%d\n",a,b);
}
```

程序运行后的输出结果是（　　）。

A．9,1

B．4,0

C．4,3

D．9,3

【答案】A

【解析】程序执行完成后，变量 a 的值不变，始终为 9，a 转化成二进制数 1001B，1001 >> 3 = 0001B = 1，1 再对 4 求余得到余数 1，赋给 b 值。答案选择 A 选项。

93．有以下程序：

```
#include <stdio.h>
main()
{
   int c,d;
   c=(13>>1)|1;
   d=(13>1)||1;
   printf("%d,%d\n",c,d);
}
```

程序运行后的输出结果是（　　）。

A．6,1

B．1,1

C．7,1

D．7,2

【答案】C

【解析】表达式 c = (13>>1)|1 中，>>是右移符号，数字 13 换算成二进制 1101 后右移 1 位为 0110，再和 1 进行按位或运算，0110 ^ 0001 = 0111b，结果为 7；表达式 d = (13>1)||1 中，首先判断 13 是否大于 1，此时条件成立，左边表达式值为 1 后，则不再计算右边表达式，d=1，答案选择 C 选项。

94．有以下程序：

```
#include <stdio.h>
main()
{
   char c='A';
   int x=36,b;
   b=(x>>2)&&(c<'a');
   printf("%d\n",b);
}
```

程序运行后的输出结果是（　　）。

A．1

B．0

C．2

D．4

【答案】A

【解析】“>>”右移运算符每次右移一位等价于除以 2，x=36，执行 x>>2 后，x=9；“&&”逻辑与运算，先计算左表达式真假，若为假，不计算右表达式，整个逻辑表达式为假，若左表达式为真，再计算右表达式，之后做逻辑与运算；x>>2 为真，A 的 ASCII 码为 65，a 的 ASCII 码为 97，c<'a'为真，整个逻辑表达式(x>>2)&&(c<'a')为真，得 b=1。答案选择 A 选项。

95．有以下程序：

```
#include <stdio.h>
main()
{
   int i,array[6] = {1,5,0,4};
   for(i=0; i<5;i++)
      printf("%d,",array[i]&4);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．1,2,1,2,0,

B．1,5,0,4,0,

C．1,5,5,4,0,

D．0,4,0,4,0,

【答案】D

【解析】本函数的功能是将数组中所有元素与 4 进行与运算，首先将元素转换为二进制，然后和二进制数 100(十进制 4)进行与运算，第一个元素为 001&100 即为 000。也就是 0，同理可得答案分别为 0、4、0、4、0，注意第五位按照 0 进行运算。答案选择 D 选项。

96．有以下程序：

```
#include <stdio.h>
main()
{
   int i, array[5]={3,5,10,4};
   for(i=0;i<5;i++)printf("%d,",array[i] &3);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．3,1,2,0,0,

B．3,5,10,4,0,

C．3,3,3,3,0,

D．3,2,2,2,0,

【答案】A

【解析】在对数组进行初始化时，如果在声明数组时给出了长度，但没有给所有的元素赋予初始值，那么 C 语言将自动对余下的元素赋初值 0，则 array={3,5,10,4,0}。按位与运算“&”，当参加运算的两个二进制数的对应位都为 1，则该位的结果为 1，否则为 0。将数组元素与 3 按位与，即 3&3=3，5&3=1，10&3=2，4&3=0，0&3=0。for 循环输出与运算结果：3,1,2,0,0,。答案选择 A 选项。

二、填空题

下列给定程序中，函数 fun 的功能是：把形参 a 所指数组中的偶数按原顺序依次存放到 a[0]、a[1]、a[2]…中，把奇数从数组中删除，偶数的个数通过函数值返回。

例如，若 a 所指数组中的数据最初排列为：9、1、4、2、3、6、5、8、7，删除奇数后，a 所指数组中的数据为：4、2、6、8，返回值为 4。

请在程序的下划线处填入正确的内容并将下划线删除，使程序得出正确的结果。

注意：不得增行或删行，也不得更改程序的结构！

试题程序如下：

```
#include<stdio.h>
#define N 9
int fun(int a[],int n)
{
   int i,j;
   j=0;
   for(i=0;i<n;i++)
      /*********found*********/
      if(①_____==0)
      {
         /*********found*********/
         ②_____=a[i];
         j++;
      }
   /*********found*********/
   return ③_____;
}
main()
{
   int b[N]={9,1,4,2,3,6,5,8,7},i,n;
   printf("\nThe original data:\n");
   for(i=0;i<N;i++)
      printf("%4d",b[i]);
   printf("\n");
   n=fun(b,N);
   printf("\nThe number of even:%d\n",n);
   printf("\nThen even:\n");
   for(i=0;i<n;i++)
      printf("%4d",b[i]);
   printf("\n");
}
```

【答案】

①a[i]%2

②a[j]

③j

【解析】

填空 1：if 语句条件表达式，判断数组元素是否为偶数，对 2 求余，结果为 0，则为偶数；结果为 1，则为奇数。

填空 2：将为偶数的数组元素按原顺序保存至数组 a 中，对应的保存下标为 j。

填空 3：最后按要求将偶数个数通过 return 语句返回给 main 函数。

三、改错题

下列给定程序中，函数 fun 的功能是：传入一个整数 m，计算如下公式的值。

$$t=\frac{1}{2}-\frac{1}{3}-\cdots-\frac{1}{m}$$

例如，若输入 5，则应输出-0.283333。

请改正程序中的错误，使它能得出正确的结果。

注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构！

试题程序如下：

```
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
double fun(int m)
{
   double t=1.0;
   int i;
   for(i=2;i<=m;i++)
      /*********found*********/
      t=1.0-1/i;
   /*********found*********/
   ;
}
main()
{
   int m;
   printf("\nPlease enter 1 integer number:\n");
   scanf("%d",&m);
   printf("\n\nThe result is:%lf\n",fun(m));
}
```

【答案】

（1）错误：t=1.0-1/i;

正确：t-=1.0/i;

（2）错误：;

正确：return t;

【解析】

错误 1：变量 t 初值为 1.0 中，运算通过循环语句进行，并要进行类型转换，由公式知应改为 t-=1.0/i;。

错误 2：循环结束后将 t 作为函数值返回。

四、设计题

函数 fun 的功能是：将 s 所指字符串中除了下标为奇数、同时 ASCII 值也为奇数的字符之外，其余的所有字符都删除，串中剩余字符所形成的一个新串放在 t 所指的数组中。

例如，若 s 所指字符串中的内容为“ABCDEFG12345”，其中字符 A 的 ASCII 码值虽为奇数，但所在元素的下标为偶数，因此必须删除；而字符 1 的 ASCII 码值为奇数，所在数组中的下标也为奇数，因此不应当删除，其他依此类推。最后 t 所指的数组中的内容应是“135”。

注意：部分源程序存在文件 PROG1.C 中。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入编写的若干语句。

试题程序如下：

```
#include <stdio.h>
#include <string.h>
void fun(char *s,char t[])
{

}
main()
{
   char s[100],t[100];
   printf("\nPlease enter string S:");
   scanf("%s",s);
   fun(s,t);
   printf("\nThe result is:%s\n",t);
}
```

答：

```
void fun(char *s,char t[])
{
   int i,j=0,n;
   n=strlen(s);
   for(i=0;i<n;i++)
      if(i%2!=0&&s[i]%2!=0)
      {
         t[j]=s[i];
         j++;
      }
   t[j]='\0';
}
```

【解析】定义局部变量 i，i%2!=0 表明 i 为奇数，当字符型变量与整数进行相关运算时，等价于该字符变量的 ASCII 码参与运算，因此 s[i]%2!=0 表明字符 s[i]的 ASCII 码为奇数。

# 第 4 章　选择结构程序设计

## 一、选择题

1．以下关于逻辑运算符两侧运算对象的叙述中正确的是（　　）。

A．可以是任意合法的表达式

B．只能是整数 0 或非 0 整数

C．可以是结构体类型的数据

D．只能是整数 0 或 1

【答案】A

【解析】C 语言的逻辑运算符比较特别，它的操作数没有明确的数据类型，可以是任意合法的表达式。答案选择 A 选项。

2．以下关于 C 语言数据类型使用的叙述中错误的是（　　）。

A．若要准确无误差地表示自然数，应使用整数类型

B．若要保存带有多位小数的数据，应使用双精度类型

C．若要处理如“人员信息”等含有不同类型的相关数据，应自定义结构体类型

D．若只处理“真”和“假”两种逻辑值，应使用逻辑类型

【答案】D

【解析】A 项正确，整数的表示是不存在误差的。B 项正确，float 类型称为单精度类型，double 类型称为双精度类型，一般系统中，为 float 类型的变量分配 4 个字节的存储单元，为 double 类型的变量分配 8 个字节的存储单元。C 项正确，struct 结构体，可以用来描述包含多种基本类型的复杂对象。D 项错误，C 语言中没有逻辑类型，在 C++中才引入的。答案选择 D 选项。

3．以下选项中，能表示逻辑值“假”的是（　　）。

A．1

B．0.000001

C．0

D．100.0

【答案】C

【解析】在 C 语言中，没有专门的“逻辑值”，而是用非 0 表示“真”，用 0 表示“假”。答案选择 C 选项。

4．下列关系表达式中，结果为“假”的是（　　）。

A．(3+4)>6

B．(3!=4)>2

C．3<=4||3

D．(3<4)==1

【答案】B

【解析】A 项，先执行 3+4=7，7>6，结果为真；B 项，先执行 3!=4 为真，即为 1，1>2 结果为假；C 项，先执行 3<=4 结果为真，即 1，再逻辑或运算只要左边值为 1，就不在执行右边运算，结果为真；D 项，先执行 3<4 结果为真，即为 1，再执行 1==1，结果为真。答案选择 B 选项。

5．以下表达式的值与 x 无关、其值恒为真的是（　　）。

A．0<x<5

B．x>10&&x<5

C．x>10||x<5

D．x<10&&x>5

【答案】A

【解析】逻辑与运算只有在“&&”符号两边操作均为真时，逻辑表达式为真；逻辑或当且只当“||”符号两边操作至少有一个为真时，逻辑表达式为真。B、C、D 中，使得表达式恒为真，都跟 x 的取值有关，错误；A 项，0<x<5，从左至右依次运算，0<x 为 0 或 1，均小于 5，故恒真。答案选择 A 选项。

6．当变量 c 的值不为 2、4、6 时，值也为“真”的表达式是（　　）。

A．(c==2)||(c==4)||(c==6)

B．(c>=2&&c<=6)||(c!=3)||(c!=5)

C．(c>=2&&c<=6)&&!(c%2)

D．(c>=2&&c<=6)&&(c%2!=1)

【答案】B

【解析】A 项，当变量 c 的值为 2、4、6 中任意一个时为真，不为 2、4、6 则为假；B 项，(c>=2&&c<=6)表示 c 大于等于 2 小于等于 6 是为真，(c!=3)表示 c 不等于 3 为真，(c!=5)表示 c 不等于 5 为真，“||”运算为有一项为真即为真，因此所有的整数都可使 B 项为真；C 项，“!(c%2)”表示 c 必须为偶数，并且 2=<c<=6，所以只有 c 为 2、4、6 时为真；D 项，表达式与 C 项中表达式等价。答案选择 B 选项。

7．设有定义：int a=0,b=1;，以下表达式中，会产生“短路”现象，致使变量 b 的值不变的是（　　）。

A．+a||++b

B．a++||++b

C．++a&&b++

D．a++&&b++

【答案】D

【解析】产生“短路”现象，即逻辑表达式靠后的部分不会被运算。对于逻辑或运算，表达式前半部分运算结果为 1 时产生“短路”现象，A 选项，+a 的值是 0，需要计算后半部分，B 选项，a 的值是先使用再自增 1，a++值是 0，也需要计算后半部分；对于逻辑与运算，表达式前半部分运算结果为 0 时产生“短路”现象，C 选项，a 的值是先自增 1 再传递，传值是 1，需要计算后半部分；D 项，a 的值是先传递再自增 1，传值是 0，无需要计算后半部分，可忽视 b++的值，因此 b 值不发生改变。答案选择 D 选项。

8．若有定义语句

int b=2;

则表达式(b<<2)/(3||b)的值是（　　）。

A．4

B．8

C．0

D．2

【答案】B

【解析】b=2，左移两位相当于乘以 4，2*4=8。3||b 的值为真即为 1，表达式的值是 8/1=8。答案选择 B 选项。

9．若 a 是数值类型，则逻辑表达式(a==1)||(a!=1)的值是（　　）。

A．1

B．0

C．2

D．不知道 a 的值，不能确定

【答案】A

【解析】运算符“||”为逻辑或运算符，即只要两边的条件表达式中有一个为“真”，则“逻辑或”的运算结果就为“真”。当 a==1 时，运算符左边的表达式为“真”；当 a!=1 时，运算符右边的表达式为“真”。因此，逻辑表达式(a==1)||(a!=1)的值恒为 1。答案选择 A 选项。

10．以下选项中，当 x 为大于 1 的奇数时，值为 0 的表达式是（　　）。

A．x%2==1

B．x/2

C．x%2!=0

D．x%2==0

【答案】D

【解析】当 x 为大于 1 的奇数，x%2==1，则表达式 x%2==0 为假（即值为 0），答案选择 D 选项。

11．若有定义语句

int k1=10,k2=20;

执行表达式(k1=k1>k2)&&(k2=k2>k1)后，k1 和 k2 的值分别为____。

A．0 和 1

B．0 和 20

C．10 和 1

D．10 和 20

【答案】B

【解析】在 C 语言中，没有专门的“逻辑值”，而是用非 0 表示“真”，用 0 表示“假”。根据优先级，先计算括号内的。括号内先比较 k1 和 k2 的大小，由于 k2>k1，故左边括号中的 k1 与 k2 比较的结果为假，值为 0，再赋值给 k1，可得 k1 的值为 0，由于 0 与任何值相与结果都是 0，故右边一个括号不需运算，即 k2 的值不变，仍为 20。答案选择 B 选项。

12．已知 a=5，b=6，c=7，d=8，m=2，n=2，执行(m=a>b)&&(n=c<d)后 n 的值为（　　）。

A．1

B．0

C．2

D．-1

【答案】C

【解析】“=”优先级低于“<”“>”。“&&”逻辑与表达式。exp1 && exp2，规则为：对 exp1 求值，若为 0，则表达式为 0，则不计算 exp2；若 exp1 非 0，则求 exp2 值，作为表达式值。本题计算过程为：先判断 a>b 为假，m=0，整个逻辑表达式为假，不计算右表达式，n=2。答案选择 C 选项。

13．执行以下程序段后，w 的值为（　　）。

```
int w='A',x=14,y=15;
w=((x||y)&&(w<'a'));
```

A．-1

B．NULL

C．1

D．0

【答案】C

【解析】因为 x=14，y=15 都是非零整数，因此 x||y=1。大写字母的 ASCII 码值小于小写字母，w<'a'也为真，因此(x||y)&&(w<'a')的值为 1。答案选择 C 选项。

14．有以下程序：

```
#include<stdio.h>
main()
{
    int a,b,c=241;
    a=c/100%9;
    b=(-1)&&(-1);
    printf("%d,%d\n",a,b);
}
```

程序运行后的输出结果是（　　）。

A．2,1

B．6,1

C．6,0

D．2,0

【答案】A

【解析】程序前面表达式等同于 a=(c/100)%9，程序执行过程为：c=241，c/100=2，2%9=2，得 a=2；“=”优先级低于“&&”，非 0 值为真，0 值为假，即(-1)为真，逻辑与表达式(-1)&&(-1)为真，得 b=1。答案选择 A 选项。

15．有以下程序：

```
#include<stdio.h>
main()
{
   int a=-2,b=0;
   while(a++&&++b);
   printf("%d,%d\n",a,b);
}
```

程序运行后的输出结果是（　　）。

A．1,3

B．0,2

C．0,3

D．1,2

【答案】D

【解析】第一次循环之后 a=-1，b=1，第二次循环之后，a=0，b=2，第三次循环时，由于 a++表示 a 加之前的值，即 a++=0，逻辑与值为 0，所以不再执行++b，这时，a 自增 1，由 0 变为 1，而 b 的值仍为 2。答案选择 D 选项。

16．有以下程序：

```
#include <stdio.h>
main()
{
   int n=2,k=0;
   while(k++&&n++>2);
   printf("%d %d\n",k,n);
}
```

程序运行后的输出结果是（　　）。

A．0 2

B．1 3

C．5 7

D．1 2

【答案】D

【解析】(表达式 1)&&(表达式 2)中，如果表达式 1 为假或 0，那么表达式 2 就不会被执行。程序首先进入 while 的判断语句，执行 k++和 n++。k++为零，不满足循环条件，所以 n++不会被执行，while 循环结束后，k 自增为 1，n 没有进行运算，仍为 2。答案选择 D 选项。

17．以下叙述中正确的是（　　）。

A．分支结构是根据算术表达式的结果来判断流程走向的

B．在 C 语言中，逻辑真值和假值分别对应 1 和 0

C．对于浮点变量 x 和 y，表达式：x==y 是非法的，会出编译错误

D．关系运算符两边的运算对象可以是 C 语言中任意合法的表达式

【答案】D

【解析】C 语言中，运算符两侧的运算对象可以任意合法的表达式，选项 D 正确；非 0 和 0 分别对应逻辑真值和假值，选项 B 错；关系运算符“==”两侧的运算对象只要为合法的表达式就可以编译通过，选项 C 错；分支结构是根据 if 或者 else if 后面的圆括号内的表达式（不仅仅是算术表达式，还有赋值表达式、逗号表达式、关系表达式、逻辑表达式等）的值来判定流程走向的，选项 A 错误；答案选择 D 选项。

18．if 语句的基本形式是：
if(表达式)语句
以下关于“表达式”值的叙述中正确的是（　　）。
A．必须是逻辑值
B．必须是整数值
C．必须是正数
D．可以是任意合法的数值
【答案】D
【解析】if 中的表达式一般是关系表达式或逻辑表达式，用于描述选择结构的条件，但也可以是其他类型表达式，在其值非零时为真，所以任意合法的表达式都可以做 if 语句的判断条件。答案选择 D 选项。

19．以下是 if 语句的基本形式：
if(表达式)语句
其中“表达式”（　　）。
A．必须是逻辑表达式
B．必须是关系表达式
C．必须是逻辑表达式或关系表达式
D．可以是任意合法的表达式
【答案】D
【解析】if 中的表达式一般是关系表达式或逻辑表达式，用于描述选择结构的条件，但也可以是其他类型表达式，在其值非零时为真，所以任意合法的表达式都可以做 if 语句的判断条件。答案选择 D 选项。

20．对于“if(表达式)语句”，以下叙述正确的是（　　）。
A．“表达式”的值可以是任意合法的数值
B．在“表达式”中不能出现变量
C．在“表达式”中不能出现常量
D．“表达式”的值必须是逻辑值
【答案】A
【解析】单分支选择语句 if 语句格式如下：if(表达式)语句体。“表达式”可以是任意合法的 C 语言表达式，包括关系表达式和逻辑表达式等，也可以是任意的数值类型（包括整型、实型、字符型等），A 项正确。“表达式”中可以出现变量或者常量，也不一定必须是逻辑值，B、C、D 项错误。答案选择 A 选项。

21．若变量已正确定义，在 if(W)printf("%d\n",k);中，以下不可替代 W 的是（　　）。
A．a<>b+c
B．c=getchar()
C．a==b+c
D．a++
【答案】A
【解析】在 C 语言中，表示不等于不能用“<>”，而只能使用“!=”。答案选择 A 选项。

22．为了避免在嵌套的 if-else 语句中产生二义性，C 语言规定与 else 子句配对是（　　）。
A．与其在同一行上的 if 子句
B．在其之后最近的不带 else 的 if 子句
C．与其缩排位置相同的 if 子句
D．在其之前最近的不带 else 的同层 if 子句

【答案】D

【解析】if 总是与它上面最近的、未配对的 else 配对。答案选择 D 选项。

23．下列条件语句中，输出结果与其他语句不同的是（　　）。

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

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

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

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

【答案】D

【解析】A 项，如果 a 的值为 1，即输出 x，否则输出 y；B 项，如果 a 的值为 0，则输出 y，否则输出 x；C 项，如果 a 不等于 0，则输出 x，否则输出 y；D 项，如果 a 的值为 0，则输出 x，否则输出答案 y。因此 D 项与其他三个选项不同。答案选择 D 选项。

24．设有定义：

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

以下语句中执行效果与其他三个不同的是（　　）。

A．if(a>b)c=a,a=b,b=c;

B．if(a>b){c=a,a=b,b=c;}

C．if(a>b)c=a;a=b;b=c;

D．if(a>b){c=a;a=b;b=c;}

【答案】C

【解析】C 语言中 if 语句后面只跟一条语句时，可以省略大括号。即 if 语句仅作用于紧随其后的那条语句或者是复合语句的内容，所以 A 项，执行三条语句组成的复合语句；BD 两项执行大括号中的三条语句；而 C 项只执行 c=a;。答案选择 C 选项。

25．有以下程序段，若变量已正确定义并赋值（　　）。

```
if(a>b) printf("x=%d,",x);
else printf("y=%d,",y);
if(a<=b)i++;
else j++;
```

则与此程序段功能相同的选项是（　　）。

A.

```
if(a>b)
{
   printf("x=%d,",x);
   j++
}
else
{
   printf("y=%d,",y);
   i++
}
```

B.

```
if(a>b)
{
   printf("x=%d,",x);
   i++
}
else
{
   printf("y=%d,",y);
   j++
}
```

C.

```
if(a<=b)
{
   printf("x=%d,",x);
   i++
}
else
{
   printf("y=%d,",y);
   j++
}
```

D.

```
if(a>=b)
{
   printf("x=%d,",x);
   i++
}
else
{
   printf("y=%d,",y);
   j++
}
```

【答案】A

【解析】程序段执行过程为：如果 a>b，输出 x，否则输出 y；如果 a<=b，i 加 1，否则 j 加 1。A 项如果 a>b，输出 x 且 j 加 1，否则输出 y 且 i 加 1，与题目中功能相同，A 项正确。B 项如果 a>b，输出 x 且 i 加 1，否则输出 y 且 j 加 1，与题目中功能不相同，B 项错误。C 项如果 a<=b，输出 x 且 i 加 1，否则输出 y 且 j 加 1，与题目中功能不相同，C 项错误。D 项判断条件为 a>=b，多了 a=b，D 项错误。答案选择 A 选项。

26．有以下程序：

```
#include<stdio.h>
main()
{
   int x=0x13;
   if(x=0x12)printf("True");
   printf("False\n");
}
```

程序运行后的输出结果是（　　）。

A．True

B．TrueFalse

C．False

D．TrueFalseTrue

【答案】B

【解析】if(x=0x12)条件语句为赋值语句，注意赋值操作符“=”与相等操作符“==”的区别，此处是赋值语句“=”0x12 即十进制的 18，给 x 赋值 18，因此 if 的判断条件为真，执行输出语句，输出 True；之后再执行下一个输出语句，输出 False，答案选择 B 选项。

27．有以下程序段：

```
scanf("%d%d%d",&a,&b,&c);
if(a>b)a=b;
if(a>c)a=c;
printf("%d\n",a);
```

该程序段的功能是（　　）。

A．输出 a、b、c 中的最小值

B．输出 a、b、c 中的最大值

C．输出 a 的原始值

D．输出 a、b、c 中值相等的数值

【答案】A

【解析】程序执行过程为：从键盘读入三个整型数据，依次赋给 a，b，c，判断 a>b，若成立，将较小的值 b 赋值给较大的值 a，判断 a>c，若成立，将较小的值 c 赋值给较大的值 a，实现了将从键盘读入的数据中最小值赋给 a 的功能，最后输出 a，即输出最小值。答案选择 A 选项。

28．有以下程序段：

```
scanf("%d%d%d",&a,&b,&c);
if(a<b)a==b;
if(a<c)a==c;
printf("%d\n",a);
```

该程序段的功能是（　　）。

A．输出 a、b、c 中值相等的数值

B．输出 a、b、c 中的最大值

C．输出 a、b、c 中的最小值

D．输出 a 的原始值

【答案】D

【解析】程序段执行过程为：从键盘读入 3 个整型数据分别赋值给 a，b，c，如果 a<b，判断 a 与 b 是否相等，无论结果如何不做任何改变。如果 a<c，判断 a 与 c 是否相等，无论结果如何均不做任何改变。区分“==”操作符和“=”操作符。最后对于输入的 a，b，c 的值不做任何改变，即输出 a 的原始值，答案选择 D 选项。

29．以下程序的功能是判断输入的一个整数是否能被 3 或 7 整除，若能整除，输出 YES，否则输出 NO。在下划线处应填入的选项是（　　）。

```
#include<stdio.h>
main()
{
   int k;
   printf("Enter a int number:");
   scanf("%d",&k);
   if _____ printf("YES\n");
   else printf("NO\n");
   printf("%d\n",k%3);
}
```

A. ((k%3==0)||(k%7==0))
B. (k/3==0)||(k/7==0)
C. ((k%3=0)||(k%7=0))
D. ((k%3==0)&&(k%7==0))

【答案】A

【解析】是否能被 3 或 7 整除表达式为(k%3==0)||(k%7==0)，A 项正确。B 项“/”为除号，错误。C 项中“=”为赋值运算符，不是逻辑运算符“==”，错误。D 项“&&”为逻辑与，本题应该为逻辑或“||”，错误。答案选择 A 选项。

30．有以下程序：

```
#include<stdio.h>
main()
{
   if('\0'==0)putchar('1');
   if('0'==0)putchar('2');
   if('a'>'b')putchar('3');
}
```

程序运行后的输出结果是（　　）。

A. 1
B. 123
C. 23
D. 3

【答案】A

【解析】'\0'是转义字符，表示空字符，对应的 ASCII 码为 0，成立，输出 1；'0'是字符常量和 0 不等，'a'和'b'都是字符常量，因为'b'>'a'，不执行 if 后面的语句，所以最后输出只有 1，答案选择 A 选项。

31．有以下程序：

```
#include <stdio.h>
main()
{
   int x;
   scanf("%d",&x);
   if(x>10) printf("1");
   else if(x>20) printf("2");
   else if(x>30) printf("3");
}
```

若运行时输入：35<回车>，则输出结果是（　　）。

A．123
B．2
C．3
D．1
【答案】D
【解析】程序执行过程为：输入 35<回车>，scanf 函数从键盘读入 35 赋值给 x，对 if 条件进行判断，35>10，条件成立，输出 1，不再执行下面的 else if 语句，程序结束。答案选择 D 选项。

32．有以下程序段

```
#include <stdio.h>
main()
{
   int a,b,c;
   a=10;b=50;c=30;
   if(a>b)a=b,b=c;c=a;
   printf("a=%d b=%d c=%d\n",a,b,c);
}
```

程序的输出结果是（　　）。
A．a=10 b=50 c=30
B．a=10 b=50 c=10
C．a=10 b=30 c=10
D．a=50 b=30 c=50
【答案】B
【解析】C 语言中使用分号来作为语句的结束，所以 a=b,b=c;是一条含有逗号运算符的语句，是 if 语句的执行体。因为题中 a <b，if 条件不满足，直接执行 c=a。答案选择 B 选项。

33．若变量已正确定义，有以下程序段：

```
int a=3,b=5,c=7;
if(a>b) a=b;c=a;
if(c!=a)c=b;
printf("%d,%d,%d\n",a,b,c);
```

其输出结果是（　　）。
A．程序段有语法
B．3,5,3
C．3,5,5
D．3,5,7
【答案】B
【解析】a=3，b=5，因此 a>b 条件不成立，所以不执行 a=b。然后 c=a=3。此时“c!=a”条件不成立，不执行“c=b;”语句。结果为 a=3，b=5，c=3。答案选择 B 选项。

34．以下函数按每行 8 个输出数组中的数据：

```
void fun(int *w,int n)
{
   int i;
   for(i=0;i<n;i++)
   {
      _____
      printf("%d",w[i]);
   }
   printf("\n");
}
```

下画线处应填入的语句是（　　）。

A. if(i/8==0)printf("\n");

B. if(i/8==0)continue;

C. if(i%8==0)printf("\n");

D. if(i%8==0)continue;

【答案】C

【解析】每行输出 8 个数组数据后输入一个换行，所以应该采取对 8 取余的方法，余数循环一次便换行一次，所以语句为 if(i%8==0) printf("\n");，答案选择 C 选项。

35. 有以下程序：

```
#include<stdio.h>
main()
{
   int x=1,y=2,z=3;
   if(x>1)
      if(y>x)putchar('A');
      else putchar('B');
   else
      if(z<x)putchar('C');
      else putchar('D');
}
```

程序的运行结果是（　　）。

A. D

B. C

C. B

D. A

【答案】A

【解析】A 项正确，main 函数中，首先判断条件 x>1，因为 x=1，不满足条件，程序跳入 else 语句判定 z<x 是否成立，z=3，x=1，条件不成立，输出字母 D。答案选择 A 选项。

36. 有以下程序：

```
#include<stdio.h>
main()
{
   int x=1,y=2,z=3;
   if(x>y)
      if(y<z) printf("%d",++z);
      else printf("%d",++y);
   printf("%d\n",x++);
}
```

程序运行的结果是（　　）。

A．331

B．41

C．2

D．1

【答案】D

【解析】else 子句总是与前面最近的不带 else 的 if 相结合，与书写格式无关，所以程序中的 else 语句与第二个 if 语句配对，且 if 和 else 都在第一个 if 的控制范围内。首先判断 x>y 不成立，退出第一个 if 语句，执行 printf("%d\n",x++)。输出结果为 1。答案选择 D 选项。

37．有以下程序

```
#include <stdio.h>
main()
{
   int x=1, y=0;
   if (!x) y++;
   else if (x==0)
      if (x) y+=2;
      else y+=3;
   printf("%d\n", y);
}
```

程序运行后的输出结果是（　　）。

A．0

B．2

C．1

D．3

【答案】A

【解析】在该题中，选择结构的表达式都不成立，所以整个选择语句都没有执行，y 值没有发生改变，答案选择 A 选项。

38．有以下程序

```
#include<stdio.h>
main()
{
   int a=1,b=0;
   if(!a)b++;
   else if(a==0) if(a) b+=2;
   else b+=3;
   printf("%d\n",b);
}
```

程序运行后的输出结果是（　　）。

A．0

B．1

C．2

D．3

【答案】A

【解析】根据在 if-else 语句中，else 总是和最近的 if 配对的原则，本题中层次关系是：if(!a)与 else if(a==0)是一组，在最外层。而 if(a)与 else 是一组，位于 else if(a==0)条件的内层。if(!a)与 else if(a==0)条件均不成立，所以 b 未进行任何操作仍为初始值 0。答案选择 A 选项。

39．有以下程序：

```
#include<stdio.h>
main()
{
   int x;
   scanf("%d",&x);
   if(x<=3);
   else if(x!=10)
      printf("%d\n",x);
}
```

程序运行时，输入的值在（　　）范围才会有输出结果。

A．不等于 10 的整数

B．大于 3 且不等于 10 的整数

C．大于 3 或等于 10 的整数

D．小于 3 的整数

【答案】B

【解析】第一个 if 有一个 else 语句，若 x<=3，则不进行任何操作，若 x>3，则进入 else 语句中，在 else 语句中只有当 x!=10 时才会有输出结果，所以输入的值的范围应为大于 3 且不等于 10 的整数。答案选择 B 选项。

40．有以下程序：

```
#include <stdio.h>
main()
{
   int a=1,b=2,c=3,d=0;
   if(a==1&&b++==2)
      if(b!=2||c--!=3)
         printf("%d,%d,%d\n",a,b,c);
      else printf("%d,%d,%d\n",a,b,c);
   else printf("%d,%d,%d\n",a,b,c);
}
```

程序运行后的输出结果是（　　）。

A. 1,2,3

B. 1,3,2

C. 1,3,3

D. 3,2,1

【答案】C

【解析】初始 a=1，b=2，c=3，d=0；因为 a 为 1，b++本身为 b 加之前的值，即 b++为 2，所以第一个 if 语句的判断条件为真，进入 if 语句。b 经过 b++运算后值为 3，所以第二个 if 语句为真（或运算，一个为真，整体都为真，并且第一个为真时第二个运算不会被执行），c--操作没有执行，所以 c 值仍为 3。答案选择 C 选项。

41．若有以下程序

```
#include<stdio.h>
main()
{
   int a=1,b=2,c=3,d=4;
   if((a==2)||(b==1))c=2;
   if((c==3)&&(d==-1))a=5;
   printf("%d,%d,%d,%d\n",a,b,c,d);
}
```

则程序的输出结果是（　　）。

A. 2,2,2,4

B. 2,1,2,-1

C. 5,1,2,-1

D. 1,2,3,4

【答案】D

【解析】&&和||是逻辑运算符，&&符号两侧都为真时，结果为真，否则返回假；||符号两侧都为假时，结果为假，否则返回真。题目中两个 if 语句条件判定都是 false，abcd 的值不会发生改变。答案选择 D 选项。

42．若有以下程序

```
#include<stdio.h>
main()
{
   int a=1,b=2,c=3,d=4;
   if((a==2)&&(b==1))c=2;
   if((c==3)||(d==-1))a=5;
   printf("%d,%d,%d,%d\n",a,b,c,d);
}
```

则程序的输出结果是（　　）。

A．5,2,3,4

B．2,1,2,-1

C．2,2,2,4

D．1,2,3,4

【答案】A

【解析】&&和||是逻辑运算符，&&符号两侧都为真时，结果为真，否则返回假；||符号两侧都为假时，结果为假，否则返回真。第一个 if 判定表达式中是两个赋值语句，第一个结果为假，参数值均未改变；在第二个 if 判定表达式中，c==3 判定为真，然后执行 d==-1 为假，所以执行 a=5，其余参数值未改变，答案选择 A 选项。

43．有以下程序

```
#include<stdio.h>
main()
{
   int a=0,b=0,c=0,d=0;
   if(a=1)b=1;c=2;
   else d=3;
   printf("%d,%d,%d,%d\n",a,b,c,d);
}
```

程序输出（　　）。

A．0,0,0,3

B．编译有错

C．1,1,2,0

D．0,1,2,0

【答案】B

【解析】如果 if 的执行语句含有多个语句（两个以上），则必须使用复合语句，即用花括号把一组语句括起来；否则，紧跟 if 的下一条语句是它的执行语句，因此 c=2 不是 if 执行语句，它是在 if 和 else 之间的语句。在程序中 else 必须与 if 配对，共同组成一条 if-else 语句，中间不能出现其他语句，因此该程序编译错误。答案选择 B 选项。

44．有以下程序：

```
#include <stdio.h>
main()
{
   int x=0x13;
   if(x=0x18)printf("T");
   printf("F");
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．TF

B．T

C．F

D．TFT

【答案】A

【解析】x=0x18 为赋值表达式，十六进制数 0x18 非 0，故 x 非 0，if 条件成立输出 T，之后再输出 F 与回车符。程序运行后的输出结果是 TF。答案选择 A 选项。

45．有以下计算公式：

$$y=\begin{cases}\sqrt{x} & (x\geq 0)\\ -x & (x<0)\end{cases}$$

若程序前面已在命令行中包含 math.h 文件，不能够计算上述公式的程序段是（　　）。

A．y=sqrt(x); if(x<0)y=sqrt(-x);

B．if(x>=0)y=sqrt(x); else y=sqrt(-x);

C．if(x>=0)y=sqrt(x); if(x<0)y=sqrt(-x);

D．y=sqrt(x>=0?x:-x);

【答案】A

【解析】sqrt 函数是 math.h 文件中用来计算平方根的苦函数。A 项错误，当 x 小于零时，A 项会出现运行时错误。B 项使用 if-else 语句实现功能；C 项使用两条 if 语句实现；D 项使用三目运算符实现，结果等价于 BC 两项。答案选择 A 选项。

46．如有表达式(w)?(-x):(++y)，则其中与 w 等价的表达式是（　　）。

A．w==1

B．w==0

C．w!=1

D．w!=0

【答案】D

【解析】条件表达式形式为<表达式 1>?<表达式 2>:<表达式 3>。表达式 1 的值为真，结果为表达式 2 的值；表达式 1 的值为假，结果为表达式 3 的值。可见表达式 w 等价于 w!=0。答案选择 D 选项。

47．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（　　）。

A．(x-y==0)

B．(x-y<0)

C．(x-y>0)

D．(x-y<0||x-y>0)

【答案】D

【解析】条件表达式：x=表达式 1?表达式 2:表达式 3，其含义是：先求解表达式 1，若其值为非 0（真），则求解表达式 2，将表达式 2 的值赋给 x，若表达式 1 的值为 0（假），则求解表达式 3，将表达式 3 的值赋给 x。在本题中与表达式(x-y)等价的是(x-y)!=0，即(x-y<0||x-y>0)。答案选择 D 选项。

48．有如下嵌套的 if 语句

```
if(a<b)
   if(a<c)k=a;
   else k=c;
else
   if(b<c)k=b;
   else k=c;
```

以下选项中与上述 if 语句等价的语句是（　　）。

A．k=(a<b)?a:b;k=(b<c)?b:c;

B．k=(a<b)?((b<c)?a:b):((b<c)?b:c);

C．k=(a<b)?((a<c)?a:c):((b<c)?b:c);

D．k=(a<b)?a:b;k=(a<c)?a:c;

【答案】C

【解析】C 语言的语法规定：else 子句总是与前面最近的不带 else 的 if 相结合，与书写格式无关。本题中，嵌套的 if 语句功能是将 k 赋值为 a、b、c 中的最小值。A 项，没有比较 a、c 的大小；B 项，当 a<b 且 b>c 时 k

赋值为 b，此时 b 是最大值，与题意不符；D 项，中没有比较 b、c 大小。答案选择 C 选项。

49．以下程序段中，与语句：

k=a>b?(b>c?1:0):0;

功能相同的是（　　）。

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;
```

【答案】A

【解析】三元运算符表达式的形式为：表达式 1？表达式 2：表达式 3，当表达式 1 的值为真时，结果为表达式 2 的值；当表达式 1 的值为假时，结果为表达式 3 的值。首先判断 a、b 的关系：①如果 a>b，执行语句（b>c?1:0）；判断 b、c 的关系，如果 b>c，k=1，否则 k=0；②如果 a≤b，则 k=0。综上所述：当 a>b 且 b>c 时，k=1，否则 k=0，与 A 项语句功能相同。答案选择 A 选项。

50．有语句：

k=x<y?(y<z?1:0):0;

以下选项中，与此语句功能相同的是（　　）。

A．if(x<y||y<z)k=1;else k=0;

B．if(x<y)k=0;else if(y<z)k=1;

C．if(x<y)if(y<z)k=1;else k=0;

D．if(x<y && y<z)k=1;else k=0;

【答案】D

【解析】D 项正确，题中，先判断 x<y 是否成立，如果为假，直接返回 0，如果为真，再判断 y<z，如果为真返回为 1，否则返回 0，综合所知，只有 x<y 且 y<z 时返回 1，否则返回 0。答案选择 D 选项。

51．以下程序段中，不能实现条件“如果 a<b 则 x=10，否则 x=-10”的是（　　）。

A．x=(a>=b)?-10:10;

B．if(a<b)x=10;else x=-10;

C．x=-10; if(b<a)x=10;

D．if(a<b)x=10;if(b<a)x=-10;

【答案】D

【解析】A 项：条件运算符?运算过程为：如果 a>=b，x=-10，否则 x=10，能实现题目中功能。B 项：如果 a<b，则 x=10，否则 x=-10，能实现题目中功能。C 项：首先赋值 x=-10，如果 a<b，则 x=10，即在 a>=b 情况下有 x=-10，能实现题目中功能。D 项：如果 a<b，则 x=10，如果 b<a，则 x=-10，其中没有对 a=b 进行判断，不能实现题目中功能，答案选择 D 选项。

52．若有定义：

int a=0,b=0,c=0,d=0;

有 C 语言表达式(a++&&b++)?c++:d++，以下关于其执行顺序的叙述正确是（　　）。

A．先执行 a++，表达式 a++的值为 0，由此即可确定(a++ && b++)值为 0，因此执行 d++

B. 先执行 a++，表达式 a++的值为 0；再执行 b++，表达式 b++的值为 0，由此可确定(a++&&b++)值为 0，因此执行 d++

C. 先执行 a++，表达式 a++的值为 1；再执行 b++，表达式 b++的值为 1，由此可确定(a++&&b++)值为 1，因此执行 c++

D. 先执行 b++，表达式 b++的值为 1；再执行 a++，表达式 a++的值为 1，由此可确定(a++&&b++)值为 1，因此执行 c++

【答案】A

【解析】把握前置运算和后置运算的基本运算规则。表达式 1?表达式 2:表达式 3，若表达式 1 为真，则执行表达式 2，否则执行表达式 3。逻辑与运算符遵循“短路求值”策略，即在进行求值时，只要最终的结果已经可以确定是假，求值过程便告终止，表达式(a++&&b++)中，左操作数 a++的值为 0，已经可以确定整个逻辑表达式的结果为 0，因此右操作数 b++不再求解，直接执行表达式 3，即 d++，答案选择 A 选项。

53. 若有定义：

int a=0,b=0,c=0,d=0;

以下关于 C 语言表达式：(++a ||++b)? ++c:++d 执行顺序的叙述正确的是（　　）。

A. 先执行++a，表达式++a 的值为 1；再执行++b，表达式++b 的值为 1，由此可确定(++a||++b)值为 1，因此执行++c

B. 先执行++a，表达式++a 的值为 1，由此可确定(++a||++b)值为 1，因此执行++c

C. 先执行++b，表达式++b 的值为 1；再执行++a，表达式++a 的值为 1，由此可确定(++a||++b)值为 1，因此执行++c

D. 先执行++a，++b，++c，++d，使得 a，b，c，d 的值都为 1，由此可确定(++a||++b)值为 1，因此执行++c

【答案】B

【解析】++a 表示先将变量 a+1，再执行其他操作，逻辑或运算符遵循“短路求值”策略，即只有在仅靠左操作数的值无法确定该逻辑表达式的结果时，才会求解右操作数，表达式(++a||++b)中，左操作数++a 的值为 1，已经可以确定整个逻辑表达式的结果为 1，因此右操作数++b 不再求解，直接执行表达式 2，即++c，B 项正确。答案选择 B 选项。

54. 有以下程序：

```
#include <stdio.h>
main()
{
   char a='H';
   a=(a>='A'&&a<='Z')?(a-'A'+'a'):a;
   printf("%c\n",a);
}
```

程序运行后的输出结果是（　　）。

A. A

B. a

C. H

D. h

【答案】D

【解析】首先将 H 赋给变量 a，问号前一个表达式成立，问号后面减去一个大写字母 A 再加上小写字母 a 实际是将原来的大写字母转换成小写字母，即 H 转换成 h，再将结果赋给 a。答案选择 D 选项。

55. 有以下程序

```
#include<stdio.h>
main()
{
   int x;
   for(x=3;x<6;x++)
      printf((x%2)?("*%d"):("#%d"),x);
   printf("\n");
}
```

程序的输出结果是（　　）。

A. *3#4*5

B. #3*4#5

C. *3*4#5

D. *3#4#5

【答案】A

【解析】若满足 x%2!=0，输出*x，否则，输出#。答案选择 A 选项。

56. 有以下程序：

```
#include<stdio.h>
int m1(int x,int y)
{
   return x<=y?x:y;
}
int m2(int x,int y)
{
   return x<=y?y:x;
}
int fun(int a,int b)
{
   return a+b;
}
main()
{
   int x=2,y=3,z=1;
   printf("%d\n",fun(m1(x,y),m2(y,z)));
}
```

程序的运行结果是（　　）。

A. 6

B. 5

C. 4

D. 3

【答案】B

【解析】条件运算符“?:”语法形式为：exp1?exp2:exp3。执行规则为：计算表达式 exp1 的值，测试其是否为 0；如果 exp1 的值非 0，则对 exp2 求值，并把这个值作为条件表达式的结果输出，不计算 exp3；如果 exp1 的值为 0，则对 exp3 求值，并把这个值作为条件表达式的结果输出，不计算 exp2。函数 m1 实现返回传入两个参数中的最小值。函数 m2 实现返回传入两个参数中的最大值。函数 fun 实现返回传入两个参数之和。程序执行过程为：调用 m1(2,3)返回 2，调用 m2(3,1)返回 3，调用 fun(2,3)返回 5，输出结果 5，答案选择 B 选项。

57. 有以下程序：

```
#include<stdio.h>
main()
{
   int a=0,b=0,c=0,d=0;
   (++a||++b)?++c:++d;
   printf("%d,%d,%d,%d\n",a,b,c,d);
}
```

程序的运行结果是（　　）。

A. 1,0,1,0

B. 1,1,0,1

C. 1,0,0,1

D. 1,1,1,0

【答案】A

【解析】本题程序执行过程为：求解逻辑表达式(++a||++b)，++a 的值为 1，则整个表达式为真，不计算++b，然后求解++c，不计算++d，最后得到 a=1，b=0，c=1，d=0，答案选择 A 选项。

58. 下列叙述中正确的是（　　）。

A. 在 switch 语句中，不一定使用 break 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. break 语句只能用于 switch 语句

【答案】A

【解析】break 语句功能是跳出正在执行的条件语句或循环语句，switch 语句中可以根据需要选择是否使用 break 语句，A 项正确；default 语句在 switch 语句中可以省略，因此 B 项错误；switch 语句中并非每个 case 后都需要使用 break 语句，因此 C 项错误；break 语句还可以用于 for 等循环结构中，因此 D 项错误。答案选择 A 选项。

59. 若有定义：

```
float x=1.5;
int a=1,b=3,c=2;
```

则正确的 switch 语句是（　　）。

A.

```
switch(a+b)
{
   case 1: printf("*\n");
   case c: printf("**\n");
}
```

B.

```
switch((int)x);
{
   case 1: printf("*\n");
   case 2: printf("**\n");
}
```

C.

```
switch(x)
{
   case 1.0: printf("*\n");
   case 2.0: printf("**\n");
}
```

D.

```
switch(a+b)
{
   case 1: printf("*\n");
   case 2+1: printf("**\n");
}
```

【答案】D

【解析】D 项正确，标号可以是整型表达式。A 项错误，case 是关键字，与其后面的常量表达式合称 case 语句标号。常量表达式的类型必须与 swicth 后圆括号中的表达式类型相同，各 case 语句标号的值应该互不相同。c 是变量，不能作为表达式放在 case 后面；B 项错误，switch 后面不应该有分号；C 项错误，switch 参数值类型必须是这几种类型之一：int，long，short，byte，char。答案选择 D 选项。

60．若以下选项中的变量全部为整型变量，且已正确定义并赋值，则语法正确的 switch 语句是（　　）。

A.

```
switch(a+9)
{
   case c1:y=a-b;
   case c2:y=a+b;
}
```

B.

```
switch a*b
{
   case 10:x=a+b;
   default:y=a-b;
}
```

C.

```
switch(a+b)
{
   case1:case2:case3:y=a+b;break;
   case0:case4:y=a-b;
}
```

D.

```
switch(a*a+b*b)
{
   default:break;
   case 3:y=a+b;break;
   case 2:y=a-b;break;
}
```

【答案】D

【解析】A 项错误，case 后面的应该为整形或字符型常量；B 项错误，switch 后面的表达式需要在括号内；C 项错误，case 和后面的常量表达式应该由空格隔开。答案选择 D 选项。

61．若有定义语句

int a,b;double x;

则下列选项中没有错误的是（　　）。

A.

```
switch(x%2)
{
   case 0:a++;break;
   case 1:b++;break;
   default:a++;b++;
}
```

B.

```
switch((int)x/2.0)
{
   case 0:a++;break;
   case 1:b++;break;
   default:a++;b++;
}
```

C.

```
switch((int)x%2)
{
   case 0:a++;break;
   case 1:b++;break;
   default:a++;b++;
}
```

D.

```
switch((int)(x)%2)
{
   case 0.0:a++;break;
   case 1.0:b++;break;
   default:a++;b++;
}
```

【答案】C

【解析】switch 语句中，表达式的类型应与 case 语句后的常量类型保持一致，并且 switch 的判断条件只能为整型或字符型，case 后面为常量表达式。A 项，x%2 得到的是浮点型数据，而 case 语句后的常量是整型数据，类型不一致；B 项，(int)x/2.0 得到的也是浮点型数据，类型不一致；D 项，(int)x%2.0 得到的是整型数据，而 case 语句后的常量是浮点型数据，类型也不一致。答案选择 C 选项。

62．以下选项中与 if(a==1) a=b;else a++;语句功能不同的 switch 语句是（　　）。

A.

```
switch(a)
{
   case 1: a=b; break;
   default: a++;
}
```

B.

```
switch(a==1)
{
   case 0: a=b; break;
   case 1:a++;
}
```

C.

```
switch(a)
{
   default :a++; break;
   case 1: a=b;
}
```

D.

```
switch(a==1)
{
   case 1: a=b;break;
   case 0: a++;
}
```

【答案】B

【解析】在 switch 语句的执行过程中，执行完 case 后面的语句后，如果遇到 break 语句就停止，否则将继续执行下一个 case 中的语句，直到遇到 break 语句。B 项，当 a==1 是 a++，a!=1 时 a=b，刚好和题干要求相反。答案选择 B 选项。

63．有以下程序：

```
#include <stdio.h>
main()
{
   int k,n=0;char c,str[]="teach";
   for(k=0;str[k];k++)
   {
      c=str[k];
      switch(k)
      {
         case 1:case 3:case 5: putchar(c); printf("%d",++n); break;
         default:putchar('N');
      }
   }
}
```

程序的运行结果是（　　）。

A. Ne1NN

B．e1a2e3

C．Ne1Nc2N

D．Na1NNNN

【答案】C

【解析】程序执行过程：k=0 时，c=str[0]='t'，执行 default 分支，输出 N；k=1 时，c='e'，执行 case 1 分支，没有 break 语句，继续执行 case 3 分支，没有 break 语句，继续执行 case 5 分支，输出 e1；k=2 时，c='a'，输出 N；k=3，c='c'，输出 c2；k=4，c='h'，输出 N。故程序的输出结果为 Ne1Nc2N。答案选择 C 选项。

64．有以下程序：

```
#include <stdio.h>
main()
{
   char c;
   while((c=getchar())!='\n')
   {
      switch(c-'2')
      {
         case 0:
         case 1:putchar(c+4);
         case 2:putchar(c+4);break;
         case 3:putchar(c+3);
         default:putchar(c+2);break;
      }
   }
   printf("\n");
}
```

程序运行后从第一列开始输入以下数据：

2473<回车>

程序的输出结果是（　　）。

A．668977

B．4444

C．6677877

D．68766

【答案】A

【解析】本题执行过程为：读入 c='2'，c-'2'=0，首先匹配 case0，依次输出 6，6，后执行 break 语句，跳出分支结构；读入 c='4'，c-'2'=2，匹配 case2，输出 8，执行 break 语句，跳出分支结构；读入 c='7'，c-'2'=5，匹配 default，输出 9，执行 break 语句，跳出分支结构；读入 c='3'，c-'2'=1，匹配 case1，依次输出 7，7，执行 break 语句，跳出分支结构。输入回车，结束循环。答案选择 A 选项。

65．有以下程序

```
#include<stdio.h>
main()
{
   int s=0,n;
   for(n=0;n<3;n++)
   {
      switch(s)
      {
         case 0:
         case 1:s+=1;
         case 2:s+=2;break;
         case 3:s+=3;
         default:s+=4;
      }
      printf("%d,",s);
   }
}
```

程序运行后的输出结果是（　　）。

A. 1,2,4,

B. 1,3,6,

C. 3,10,14,

D. 3,6,10,

【答案】C

【解析】第一次循环：s=0,n=0,执行 switch 语句中的 case 1 和 case 2 语句，输出 3；
第二次循环：s=3,n=1,执行 switch 语句中的 case 3 和 default 语句，输出 10；
第三次循环：s=10,n=2,执行 switch 语句中的 default 语句，输出 14。答案选择 C 选项。

66．有以下程序：

```
#include <stdio.h>
main()
{
   int a,b;
   for(a=0; a<3; a++)
   {
      scanf("%d", &b);
      switch(b)
      {
         default: printf("%d,", ++b);
         case 1: printf("%d,", ++b);
         case 2: printf("%d,", ++b);
      }
   }
}
```

执行时输入：1 2 3<回车>，则输出结果是（　　）。

A. 2,3,3,4,5,6,

B. 2,3,4,

C. 2,2,3,4,4,4,

D. 2,3,4,3,4,4,

【答案】A

【解析】考查 C 语言中的 switch-case 语句。执行 switch-case 语句时，一定会先进行匹配，匹配成功则执行当前 case 语句，再根据是否有 break，判断是否继续输出，或是跳出判断。程序中输入 1 时，与 case 1 匹配成功，执行后面的输出语句，输出 2，且此时 b=2，因为后面没有 break 语句，继续执行下一条输出语句，输出 3。继续输入 2 时，与 case 2 匹配成功，执行后面的输出语句，输出 3。输入 3 时，没有可匹配的 case 语句，执行 default 语句，输出 4，b 变为 4；后面没有 break 语句，继续执行 case 1 后的输出语句，输出 5，b 变为 5；同理继续执行 case2 后面的输出语句，输出 6。A 选项正确。

67．有以下程序：

```
#include <stdio.h>
main()
{
   int i=1,k=0;
   for(;i<6;)
   {
      switch(i%3)
      {
         case 0:k++;
         case 1:k++;break;
         case 2:k++;continue;
      }
      i+=1;
   }
   printf("%d\n",k);
}
```

程序的运行情况是（　　）。

A．形成无限循环

B．输出 6

C．输出 5

D．输出 4

【答案】A

【解析】本题执行过程为：i=1，k=0，i<6 成立，执行 for 循环：i%3=1，匹配 case1，k=1，退出 switch，i=2；i<6 成立，执行 for 循环：i%3=2，匹配 case2，k=2，执行 continue，退出 switch，且不执行 i+=1，i=2 不变，i<6 成立，执行下一次 for 循环，以后的循环情况完全一致，i=2 不会改变，形成无限循环。答案选择 A 选项。

68．有以下程序：

```
#include <stdio.h>
main()
{
   int s;
   scanf("%d",&s);
   while(s>0)
   {
      switch(s)
      {
         case 1:printf("%d",s+5);
         case 2:printf("%d",s+4);break;
         case 3:printf("%d",s+3);
         default:printf("%d",s+1);break;
      }
      scanf("%d",&s);
   }
}
```

运行时，若输入 1 2 3 4 5 0<回车>，则输出结果是（　　）。

A. 6566456

B. 66656

C. 66666

D. 6666656

【答案】A

【解析】输入 1：执行 case1，输出 6，没有遇到 break，继续执行 case2，输出 5，遇到 break，跳出；
输入 2：执行 case2，输出 6，遇到 break，跳出；
输入 3：执行 case3，输出 6，没有遇到 break，执行 default，输出 4；
输入 4：执行 default，输出 5；
输入 5：执行 default，输出 6。
最后输出结果是 6566456。答案选择 A 选项。

69. 有以下程序：

```
int i,n;
for(i=0; i<8; i++)
{
   n=rand()%5;
   switch(n)
   {
      case 1:
      case 3:printf("%d\n",n);break;
      case 2:
      case 4:printf("%d\n",n);continue;
      case 0:exit(0);
   }
   printf("%d\n",n);
}
```

以下关于程序执行情况的叙述中，正确的是（　　）。

A. for 循环语句固定执行 8 次

B. 当产生的随机数 n 为 4 时结束循环操作

C．当产生的随机数 n 为 1 和 2 时不做任何操作

D．当产生的随机数 n 为 0 时结束程序运行

【答案】D

【解析】当产生随机数为 1 或 3 时，会顺序执行 case1 或 case2 下面的语句，进而输出结果；当产生随机数为 2 或 4 时，会继续执行循环；当产生随机数为 0 时，正常结束程序的运行。for 循环语句随着产生的随机数的不同，执行的次数也不相同。答案选择 D 选项。

70．有以下程序：

```
#include<stdio.h>
main()
{
   int a[]={2,3,5,4},i;
   for(i=0;i<4;i++)
      switch(i%2)
      {
         case 0:switch(a[i]%2)
                {
                   case 0:a[i]++;break;
                   case 1:a[i]--;
                 } break;
         case 1:a[i]=0;
      }
   for(i=0;i<4;i++) printf("%d",a[i]);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．3344

B．2050

C．3040

D．0304

【答案】C

【解析】main 函数的主体是一个 for 循环语句，for 循环中包含一个 switch 语句，如果判断条件为 0 则进入第二个 switch 语句，如果判断语句为 1 则执行“a[i]=0”，最后将数组顺序输出。所以程序功能是将数组下标为奇数的项设为 0，数组下标为偶数的项，如果对应的元素为偶数则加 1，如果对应的元素为奇数则减 1。答案选择 C 选项。

71．有以下程序：

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

程序的运行结果是（　　）。

A. a=1,b=0

B. a=2,b=2

C. a=1,b=1

D. a=2,b=1

【答案】D

【解析】当执行 swicth 语句时，首先计算紧跟其后一对括号中的表达式的值，然后在 switch 语句体内寻找与该值吻合的 case 标号。如果有与该值相等的标号，则执行该标号后开始的各语句，包括在其后的所有 case 和 default 中的语句，直到 switch 语句体结束；每当执行到 break 语句时，立即跳出 switch 语句体。switch 语句通常总是和 break 语句联合使用，使得 switch 语句真正起到分支的作用。本题中，x=1，首先进入外层分支 1，接下来 y=0，再进入内层分支 0，执行 a++，此时 a=1，接下来遇到 break，跳出内层 switch；由于外层分支中没有 break 语句，会继续执行外层分支 2，执行 a++，b++，此时 a=2，b=1；然后遇到 break 语句，跳出外层 switch。答案选择 D 选项。

72．有以下程序

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

程序的运行结果是（　　）。

A. a=2,b=1

B. a=2,b=2

C. a=1,b=1

D. a=1,b=0

【答案】A

【解析】当执行 swicth 语句时，首先计算紧跟其后一对括号中的表达式的值，然后在 switch 语句体内寻找与该值吻合的 case 标号。如果有与该值相等的标号，则执行该标号后开始的各语句，包括在其后的所有 case 和 default 中的语句，直到 switch 语句体结束；每当执行到 break 语句时，立即跳出 switch 语句体。switch 语句通常总是和 break 语句联合使用，使得 switch 语句真正起到分支的作用。本题中，x=1，首先进入外层分支 1，接下来 y=0，再进入内层分支 0，执行 a++，此时 a=1，接下来遇到 break，跳出内层 switch；由于外层分支中没有 break 语句，会继续执行外层分支 2，执行 a++，b++，此时 a=2，b=1；然后遇到 break 语句，跳出外层 switch。答案选择 A 选项。

73. 若有以下程序

```
#include <stdio.h>
main()
{
   int s=0,n;
   for(n=0;n<4;n++)
   {
      switch(n)
      {
         default:s+=4;
         case 1:s+=1;break;
         case 2:s+=2;break;
         case 3:s+=3;
      }
   }
   printf("%d\n",s);
}
```

则程序的输出结果是（　　）。

A．13

B．10

C．11

D．15

【答案】C

【解析】default 也是关键字，起标号的作用，代表所有 case 标号之外的那些标号。default 标号可以出现在语句体中任何标号位置上。在 switch 语句体中也可以没有 default 标号。

先判定 case 语句，如果没有与判定值相等的标号，并且存在 default 标号，则从 default 标号后的语句开始执行。如果 default 语句没有 break，则其下面的 case 语句不加判定的继续执行（default 位置在 case 前面时），直到遇到 break 或 switch 结尾。

①n=0，执行 default，s=4，没有遇到 break，不与 case 标号比较，直接执行 case 后的语句，s=5，遇到 break，跳出循环；

②n=1，s+=1，s=6；

③n=2，s+=2，s=8；

④n=3，s+=3，s=11。

答案选择 C 选项。

74．有以下程序

```
#include <stdio.h>
main()
{
   int k=5,n=0;
   do
   {
      switch(k)
      {
         case 1:case 3:n+=1;k--;break;
         default:n=0;k--;
         case 2:case 4:n+=2;k--;break;
      }
      printf("%d",n);
   }while(k>0&&n<5);
}
```

程序运行后的输出结果是（　　）。

A．235

B．0235

C．02356

D．2356

【答案】A

【解析】第一次循环 k 为 5，执行“n=0;k--;”和“n+=2;k--;break;”输出 n 为 2，k 的值为 3，第二次循环 k 为 3，执行“n+=1;k--,break;”输出 n 为 3，k 的值为 2，第三次循环 k 为 2，执行“n+=2;k--;break;”，输出 n 为 5，k 的值为 1，循环条件不成立，所以输出 235。答案选择 A 选项。

75．有以下程序：

```
#include <stdio.h>
main()
{
   char *s="120119110";
   int n0,n1,n2,nn,i;
   n0=n1=n2=nn=i=0;
   do
   {
      switch(s[i++])
      {
         default:nn++;
         case '0':n0++;
         case '1':n1++;
         case '2':n2++;
      }
   }while(s[i]);
   printf("n0=%d,n1=%d,n2=%d,nn=%d\n",n0,n1,n2,nn);
}
```

程序的运行结果是（　　）。

A. n0=3,n1=8,n2=9,nn=1

B. n0=2,n1=5,n2=1,nn=1

C. n0=2,n1=7,n2=10,nn=1

D. n0=4,n1=8,n2=9,nn=1

【答案】A

【解析】本题执行过程为：s[0]='1'，匹配 case '1'，n1=1，n2=1；s[1]='2'，匹配 case '2'，n2=2；s[2]='0'，匹配 case '0'，n0=1，n1=2，n2=3；s[3]='1'，匹配 case '1'，n1=3，n2=4；s[4]='1'，匹配 case '1'，n1=4，n2=5；s[5]='9'，匹配 default，nn=1，n0=2，n1=5，n2=6；s[6]='1'，匹配 case '1'，n1=6，n2=7；s[7]='1'，匹配 case '1'，n1=7，n2=8；s[8]='0'，匹配 case '0'，n0=3，n1=8，n2=9；s[9]='\0'，退出循环。输出 n0，n1，n2，nn 为 3，8，9，1。答案选择 A 选项。

二、填空题

给定程序中，函数 fun 的功能是：将形参 n 所指变量中，各位上为偶数的数去除，剩余的数按原来从高位到低位的顺序组成一个新的数，并通过形参指针 n 传回所指变量。例如，输入一个数 27638496，新的数为 739。

请在程序的下划线处填入正确的内容并把下划线删除，使程序得出正确的结果。

注意：源程序存放在考生文件夹下的 BLANK1.C 中。

不得增行或删行，也不得更改程序的结构！

试题程序如下：

```
#include <stdio.h>
void fun(unsigned long *n)
{
   unsigned long x=0, i;
   int t;
   i=1;
   while(*n)
   {
      /**********found**********/
      t=*n % ①_____;
      /**********found**********/
      if(t%2!= ②_____)
      {
         x=x+t*i;
         i=i*10;
      }
      *n =*n/10;
   }
   /**********found**********/
   *n=③_____;
}
main()
{
   unsigned long n=-1;
   while(n>99999999||n<0)
   {
      printf("Please input(0<n<100000000): ");
      scanf("%ld",&n);
    }
   fun(&n);
   printf("\nThe result is: %ld\n",n);
}
```

【答案】

①10

②0

③x

【解析】

填空 1：通过 t 对 10 求余，取出该数值的个位上的数。

填空 2：通过 if 条件语句实现奇偶数的判定。如果条件表达式对 2 求余为 0 即是偶数，反之是奇数。

填空 3：最后将剩余的数赋给 n 指向的元素。

三、改错题

下列给定程序中，函数 fun 的功能是：按以下递归公式求函数的值。

$$fun(n)=\begin{cases}10 & (n=1)\\ fun(n-1)+2 & (n>1)\end{cases}$$

例如，当给 n 输入 5 时，函数值为 18；当给 n 输入 3 时，函数值为 14。请改正程序中的错误，使它能得出正确的结果。

注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构！

试题程序如下：

```
#include <stdio.h>
/*********found*********/
fun(n)
{
   int c;
   /*********found*********/
   if(n=1)
      c=10;
   else
      c=fun(n-1)+2;
   return(c);
}
main()
{
   int n;
   printf("Enter n:\n");
   scanf("%d",&n);
   printf("The result is:%d\n\n",fun(n));
}
```

答：

（1）错误：fun(n)

正确：int fun(int n)

（2）错误：if(n=1)

正确：if(n==1)

【解析】

错误 1：有参数传入的函数要定义形参变量类型，由于主函数中传递的参数为 int 型，所以形参定义为 int 型，由 fun 函数的返回值可知，函数返回类型为 int。

错误 2：n=1 是赋值表达式，不能作为判断条件，==用于比较判断。

# 第 5 章　循环结构程序设计

一、选择题

1．在以下给出的表达式中，与 while(E)中的(E)不等价的表达式是（　　）。

A．(!E==0)

B．(E>0||E<0)

C．(E==0)

D．(E!=0)

【答案】C

【解析】ABD 三项都是 E 不等于零时执行 while 循环，C 项是 E 等于零时执行 while 循环。答案选择 C 选项。

2．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．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);

【答案】A

【解析】A 项正确，“(ch=getchar())!='N'”表示先把 getchar()函数的返回值赋值给 ch，然后判断，当读入字母不为 N 时，ch 与'N'不相等，执行打印函数 printf，当读入字母 N 时，ch 与'N'相等，循环结束，同理知选项 D 错误。B 项错误，“ch=getchar()='N'”表达式从右向左计算，会把 N 赋值给 getchar()函数的返回值，而 getchar()函数的返回值是右值，不可改变，出现编译错误；C 项错误，“ch=getchar()=='N'”中“==（等于）”的优先级高于“=（赋值）”，它等价于“ch=(getchar()=='N')”，此时 ch 的取值是 0 或 1。答案选择 A 选项。

3．对于“while(!E)s;”，若要执行循环体 s，则 E 的取值应为（　　）。

A．E 等于 1

B．E 不等于 0

C．E 不等于 1

D．E 等于 0

【答案】D

【解析】如果执行循环体，要求!E 成立，则 E 等于 0。答案选择 D 选项。

4．关于“do{循环体}while(条件表达式)”，以下叙述正确的是（　　）。

A．循环体的执行次数总是比条件表达式的执行次数多一次

B．条件表达式的执行次数总是比循环体的执行次数多一次

C．条件表达式的执行次数与循环体的执行次数一样

D．条件表达式的执行次数与循环体的执行次数无关

【答案】D

【解析】考查 do-while 循环语句。其执行过程为：先执行 do 循环体语句，然后判定圆括号内的表达式，如果为真则继续执行 do 循环体语句，如果为假则结束循环。考虑情况一：即由于条件表达式为假而退出循环，此时二者执行次数一样；情况二：即在循环体中执行了 break 语句而退出循环，此时条件表达式的执行次数比循环体的执行次数少一次。因此，答案选择 D 选项。

5．有以下程序：

```
#include <stdio.h>
main()
{
   …
   while(getchar()!='\n');
   …
}
```

以下叙述中正确的是（　　）。

A. 此 while 语句将无限循环

B. getchar()不可以出现在 while 语句的条件表达式中

C. 当执行此 while 语句时，只有按回车键程序才能继续执行

D. 当执行此 while 语句时，按任意键程序就能继续执行

【答案】C

【解析】本题中 while 循环条件为 getchar()!='\n'，表示只要不输入回车键，getchar()!='\n'语句一直为真，则 while 循环会出现空循环，当按下回车键后跳出 while 循环执行下一条语句。答案选择 C 选项。

6．若要实现 total=1+2+3+4+5 求和，以下程序段错误的是（　　）。

A.

```
int i=1,total=1;
while(i<5)
{
   total +=i;
   i+=1;
}
```

B.

```
int i=1,total=0;
while(i<=5)
{
   total +=i;
   i+=1;
}
```

C.

```
int i=0,total=0;
while(i<5)
{
   i +=1;
   total+=i;
}
```

D.

```
int i=0,total=0;
while(i<=5)
{
   total +=i;
   i+=1;
}
```

【答案】A

【解析】A 项执行过程为：i=1，total=1，判断 while 循环条件 1<5 成立，total=1+1，i=2；判断循环条件执行函数体，直到 i=4，total=1+1+2+3+4，i=5；i=5 循环条件不成立，退出循环，A 项不能实现题目要求，错误。B 项执行过程：i=1，total=0，1<=5 成立，total=0+1=1，i=2；直到 i=5，total=1+2+3+4+5，i=6；i=6 条件不成立，退出循环，程序正确。C 项执行过程：i=0，total=0，0<5 成立，i=1，total=1；直到 i=5，total=1+2+3+4+5；i=5 条件不成立，退出循环，程序正确。D 项执行过程：i=0，total=0，0<=5 成立，total=0，i=1；直到 i=5，total=1+2+3+4+5，i=6；i=6 条件不成立，退出循环，程序正确。答案选择 A 选项。

7．有以下程序

```
#include<stdio.h>
main()
{
   int y=10;
   while(y--);
   printf("y=%d\n",y);
}
```

程序执行后的输出结果是（　　）。

A．y=0

B．y=-1

C．y=1

D．while 构成无限循环

【答案】B

【解析】本程序的功能是当循环条件表达式 y--的值为 0（即逻辑假）时，跳出循环，执行后面的输出语句，且每循环一次变量 y 的值自减 1。当执行第 11 次循环时，y=0，表达式 y--的值为 0，循环条件为“假”，退出循环，而此时变量 y 的值经自减运算后变为-1。因此，输出结果为-1。答案选择 B 选项。

8．有以下程序

```
#include<stdio.h>
main()
{
   int k=5;
   while(--k)printf("%d\n",k-=3);
   printf("\n");
}
```

执行后的输出结果是（　　）。

A．1

B．2

C．4

D．死循环

【答案】A

【解析】“while(--k)”是先执行 k 减 1，然后判定 k 是否等于 0；“printf("%d\n",k-=3);”是先执行 k-=3，然后输出 k 的值。k 的初始值是 5，--k 后 k 变成 4，进入循环体内部，首先执行 k-=3，得到 k 等于 1，输出 1；接下来继续执行--k，k 变成 0，不满足条件，循环结束。答案选择 A 选项。

9．有以下程序：

```
#include <stdio.h>
main()
{
   int a=7;
   while(a--);
   printf("%d\n", a);
}
```

程序运行后的输出结果是（　　）。

A. -1

B. 0

C. 1

D. 7

【答案】A

【解析】“++”和“--”运算，当以前缀形式出现时，则先进行加一或减一操作，再取值，当以后缀形式出现时，则先取值，再进行加一或减一操作。程序中执行 a--，直到 while 判断为 0 时才跳出循环，执行下条语句，即 a 为 0 时再执行 a--，此时跳出 while 循环，最终输出的结果为-1。答案选择 A 选项。

10．有以下程序：

```
#include<stdio.h>
main()
{
   char *s="12134";
   int k=0,a=0;
   while(s[k+1]!='\0')
   {
      k++;
      if(k%2==0)
      {
         a=a+s[k]-'0'+1;
         continue;
      }
      a=a+(s[k]-'0');
   }
   printf("k=%d a=%d\n",k,a);
}
```

程序运行后的输出结果是（　　）。

A. k=6 a=11

B. k=3 a=14

C. k=4 a=12

D. k=5 a=15

【答案】C

【解析】第一次循环，k=1，s[1]=2，执行 a=a+(s[k]-'0')=2；第二次循环，k=2，s[2]=1，执行 a=a+(s[k]-'0'+1)=2+2=4；第三次循环，k=3，s[3]=3，执行 a=a+(s[k]-'0')=4+3=7；第四次循环，k=4，s[4]=4，执行 a=a+(s[k]-'0'+1)=7+5=12，故最终输出 k=4，a=12。答案选择 C 选项。

11．有以下程序：

```
#include<stdio.h>
main()
{
   int a=1,b=2;
   while(a<6)
   {
      b+=a;
      a+=2;
      b%=10;
   }
   printf("%d,%d\n",a,b);
}
```

程序运行后的输出结果是（　　）。

A. 5,11

B. 7,1

C. 7,11

D. 6,1

【答案】B

【解析】第一次循环，a=1，b=b+a=3，a=a+2=3，b=b%10=3；第二次循环，a=3，b=b+a=6，a=a+2=5，b=b%10=6；第三次循环，a=5，b=b+a=11，a=a+2=7，b=b%10=1；结束循环。答案选择 B 选项。

12．若有以下程序

```
#include <stdio.h>
main()
{
   int a=-2,b=0;
   while(a++) ++b;
   printf("%d,%d\n",a,b);
}
```

则程序的输出结果是（　　）。

A. 1,2

B. 0,2

C. 1,3

D. 2,3

【答案】A

【解析】while(a++)是先判定 a 是否等于 0，如果 a 等于 0，跳出循环；否则进入循环；但是不管判定是否成功，判定结束后都要执行 a++操作。第一次循环，a 等于-2，满足条件，执行 a++，++b，a 变成-1，b 变成 1；第二次循环，a 等于-1，满足条件，执行 a++，++b，a 变成 0，b 变成 2；第三次循环，a 等于 0，不满足条件，执行 a++，a 变成 1，b 仍然是 2。答案选择 A 选项。

13．有以下程序：

```
#include<stdio.h>
main()
{
   int a=-2,b=0;
   while(a++&&++b);
   printf("%d,%d\n",a,b);
}
```

程序运行后的输出结果是（　　）。

A．1,3

B．0,2

C．0,3

D．1,2

【答案】D

【解析】第一次循环之后 a=-1，b=1，第二次循环之后，a=0，b=2，第三次循环时，由于 a++表示 a 加之前的值，即 a++=0，逻辑与值为 0，所以不再执行++b，这时，a 自增 1，由 0 变为 1，而 b 的值仍为 2。答案选择 D 选项。

14．以下叙述正确的是（　　）。

A．do-while 语句构成的循环，当 while 语句中的表达式值为 0 时结束循环

B．do-while 语句和 while-do 构成的循环功能相同

C．while-do 语句构成的循环，当 while 语句中的表达式值为非 0 时结束循环

D．do-while 语句构成的循环，必须用 break 语句退出循环

【答案】A

【解析】B 项错误，do-while 语句先执行循环体，再判断循环条件语句，while-do 循环先判断循环条件语句，再执行循环体；C 项错误，do-while 语句构成的循环，while 语句中的表达式值为 0 时结束循环；D 项错误，do-while 语句除了可以使用 break 语句退出循环外，还可以使用循环条件语句，当不满足循环条件时退出循环。答案选择 A 选项。

15．以下程序段中，循环次数不超过 10 的是（　　）。

A．int i=10;do{i=i+1;}while(i<0);

B．int i=5;do{i+=1;}while(i>0);

C．int i=1;do{i+=2;}while(i!=10);

D．int i=6;do{i-=2;}while(i!=1);

【答案】A

【解析】A 项，执行循环体 i=i+1，判断条件 i<0 不成立，退出循环，循环次数为 1，A 项正确。B 项执行一次循环体后 i=6，判断条件为 i>0，由于执行循环体时 i 始终在增加，i>0 一直成立，程序陷入无限循环，循环次数大于 10，B 项错误。C 项执行循环体一次后 i=3，判断条件为 i!=10，由于循环体中 i 每次增加 2，所以永远不可能等于 10，故程序会陷入无限循环，循环次数大于 10，C 项错误。D 项执行循环体一次后 i=4，判断条件为 i!=1，由于循环体中 i 每次减少 2，所以永远不可能等于 1，故程序会陷入无限循环，循环次数大于 10，D 项错误。答案选择 A 选项。

16．以下能够实现计算 5!的程序段是（　　）。

A．int fac=1,k=0; do{k++;fac*=k;}while(k<5);

B．int fac=0,k=1; do{fac*=k;k++;}while(k<5);

C．int fac=1,k=1; do{k++;fac*=k;}while(k<=5);

D．int fac=1,k=0; do{fac*=k;k++;}while(k<5);

【答案】A

【解析】do…while 循环语句一般形式为：do{循环体}while(表达式)，执行过程为：首先执行循环体，之后判断表达式，表达式为真，则再一次执行循环体，否则退出循环。A 项：k=1，fac=1*1，判断 k<5，条件成立进

行下一次循环，直到 k=5，fac=1*2*3*4*5，k<5 时退出循环，实现计算 5!。B 项：fac 从 0 开始，做乘法一直都是 0，无法实现 5!。C 项：k=2，fac=1*2，循环条件成立，直到 k=6，fac=1*2*3*4*5*6 才因循环条件不成立退出循环，实现 6!。D 项：k=0，fac=1*0=0，之后做乘法一直为 0，无法实现 5!。答案选择 A 选项。

17．若变量已正确定义，有以下程序段

```
i=0;
do printf("%d,",i);
while(i++);
printf("%d\n",i);
```

其输出结果是（　　）。

A．0,1

B．0,0

C．1,1

D．程序进入无限循环

【答案】A

【解析】首先进入循环体，输出 0；while(i++)是先判定 i 是否为 0，不管判定结果如何，都执行 i++。i 等于 0，判定结果为 false，执行 i++，跳出 do-while 循环，然后输出 1。答案选择 A 选项。

18．若有以下程序

```
#include <stdio.h>
main()
{
   int a=-2,b=0;
   do
   {
      ++b;
   }while(a++);
   printf("%d,%d\n",a,b);
}
```

则程序的输出结果是（　　）。

A．2,3

B．0,2

C．1,2

D．1,3

【答案】D

【解析】while(a++)是先判定 a 是否为 0，不管判定结果如何，都执行 a++。当 a++的值为 0 时，a 的值为 1，即 a 增加了 3，因此，b 也增加 3。答案选择 D 选项。

19．有以下程序：

```
#include<stdio.h>
main()
{
   int x=23;
   do
   {
      printf("%2d\n",x--);
   }
   while(!x);
}
```

程序的执行结果是（　　）。

A. 输出 321

B. 输出 23

C. 不输出任何内容

D. 陷入无限循环

【答案】B

【解析】“%2d”表示按宽度为 2，右对齐方式输出，若不够两位，左边补空格。程序执行过程为：输出 x 为 23，之后 x 自减得 x=22，!x 为假，while 条件不成立，退出循环。此题需要注意 x--和--x 的区别，在逻辑表达式中，x--是先传递 x 的值，再执行自减 1，此题就是这种情况，--x 是先自减 1，再执行传递 x 的值。答案选择 B 选项。

20．有以下程序：

```
#include <stdio.h>
main()
{
   int i=0,sum=1;
   do
   {
      sum += i++;
   }while(i<6);
   printf("%d\n",sum);
}
```

程序的输出结果是（　　）。

A. 22

B. 18

C. 20

D. 16

【答案】D

【解析】语句 sum+=i++；相当于 sum+=i；i++；程序执行过程为：sum=1，i=1；sum=2，i=2；sum=4，i=3；sum=7，i=4；sum=11，i=5；sum=16，i=6；退出循环。答案选择 D 选项。

21．有以下程序：

```
#include<stdio.h>
main()
{
   int sum=0,x=5;
   do{sum+=x;}while(!--x);
   printf("%d\n",sum);
}
```

程序的运行结果是（　　）。

A．0

B．5

C．14

D．15

【答案】B

【解析】do-while 循环，先执行循环体 sum+=x，则 sum=sum+x=0+5=5，再执行 while 中的表达式，结果为 0，退出循环，所以运行结果是 5。答案选择 B 选项。

22．有以下程序（注：字符 a 的 ASCII 码值为 97）：

```
#include   <stdio.h>
main()
{
   char *s={"abc"};
   do
   {
      printf("%d",*s%10);
      ++s;
   }while(*s);
}
```

程序运行后的输出结果是（　　）。

A．abc

B．789

C．7890

D．979800

【答案】B

【解析】a、b、c 的 ASCII 值分别为 97、98、99。程序中执行输出 s 中字符对应的 ASCII 码与 10 进行模运算后的值，s 是一个指针，首先指向字符 a，先执行 97%10，结果为 7；然后++s，指针指向下一个字符 b，执行 98%10，结果为 8，直到 s 所指为空，故最后输出的结果为 789。答案选择 B 选项。

23．有以下程序：（注意：字母 A 的 ASCII 码值为 65。）

```
#include <stdio.h>
main()
{
   char *s={"ABC"};
   do
   {
      printf("%d",*( s++)%10);
   }while(*s);
}
```

程序运行后的结果是（　　）。

A．5670

B．656667

C．567

D．ABC

【答案】C

【解析】每进行一次循环，先将输出 s 指向字母的 ASCII 码值除以 10 的余数，然后 s 执行 s++。字符 A 的 ASCII 码值为 65，第一次输出 65%10 的余数 5，执行 s++后，指针指向字母 B；第二次输出 66%10 的余数 6，执行 s++后，指针指向字母 C；第三次输出 67%10 的余数 7，执行 s++后，指针指向串结束标志'\0'，'\0'对应 ASCII 值为 0，即*s 等于 0，退出循环。因此，输出结果为 567。答案选择 C 选项。

24．有以下程序：

```
#include <stdio.h>
main()
{
   char c;
   do
   {
      c = getchar();
      putchar(c);
   }while(c!='#');
   printf("\n");
}
```

执行时如输入：abcdefg##<回车>，则输出结果是（　　）。

A．abcdefg#

B．abcdefg

C．abcdefg##

D．##

【答案】A

【解析】程序执行过程为：读入一个字符存入缓存区，判断字符是否为“#”，如不是，继续读入字符，直到读入的字符是“#”，存入缓存区，退出循环，将缓冲区字符一个个输出，结果为 abcdefg#。答案选择 A 选项。

25．有以下程序：

```
#include<stdio.h>
main()
{
   int i=5;
   do
   {
      if(i%3==1)
      if(i%5==2)
      {
         printf("*%d",i);
         break;
      }
      i++;
   }while(i!=0);
   printf("\n");
}
```

程序运行的结果是（　　）。

A. *7

B. *3*5

C. *5

D. *2*6

【答案】A

【解析】在 do while 循环中，总是先执行循环体后判断循环条件，所以循环体至少会被执行一次。在循环中，如果满足(i%3==1)&&(i%5==2)，则输出 i 的值，并执行 break;退出 do while 循环；否则 i++，继续判断 while 条件，如果 i==0，则终止 do while 循环。第 1 遍循环，5%3==1 不成立，执行 i++，i 变为 6，第 1 遍循环结束，判断(i!=0)为真，继续循环。第 2 遍循环，6%3==1 不成立，执行 i++，i 变为 7，第 2 遍循环结束，判断(i!=0)为真，继续循环。第 3 遍循环 7%3==1 成立，7%3==2 成立，执行 if 子句“printf("*%d",i);break;”，输出*7 之后跳出循环。答案选择 A 选项。

26．有以下程序：

```
#include<stdio.h>
main()
{
   int x=0,y=6;
   do
   {
      while(--y)x++;
   }
   while(y--);
   printf("%d,%d\n",x,y);
}
```

程序的运行结果是（　　）。

A. 5,0

B. 6,0

C. 5,-1

D. 6,-1

【答案】C

【解析】程序执行过程为：执行 do…while 的循环体：y=5，判断 y 为真，x=1；y=4，x=2；y=3，x=3；y=2，

x=4；y=1，x=5；y=0，判断 y 为假，退出循环体。判断 do…while 条件 y=0 为假，y=-1，退出循环。输出 x,y 为 5,-1，答案选择 C 选项。

27．以下叙述中正确的是（　　）。

A．如果根据算法需要使用无限循环（即通常所称的“死循环”），则只能使用 while 语句

B．对于“for(表达式 1;表达式 2;表达式 3)循环体”首先要计算表达式 2 的值，以便决定是否开始循环

C．对于“for(表达式 1;表达式 2;表达式 3)循环体”，只在个别情况下才能转换成 while 语句

D．只要适当地修改代码，就可以将 do-while 与 while 相互转换

【答案】D

【解析】D 项正确，C 语言中 do-while 语句和 while 语句作用是等价的，二者可以经过适当的修改互换。A 项错误，do-while 和 for 循环也能写成死循环；B 项错误，for 循环首先执行表达式 1；C 项错误，for 循环体经过适当的修改都可以转换成 while 语句。答案选择 D 选项。

28．若变量已正确定义

for(x=0,y=0; (y!=99&&x<4); x++)

则以上 for 循环（　　）。

A．执行无限次

B．执行 3 次

C．执行 4 次

D．执行次数不定

【答案】C

【解析】y!=99 始终是 true，没有起到作用；x 经过 4 次循环后变成 4，不满足 x<4 的条件，跳出循环。答案选择 C 选项。

29．若变量已正确定义，则以下 for 循环（　　）：

for(x=0,y=0; (y!=123)&&(x<4); x++);

A．执行 4 次

B．执行 3 次

C．执行次数不确定

D．执行 123 次

【答案】A

【解析】本题考查 for 语句。for 语句执行的次数与判定条件成立的次数相同。本题中，y!=123 始终为 true，x 经过四次循环后变成 4，不满足 x<4 的条件，跳出循环。由此可知执行 4 次。答案选择 A 选项。

30．若 k 是 int 类型变量，且有以下 for 语句：

for(k=-1;k<0;k++) printf("****\n");

下面关于语句执行情况的叙述中正确的是（　　）。

A．循环体执行一次

B．循环体执行两次

C．循环体一次也不执行

D．构成无限循环

【答案】A

【解析】for 语句中，由变量 k 控制循环体的执行，用来输出星号。第一次循环，k=-1，满足条件 k<0，执行循环体，进入下一次循环。此时 k=0，不满足循环条件 k<0，退出循环。因此，循环体仅执行了一次。答案选择 A 选项。

31．若 i 和 k 都是 int 类型变量，有以下 for 语句：

for=(i=0,k=-1;k=1;k++) printf("* * * * *\n");

下面关于语句执行情况的叙述中正确的是（　　）。

A．循环体执行两次

B. 循环体执行一次
C. 循环体一次也不执行
D. 构成无限循环
【答案】D
【解析】本题中 for 循环判断条件为 k=1，这个语句是赋值语句，总是正确的，所以会陷入无限循环中。答案选择 D 选项。

32. 有如下程序段：

```
int k;
for(k=2;k==0;
   printf("%d",k--);
```

则 for 循环体执行的次数是（　　）。
A. 0 次
B. 1 次
C. 2 次
D. 无限次
【答案】A
【解析】“for(k=2;k==0;)”表示给 k 赋值 2，如果 k 等于 0，则进入循环，但是 k 不满足条件，因此循环体不执行。答案选择 A 选项。

33. 以下不构成无限循环的语句或者语句组是（　　）。
A. n=0; do{++n;}while(n<=0);
B. n=0; while(1){n++;}
C. n=10; while(n); {n--;}
D. for(n=0,i=1;i++) n+=i;
【答案】A
【解析】A 项，为 do-while 循环语句，首先执行 do 后面的语句++n；得 n=1，while 条件表达式为假，退出循环；B 项，while 条件表达式的值始终为 1，条件为真，构成无限循环；C 项，while(n);语句循环体为空，n 的值在循环中一直保持不变，构成无限循环；D 项，i=1，for 语句中条件判断语句为空，永远为真，构成无限循环。答案选择 A 选项。

34. 以下程序段中，与其他三个功能不同的程序段是（　　）。
A. s=0; i=1; for(;;){s+=i; i++; if(i<=10) break;}
B. s=0,i=1; for(;i<=10;){s+=i; i++;}
C. s=0; for(i=1;i<=10;i++){s+=i; }
D. for(s=0,i=1;i<=10;s+=i,i++);
【答案】A
【解析】B、C、D 项都表示对 1 到 10 进行累加后赋给 s，而 A 项中含 break 语句，循环体只执行一次，结束整个循环过程。答案选择 A 选项。

35. 有以下程序：

```
#include<stdio.h>
main()
{
   int i=4;
   for(printf("%d",i);i<2;i++) printf("%d",i);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．12

B．4

C．1

D．0

【答案】B

【解析】for(printf("%d",i);i<2;i++)先输出 i，由于 i=4，不满足条件，跳出循环，直接输出换行符。答案选择 B 选项。

36．有以下程序：

```
#include <stdio.h>
main()
{
   int i,a;
   for(i=0;i<=10;i++)a=i;
   printf("%d,%d\n",i,a);
}
```

程序的运行结果是（　　）。

A．11,10

B．10,10

C．10,11

D．11,11

【答案】A

【解析】当 i=10 时循环继续执行，i 值赋值给变量 a。赋值完成后执行 i=i+1，此时 i=11，不满足循环条件，结束循环。然后输出 i，a 的值分别为 11，10。答案选择 A 选项。

37．有如下程序：

```
#include <stdio.h>
main()
{
   int i = 1;
   for(printf("%d",i);i<4;i++)
      printf("%d",i);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．1123

B．123

C．0123

D．001

【答案】A

【解析】for 语句的一般形式为“for(表达式 1;表达式 2;表达式 3){循环体语句}”，其执行流程是先计算表达式 1，之后计算表达式 2，根据表达式 2 的值来判定是否进行循环，若为真，则执行循环体，循环体执行完毕之后再计算表达式 3 的值。整个循环过程中，表达式 1 只会执行一次，而表达式 2、3 以及循环体每次循环都会执行，答案选择 A 选项。

38．以下函数的功能是计算 a 的 n 次方作为函数值返回：

```
double fun(double a,int n)
{
   int i;
   double s=1.0;
   for(i=1;i<=n;i++)s=_____;
   return s;
}
```

为实现上述功能，函数中下划线处应填入的是（　　）。

A. s*i

B. s*a

C. s+i*i

D. s+a*a

【答案】B

【解析】for 循环语句，i=1 时，s=s*a；i=2 时，s=s*a*a=s*$a^2$；i=n 时，s=s*$a^n$。最后返回 s。答案选择 B 选项。

39. 有以下程序

```
#include<stdio.h>
main()
{
   int a=1,b=2;
   for(;a<8;a++)
   {
      b+=a;
      a+=2;
   }
   printf("%d,%d\n",a,b);
}
```

程序运行后的输出结果是（　　）。

A. 9,18

B. 8,11

C. 7,11

D. 10,14

【答案】D

【解析】初始值 a=1，b=2，第一次循环：b=b+a=2+1=3，a=a+2=1+2=3，a=a+1=3+1=4；第二次循环：b=b+a=3+4=7，a=a+2=4+2=6，a=a+1=6+1=7；第三次循环：b=b+a=7+7=14，a=a+2=7+2=9，a=a+1=9+1=10，不满足 for 循环条件退出循环，最终 a=10，b=14。答案选择 D 选项。

40. 有以下程序：

```
#include <stdio.h>
main()
{
   char a[5][10] = {"one", "two", "three",  "four", "five"};
   int i,j;
   char t;
   for(i=0;i<4;i++)
      for(j=i+1;j<5;j++)
      {
         t=a[i][0];
         a[i][0]=a[j][0];
         a[j][0]=t;
      }
   puts(a[1]);
}
```

程序运行后的输出结果是（　　）。

A. fwo

B. fix

C. two

D. owo

【答案】A

【解析】for 循环实现的功能是将二维数组的第一列字母从小到大排序。第一列排完之后为 f，f，o，t，t，输出 a[1]即输出数组的第二行，即为 fwo。答案选择 A 选项。

41. 有以下程序

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

程序运行后的输出结果是（　　）。

A. 6

B. 10

C. 11

D. 15

【答案】C

【解析】当 i=0 时 a[b[0]]=1；当 i=1 时 a[b[1]]=3；当 i=2 时 a[b[2]]=2；当 i=3 时 a[b[3]]=4；当 i=4 时 a[b[4]]=1。故 s=1+3+2+4+1=11。答案选择 C 选项。

42. 以下函数 findmax 拟实现在数组中查找最大值并作为函数值返回，但程序中有错导致不能实现预定功能。

```
#define MIN -2147483647
int fingmax (int x[],int n)
{
   int i,max;
   for(i=0;i<n;i++)
   {
      max=MIN;
      if(max<x[i])max=x[i];
   }
   return max;
}
```

造成错误的原因是（　　）。

A．定义语句 int i,max;中 max 未赋初值

B．赋值语句 max=MIN;中，不应给 max 赋 MIN 值

C．语句 if(max<x[i]) max=x[i];中判断条件设置错误

D．赋值语句 max=MIN;放错了位置

【答案】D

【解析】本题中 for 循环中首先将 MIN 值赋值给 max，然后用 x[i]与 max 的值比较。每次都是将 MIN 值与 x[i]值进行比较，所以无论 x[i]的值是什么，都不会影响 if 的判断语句，max=x[i]始终执行。所以函数返回的是数组中最后一个元素的值。程序的错误在于 max=MIN 的位置，for 循环之前应先执行 max=MIN。所以答案选择 D 选项。

43．设变量已正确定义，以下不能统计出一行中输入字符个数（不包含回车符）的程序段是（　　）。

A．n=0; while((ch=getchar())!="\n") n++;

B．n=0; while(getchar()!="\n") n++;

C．for(n=0; getchar()!="\n"; n++);

D．n=0; for(ch=getchar()!="\n";n++);

【答案】D

【解析】要统计一行中输入字符个数（不包含回车符），首先定义一个用作统计的变量 n，赋初值为 0；一行字符是否结束的判断条件应为“getchar()!="\n"”；D 项中 for 循环表达式格式错误，应在 ch 前添加“;”。答案选择 D 选项。

44．有以下程序：

```
#include<stdio.h>
main()
{
   int a=1,b=0;
   for(;a<5;a++)
   {
      if(a%2==0)break;
      b+=a;
   }
   printf("%d\n",b);
}
```

程序的运行结果是（　　）。

A．1

B．10

C．0

D．4

【答案】A

【解析】程序的执行过程为：a=1 时，a%2=1，if 判定条件不成立，执行语句 b+=a；得到 b=1；a=2 时，a%2=0，if 判定条件成立，执行语句 break；跳出循环。输出 b=1，故答案选择 A 选项。

45．有以下程序：

```
#include<stdio.h>
main()
{
   int x=8;
   for(;x>0;x--)
   {
      if(x%3)
      {
         printf("%d,",x--);
         continue;
      }
      printf("%d,",--x);
   }
}
```

程序的运行结果是（　　）。

A．7,4,2,

B．8,7,5,2,

C．9,7,6,4,

D．8,5,4,2,

【答案】D

【解析】题目中的 for 循环等价于：

```
for(; x>0; x--)
{
   if(x%3 != 0)
   {
      printf("%d,", x);
      x--;
   }
   else
   {
      --x;
      printf("%d,", x);
   }
}
```

第一次循环，x=8，for 循环条件为真，8%3=2，不等于 0，则 if 条件表达式为真，执行第一个输出语句，先输出 x 的值 8，然后将 x 的值减 1，此时 x=7。然后执行 continue 语句结束本次循环，执行 x--表达式，得 x=6。

第二次循环，x=6，for 循环条件为真，6%3=0，则 if 条件表达式为假，执行第二个输出语句，先将 x 的值减 1 得 x=5，然后输出 x 的值 5，执行 x--表达式，得 x=4。

第三次循环，x=4，for 循环条件为真，4%3=1，不等于 0，则 if 条件表达式为真，执行第一个输出语句，先输出 x 的值 4，然后将 x 的值减 1，此时 x=3，然后执行 continue 语句结束本次循环，执行 x--表达式，得 x=2。

第四次循环，x=2，for 循环条件为真，2%3=2，不等于 0，则 if 条件表达式为真，执行第一个输出语句，先输出 x 的值 2，然后将 x 的值减 1，此时 x=1，执行 x--表达式，得 x=0，for 循环条件为假，循环结束。

答案选择 D 选项。

46．有以下程序

```
#include <stdio.h>
main()
{
   int y=9;
   for(;y>0;y--)
      if(y%3==0) printf("%d",--y);
}
```

程序的运行结果是（　　）。

A．852

B．963

C．741

D．875421

【答案】A

【解析】该程序的运行过程是 y 从 9 开始自减，每次判定 y 是否能被 3 整除，若是，则输出--y 后的值，能被 3 整除的 y 值分别为 9、6、3，对应输出 8、5、2，所以答案选择 A 选项。

47．有以下程序

```
#include <stdio.h>
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（　　）。

A．32

B．24

C．5

D．40

【答案】A

【解析】自增运算符“++”分为前缀和后缀两种形式。两种形式的作用效果是一样的，都是使运算分量的值加 1，但是它们的表达式的值不一样，前缀形式表达式的值为运算分量加 1 之后的值，而后缀形式表达式的值为运算分量加 1 之前的值。题目中使用了一个 for 循环，循环变量 i 从 1 递增到 40。在循环体中有两条嵌套的 if 语句，首先判断 i++%5==0，即判断 i++的值（i 加 1 之前的值）是否能被 5 整除（判断后 i 被加 1），然后再判断 ++i 的值（i 加 1 之后的值）是否能被 8 整除（判断后 i 被加 1），若两个条件都满足了，就输出 i 的值，只有 i=30 时，满足 i++%5==0，此时 i=31，++i%8==0 成立，此时 i=32。答案选择 A 选项。

48．有以下程序

```
#include<stdio.h>
main()
{
   int i,sum;
   for(i=1;i<6;i++)sum+=i;
   printf("%d\n",sum);
}
```

程序运行后的输出结果是（　　）。

A. 0

B. 随机值

C. 15

D. 16

【答案】B

【解析】sum 作为局部变量，没有显式初始化，sum 值代表原来内存中存储的对象，不可预知，程序结果是随机值。答案选择 B 选项。

49. 若有以下程序

```
#include<stdio.h>
main()
{
   int a=6,b=0,c=0;
   for(;a;)
   {
      b+=a;
      a-=++c;
   }
   printf("%d,%d,%d\n",a,b,c);
}
```

则程序的输出结果是（　　）。

A. 0,18,3

B. 1,14,3

C. 0,14,3

D. 0,14,6

【答案】C

【解析】题目中的 for 循环可以写成：

for(;a!=0;){b=b+a; ++c; a=a-c;}

第一次循环，a=6，进入 for 循环，b=0+6=6，c=1，a=6-1=5;

第二次循环，a=5，进入 for 循环，b=6+5=11，c=2，a=5-3=3;

第三次循环，a=3，进入 for 循环，b=11+3=14，c=3，a=3-3=0;

第四次循环，a=0，跳出 for 循环。答案选择 C 选项。

50. 有以下程序：

```
#include<stdio.h>
main()
{
   int i,j;
   for(i=3;i>=1;i--)
   {
      for(j=1;j<=2;j++) printf("%d",i+j);
      printf("\n");
   }
}
```

程序运行的结果是（　　）。

A. 234<换行>345<换行>

B. 432<换行>543<换行>45<换行>

C. 23<换行>34<换行>23<换行>

D. 45<换行>34<换行>23<换行>

【答案】D

【解析】外层主循环 i 从 3 减到 1 执行了 3 次，嵌套的循环 j 从 1 增到 2，每轮执行 2 次。每次输出 i+j 的值，就可以得到 3+1，3+2，2+1，2+2，1+1，1+2。但注意每次内循环结束要换行。答案选择 D 选项。

51．有以下程序

```
#include<stdio.h>
main()
{
   int i,j,m=1;
   for(i=1;i<3;i++)
   {
      for(j=3;j>0;j--)
      {
         if(i*j>3)break;
         m*=i*j;
      }
   }
   printf("m=%d\n",m);
}
```

程序运行后的输出结果是（　　）。

A. m=6

B. m=2

C. m=4

D. m=5

【答案】A

【解析】本程序中包含了一个双重 for 循环，当变量 i=1，且 j=3、j=2、j=1 时都能执行 m*=i*j，此时得到 m 的值为 6；当变量 i=2，且 j=3 时会执行 break 语句，内部循环直接结束，此时 i 再加 1，也会导致退出外部循环，所以最终 m 的值为 6。答案选择 A 选项。

52．有以下程序：

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

程序的运行结果是（　　）。

A. x=4

B. x=6

C. x=8

D. x=12

【答案】C

【解析】第一次循环：i=0，执行 x++后 x=1，执行内层循环 j=0，由于 j%2!=0 时才执行 x++，即只有 j 取 1 和 3 时，执行 x++，此时 x=1+1+1=3，跳出内层循环，执行下一条 x++，x=4，第二次循环：x=1 时，重复上述循环，共执行两次循环，故 x 最终结果是 4*2=8。答案选择 C 选项。注意，在循环体中遇到 continue，则跳过 continue 后的语句直接进入下一次循环的判断。

53. 有以下程序

```
#include<stdio.h>
main()
{
   int b[3][3]={0,1,2,0,1,2,0,1,2},i,j,t=1;
   for(i=0;i<3;i++)
      for(j=i;j<=i;j++)t+=b[i][b[j][i]];
   printf("%d\n",t);
}
```

程序运行后的输出结果是（　　）。

A. 1

B. 3

C. 4

D. 9

【答案】C

【解析】由 for(j=i;j<=i;j++)可知，内层循环变量 j 只可以取当前外层循环变量 i 的值。当 i=0，j=0 时，b[0][b[0][0]]=0；当 i=1，j=1 时，b[1][b[1][1]]=1；当 i=2，j=2 时，b[2][b[2][2]]=2。t 初始值为 1，故结果 t=1+0+1+2=4。答案选择 C 选项。

54. 以下程序段中的变量已正确定义：

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

```
for(k=1;k<3;k++);
printf("*");
```

程序段的输出结果是（　　）。

A. ******

B. ****

C. **

D. *

【答案】D

【解析】for 语句的循环体在没有“{}”说明时仅仅是跟随其后的第一条语句，本题中即另外一个 for 语句，第二个 for 循环有个“;”结束，所以整个双重循环将被视为空语句。然后执行下一个顺序语句，即输出语句，只输出一个“*”。答案选择 D 选项。

55．有以下程序

```
#include<stdio.h>
main()
{
   int i,j,m=55;
   for(i=1;i<=3;i++)
      for(j=3;j<=i;j++)m=m%j;
   printf("%d\n",m);
}
```

程序的运行结果是（　　）。

A. 1

B. 0

C. 2

D. 3

【答案】A

【解析】只有 i=3，j=3 时，内层循环的控制调价 j<=i 才为真，才能执行 m=m%j,即 55=55%3=1。答案选择 A 选项。

56．有以下程序

```
#include <stdio.h>
main()
{
   int i ,j;
   for(i=1;i<4;i++)
   {
      for(j=i;j<4;j++)
         printf("%d*%d=%d    ",i,j,i*j);
      printf("\n");
   }
}
```

程序运行后的输出结果是（　　）。

A.

```
1*1=1    1*2=2    1*3=3
2*2=4    2*3=6
3*3=9
```

B.

```
1*1=1   1*2=2   1*3=3
2*1=2   2*2=4
3*1=3
```

C.

```
1*1=1
1*2=2   2*2=4
1*3=3   2*3=6   3*3=9
```

D.

```
1*1=1
2*1=2   2*2=4
3*1=3   3*2=6   3*3=9
```

【答案】A

【解析】当 i=1 时，j 可以取 1、2、3，i*j 分别为 1、2、3；当 i=2 时，j 可以取 2、3，i*j 分别为 4、6；当 i=3 时，j 可以取 3，i*j 为 9。答案选择 A 选项。

57．以下叙述中正确的是（　　）。

A．break 语句只能用于 switch 语句体中

B．continue 语句的作用是使程序的执行流程跳出包含它的所有循环

C．在循环体内使用 break 语句和 continue 语句的作用相同

D．break 语句只能用在循环体内和 switch 语句体内

【答案】D

【解析】break 语句可用于 for 循环、while 循环、do-while 循环、while-do 循环、switch 语句，A 项错误、D 项正确；continue 语句的作用是跳过 continue 下面的执行语句，直接进入下一次循环的判断，B 项描述的是 break 的功能，B、C 项错误。答案选择 D 选项。

58．以下叙述中正确的是（　　）。

A．break 语句不能用于提前结束 for 语句的本层循环

B．使用 break 语句可以使流程跳出 switch 语句体

C．continue 语句使得整个循环终止

D．在 for 语句中，continue 与 break 的效果是一样的，可以互换

【答案】B

【解析】B 项正确，break 语句可以跳出 switch 语句和循环。A 项错误，break 语句在 for 循环内部的作用是跳出循环；C 项错误，continue 语句的作用是跳过本次循环体中余下尚未执行的语句，立刻进行下一次的循环条件判定，可以理解为仅结束本次循环，并没有使整个循环终止；D 项错误，continue 和 break 的效果完全不同。

59．有以下程序：

```
#include <stdio.h>
main()
{
   int i,data;
   scanf("%d",&data);
   for(i=0;i<5;i++)
   {
      if(i < data) continue;
      printf("%d,",i);
   }
   printf("\n");
}
```

程序运行时，从键盘输入：3<回车>后，程序输出结果为（　　）。

A. 3,4,

B. 1,2,3,4,

C. 0,1,2,3,4,5,

D. 0,1,2,

【答案】A

【解析】continue 语句只能用在循环结构中，其作用是结束本次循环，即不再执行循环体中 continue 语句之后的语句，而是立即转入对循环条件的判断与执行。本题执行过程为：输入 3，则 data=3；执行 for 循环，i=0，if 条件成立，结束本次循环，不输出 i 值，执行下一次循环；直到 i>=3，if 条件不成立，依次输出 i 值 3,4,；直到 i=5 退出 for 循环。答案选择 A 选项。

60．有如下程序：

```
#include<stdio.h>
main()
{
   int i,data;
   scanf("%d",&data);
   for(i=0;i<5;i++)
   {
      if(i>data)break;
      printf("%d,",i);
   }
   printf("\n");
}
```

程序运行时，从键盘输入：3<回车>后，程序输出结果为（　　）。

A. 3,4,

B. 0,1,

C. 3,4,5,

D. 0,1,2,3,

【答案】D

【解析】break 语句作用是结束整个循环过程，不再判断循环的条件是否成立，在嵌套循环中，break 语句只跳出最内层的一层循环。程序执行过程为：输入 3，则 data=3；执行 for 循环，在 i<=3 时，if 条件不成立，执行语句 printf("%d,",i);依次输出 i 值，0，1，2，3；直到 i=4 时，if 条件成立，执行 break 退出 for 循环。答案选择 D 选项。

61．有以下程序：

```
#include<stdio.h>
main()
{
   int i;
   for(i=1;i<=5;i++)
   {
      if(i%2)printf("*");
      else continue;
      printf("#");
   }
   printf("$\n");
}
```

程序运行后的输出结果是（　　）。

A. *#*#*#$

B. *#*#*$

C. *#*#$

D. *#*#*#*$

【答案】A

【解析】本题执行过程为：i=1，1%2=1，if 条件成立，输出*与#；i=2，2%2=0，if 条件不成立，执行 continue 结束本次循环，执行下一次循环；i=3，3%2=1，输出*与#；i=4，4%2=0，执行 continue，开始下一次循环；i=5，5%2=1，输出*与#；i=6 退出循环。最后输出$。答案选择 A 选项。

62．有以下程序：

```
#include <stdio.h>
main()
{
   int a=1,b=0;
   for(;a<5;a++)
   {
      if(a%2 == 0)break;
      continue;
      b += a;
   }
   printf("%d \n",b);
}
```

程序运行后的输出结果是（　　）。

A. 0

B. 1

C. 10

D. 4

【答案】A

【解析】该程序的 for 循环中，如果循环变量 a 为偶数，则执行 break 语句直接跳出循环；如果 a 为奇数，则直接执行 continue 语句进入下一次循环。因此无论 a 取何值，语句 b+=a 都不会执行，即 b 没有改变，最后输出 0。答案选择 A 选项。

二、填空题

给定程序中，函数 fun 的功能是：判定形参 a 所指的 N×N（规定 N 为奇数）的矩阵是否是“幻方”，若是，则函数返回值为 1；若不是，则函数返回值为 0。“幻方”的判定条件是：矩阵每行、每列、主对角线及反对角

线上元素之和都相等。

例如，以下 3×3 的矩阵就是一个“幻方”：

```
4  9  2
3  5  7
8  1  6
```

请在程序的下划线处填入正确的内容并把下划线删除，使程序得出正确的结果。

注意：部分源程序给出如下。

不得增行或删行，也不得更改程序的结构！

试题程序如下：

```
#include <stdio.h>
#define N 3
int fun(int (*a)[N])
{
   int i,j,m1,m2,row,colum;
   m1=m2=0;
   for(i=0; i<N;i++)
   {
      j=N-i-1;
      m1+=a[i][i];
      m2+=a[i][j];
   }
   if(m1!=m2) return 0;
   for(i=0; i<N;i++)
   {
      /*********found*********/
      row=colum=①_____;
      for(j=0; j<N;j++)
      {
         row+=a[i][j];
         colum+=a[j][i];
      }
      /*********found*********/
      if((row!=colum) ②_____ (row!=m1) ) return 0;
   }
   /*********found*********/
   return ③_____;
}
main()
{
   int x[N][N],i,j;
   printf("Enter number for array:\n");
   for(i=0; i<N;i++)
      for(j=0; j<N;j++)
         scanf("%d",&x[i][j]);
   printf("Array:\n");
   for(i=0;i<N;i++)
   {
      for(j=0;j<N;j++)
         printf("%3d",x[i][j]);
      printf("\n");
   }
   if(fun(x)) printf("The Array is a magic square.\n");
   else printf("The Array isn't a magic square.\n");
}
```

【答案】

①0

②||

③1

【解析】

填空 1：变量 row 存放每行的总和，变量 colum 存放每列的总和，应给 row、colum 赋初值为 0，因此应该填 0。

填空 2：if 判断每行的总和是否与列的总和相等、每行的总和与对角线的总和是否相等，程序中，两个条件若有一个满足，即返回 0，因此应该填写||。

填空 3：矩阵是“幻方”，函数返回 1。

三、改错题

请根据以下各小题的要求设计 C 应用程序（包括界面和代码）。

下列给定程序中，函数 fun()的功能是：将 n 个无序整数从小到大排序。请改正程序指定部位的错误，使它能得到正确结果。注意：不要改动函数 main()，不得增行或删行，也不得更改程序的结构。

试题程序如下：

```
#include <stdio.h>
void fun(int n, int *a)
{
   int i,j,p,t;
   for(j=0;j<n-1;j++)
   {
      p=j;
      /**********found**********/
      for(i=j+1;i<n-1;i++)
         if(a[p]>a[i])
            /**********found**********/
            t=i;
      if(p!=j)
      {
         t=a[j];
         a[j]=a[p];
         a[p]=t;
      }
   }
}
void putarr(int n, int *z)
{
   int i;
   for(i=1;i<=n;i++,z++)
   {
      printf("%4d",*z);
      if(!(i%10))
         printf("\n");
   }
   printf("\n");
}
void main()
{
   int aa[20]={9,3,0,4,1,2,5,6,8,10,7},n=11;
   printf("\n\nBefore sorting %d numbers:\n",n);
   putarr(n,aa);
   fun(n,aa);
   printf("\nAfter sorting %d numbers:\n",n);
   putarr(n,aa);
}
```

【答案】

（1）错误：for(i=j+1;i<n-1;i++)

正确：for(i=j+1;i<n;i++)

（2）错误：t=i;

正确：p=i;

【解析】

错误 1：选择排序法是在外循环中从第一个元素开始，依次与比它小的元素进行交换，直到交换完第 n-1 个元素，最后一个元素就是最大的元素，循环次数为 n-1 次，如果设定循环次数为 n 次，则最后一次第 n 个元素与自己交换；内循环是找出比当前元素更小的元素，故是从当前需要交换的元素的下一个元素开始，直到第 n 个元

素。故“for(i=j+1;i<n-1;i++)”应改为“for(i=j+1;i<n;i++)”。

错误 2：t 是 fun 函数中用来交换两个元素的辅助变量，不是交换元素下标的变量，因此 t=i;是错误的。p 是用来记录当前最小元素下标的，当 a[p]>a[i]时，把 i 赋给 p。所以“t=i;”应改为“p=i;”。

四、设计题

程序定义了 N×N 的二维数组，并在主函数中自动赋值。请编写函数 fun（int a[][N]，int n），该函数的功能是：使数组左下半三角元素中的值乘以 n。例如，若 n 的值为 3，a 数组中的值为：

$$a=\begin{pmatrix}1 & 9 & 7\\ 2 & 3 & 8\\ 4 & 5 & 6\end{pmatrix}$$

则返回主程序后 a 数组中的值应为：

$$\begin{pmatrix}3 & 9 & 7\\ 6 & 9 & 8\\ 12 & 15 & 18\end{pmatrix}$$

注意：部分源程序给出如下。

请勿改动函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

试题程序如下：

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#define N 5
void fun(int a[][N],int n)
{

}
main()
{
   int a[N][N],n,i,j;
   printf("*****The array *****\n");
   for(i=0;i<N;i++)
   {
      for(j=0;j<N;j++)
      {
         a[i][j]=rand()%10;
         printf("%4d",a[i][j]);
      }
      printf("\n");
   }
   n=rand()%4;
   printf("n=%4d\n",n);
   fun(a,n);
   printf("*****The result *****\n");
   for(i=0;i<N;i++)
   {
      for(j=0;j<N;j++)
         printf("%4d",a[i][j]);
      printf("\n");
   }
}
```

答：

```
void fun(int a[][N],int n)
{
   int i,j;
   if(a==NULL)return;
   for(i=0;i<N;i++)
      for(j=0;j<=i;j++)
         a[i][j]=a[i][j]*n;
}
```

【解析】首先从数组中找出要被乘以 n 的元素，即找出将被挑出的元素在原数组中的分布规律。通过观察得出，要被处理的元素下标值的范围是每行中从第一个元素开始，直到列数等于该行行数时为止。找到这个规律后，依次从数组中取得符合要求的元素，然后乘以 n。

# 第 6 章　利用数组处理批量数据

## 一、选择题

1．以下叙述中正确的是（　　）。

A．数组说明符的一对方括号中只能使用整型常量，而不能使用表达式

B．一条语句只能定义一个数组

C．每个数组包含一组具有同一类型的变量，这些变量在内存中占有连续的存储单元

D．在引用数组元素时，下标表达式可以使用浮点数

【答案】C

【解析】A 项错误，方括号中应当是整型常量表达式，可以使用表达式，只要计算结果是整型常量即可；B 项错误，可以使用逗号隔开，来定义多个数组；D 项错误，引用数组时，下标必须是自然数，还应保证下标不越界。C 项正确，数组内部的变量都是同一类型，在内存中是连续存储的。答案选择 C 选项。

2．以下叙述中正确的是（　　）。

A．语句“int a[8]={0};”是合法的

B．语句“int a[]={0};”是不合法的，遗漏了数组的大小

C．语句“char a[2]={"A","B"};”是合法的，定义了一个包含两个字符的数组

D．语句“char a[3];a="AB";”是合法的，因为数组有三个字符空间的容量，可以保存两个字符

【答案】A

【解析】A 项正确，表示定义了长度为 8 的 int 型数组，它里面的每个元素都是 0。当所赋初值少于所定义数组的元素个数时，将自动给后面的元素补以初值 0。B 项错误，C 语言规定可以通过赋初值来定义数组的大小，这时数组说明符的一对方括号中可以不指定数组的大小；C 项错误，a 是 char 类型的数组，里面的元素应该是字符而非字符串，应该用单引号括起来；D 项错误，数组变量一旦定义，其地址值不可改变，不能给数组名重新赋值。答案选择 A 选项。

3．以下选项中叙述正确的是（　　）。

A．char c1,c2,*c3,c4[40];是合法的变量定义语句

B．数组说明符的一对方括号中只能使用整型常量，而不能使用表达式

C．数组下标的下限可以是负值

D．若有数组定义 float array[4];，则语句 printf("%f",array[3.12]);是合法的

【答案】A

【解析】A 项正确定义了字符变量 c1、c2，字符型指针 c3，字符型一维数组 c4[40]，A 项正确。方括号中应当是整型常量表达式，可以使用表达式，只要计算结果是整型常量即可，B 项错误。数组下标的下限不可以是负值，C 项错误。数组下标必须是整型数据，3.12 为浮点数，D 项错误。答案选择 A 选项。

4．以下叙述中错误的是（　　）。

A．同一个数组中所有元素的类型相同

B．不可以跳过前面的数组元素，给后面的元素赋初值 0

C．定义语句：int a[10]={0};，给 a 数组中所有元素赋初值 0

D．若有定义语句：int a[4]={1,2,3,4,5};，编译时将忽略多余的初值

【答案】D

【解析】数组初始化时，若赋初值的个数多于所定义数组的元素个数时，编译器会报错。答案选择 D 选项。

5．若要定义一个具有 5 个元素的整型数组，以下定义语句错误的是（　　）。

A．int a[5]={0};

B．int b[]={0,0,0,0,0};

C．int c[2+3];

D．int i=5,d[i];

【答案】D

【解析】在 C 语言中，一维数组的下标可以是整型常量或整型表达式，但不可以是变量。D 项，i 为变量。

答案选择 D 选项。

6．设有如下程序段：

```
int a[8]={0};
int b[]={0};
char c[2]={"A","B"};
char d="AB";
```

以下叙述正确的是（　　）。

A．只有 a，b，c 的定义是合法的

B．a，b，c，d 的定义都是合法的

C．只有 a，b 的定义是合法的

D．只有 c 的定义是合法的

【答案】C

【解析】在对数组进行初始化时，如果对数组的全部元素赋以初值，定义时可以不指定数组长度，如果被定义数组的长度与初值个数不同，则数组长度不能省略；如果在说明数组时给出了长度，但没有给所有的元素赋予初始值，而只依次给前面的几个数组元素赋予初值，那么 C 语言将自动对余下的元素赋初值 0。定义整型数组 a，长度为 8，并且其每个元素赋初值 0，a 定义是合法的。定义整型数组 b，没有指定长度，但为 1 个元素赋值 0，所以自动指定数组长度为 1，b 定义是合法的。定义字符型数组 c，长度为 2，应对其元素赋值字符型常量，而“A”与“B”均为字符串，c 定义不合法。定义字符 d，赋初值字符串“AB”，这是不合法的定义，若要定义数组，必须要有“[ ]”。可知，ab 定义合法，cd 定义不合法，C 项正确。答案选择 C 选项。

7．下列选项中，能正确定义数组的语句是（　　）。

A．int N=2008;int num[N];

B．int num[];int num[N];

C．

```
#define N 2008
int num[N];
```

D．int num[0..2008];

【答案】C

【解析】一维数组定义格式是：类型名 数组名[整型常量表达式]。A 项错误，数组长度不能由变量指明；B 项错误，没有指明数组的长度；C 项正确，它使用预处理宏定义，N 可以看做是常量；D 项错误，数组长度应该整型常量表达式。答案选择 C 选项。

8．下列定义数组的语句中，正确的是（　　）。

A．int x[];

B．int N=10;int x[N];

C．int x[0..10];

D．

```
#define N 10
int x[N];
```

【答案】D

【解析】一维数组定义格式是：类型名 数组名[整型常量表达式]。D 项正确，它使用预处理宏定义，N 可以看做是常量。A 项错误，没有指明数组的长度；B 项错误，数组长度不能由变量指明；C 项错误，数组长度应该整型常量表达式。答案选择 D 选项。

9．要求定义一个具有 6 个元素的 int 型一维数组，以下选项中错误的是（　　）。

A．int N=6,a[N];

B．int a[2*3]={0};

C．

```
#define N 3
```

int a[N+N];

D．int a[]={1,2,3,4,5,6};

【答案】A

【解析】数组名后面括号内，必须是整型常量，不可以是变量，答案选择 A 选项。

10．若有定义语句：int m[]={5,4,3,2,1},i=4;，则下面对 m 数组元素的引用中错误的是（　　）。

A．m[--i]

B．m[2*2]

C．m[m[0]]

D．m[m[i]]

【答案】C

【解析】在 C 语言中，数组的下标是从 0 开始的，所以它的上限是数组元素个数减 1。如果超过这个范围来引用数据元素就会溢出，造成运行时错误。本题的数组 m 有 5 个元素，所以它的下标范围是 0～4。A 项，--i 的值为 3；B 项，2*2 的值为 4；C 项，m[0]等于 5，超出范围，出现溢出错误；D 项，m[i]的值为 4，即 m[4]的值为 1。答案选择 C 选项。

11．若有说明语句：

int*ptr[10];

以下叙述正确的是（　　）。

A．ptr 是一个具有 10 个指针元素的一维数组，每个元素都只能指向整型变量

B．ptr 是指向整型变量的指针

C．ptr 是一个指向具有 10 个整型元素的一维数组的指针

D．ptr 是一个指向 10 个整型变量的函数指针

【答案】A

【解析】A 项正确，int *ptr[10]定义一个指针数组。ptr 是指针数组不是指针，B 项错误；定义指向具有 10 个整型元素的一维数组的指针格式为：int(*ptr)[10]，C 项错误；函数指针是指向函数的指针，不会指向整型数组，D 项错误。答案选择 A 选项。

12．有以下程序：

```
#include <stdio.h>
main()
{
   int s[12] = {1,2,3,4,4,3,2,1,1,1,2,3}, c[5]={0},i;
   for(i=0;i<12;i++)
      c[s[i]]++;
   for(i=1;i<5;i++)
      printf("%d",c[i]);
   printf("\n");
}
```

程序的运行结果是（　　）。

A．4332

B．2344

C．1234

D．1123

【答案】A

【解析】s[i]元素作为 c 数组的下标，使 c 数组元素自增，执行完毕后，c[1]自增 4 次，c[2]自增 3 次，c[3]自增 3 次，c[4]自增 2 次，c 数组元素为 04332，按题目要求输出，4332。答案选择 A 选项。

13．设有 n 个数按从大到小的顺序存放在数组 x 中，以下能使这 n 个数在 x 数组中的顺序变为从小到大的是

（　　）。

A.

```
for(i=0;i<n/2;i++)
{
   t=x[i];
   x[i]=x[n-i-1];
   x[n-i-1]=t;
}
```

B.

```
for(i=0;i<n;i++)
{
   t=x[i];
   x[i]=x[n-i-1];
   x[n-i-1]=t;
}
```

C.

```
for(i=0;i<n/2;i++)
{
   t=x[i];
   x[i]=x[n-i+1];
   x[n-i+1]=t;
}
```

D.

```
for(i=0;i<n/2;i+=2)
{
   t=x[i];
   x[i]=x[n-i-1];
   x[n-i-1]=t;
}
```

【答案】A

【解析】本题使用 for 循环实现数组元素首尾倒置。A 项中控制变量上限为 n/2，用变量 t 实现数组下标值 i 与 n-i-1 的元素交换，使数组元素首尾倒置，从而实现从小到大排列，A 项正确。B 项控制变量上限为 n，用变量 t 实现数组下标值 i 与 n-i-1 的元素交换，当 i=n/2 时，已经实现数组首尾倒置，而继续运行至 i=n，将再次将新的数组首尾倒置，与原数组相比没有发生变化，依然是从大到小排列，B 项错误。C 项数组越界，i=0 时 x[n+1]不存在，C 项错误。D 项控制变量每次加 2，实现数组偶数下标值的元素首尾倒置，奇数下标值元素不变，不能实现整个数组从小到大排列，D 项错误。答案选择 A 选项。

14．设有一个 M*N 的矩阵已经存放在一个 M 行 N 列的数组 x 中，且有以下程序段：

```
sum=0;
for(i=0;i<M;i++) sum+=x[i][0]+x[i][N-1];
for(j=1;j<N-1;j++) sum+=x[0][j]+x[M-1][j];
```

以上程序段计算的是（　　）。

A．矩阵两条对角线元素之和

B．矩阵所有不靠边元素之和

C．矩阵所有元素之和

D．矩阵所有靠边元素之和

【答案】D

【解析】程序执行过程为：第一个 for 循环实现对第一列和第 N 列求和。第二个 for 循环在上一个 for 循环结果上实现对第一行和第 M 行从第二个元素到第 N-1 个元素的求和，总体来说，就是矩阵所有靠边元素之和，答案选择 D 选项。

15．有以下程序段：

```
int *p1,*p2,a[10];
p1=a;
p2=&a[5];
```

则 p2-p1 的值为（　　）。

A．5

B．10

C．12

D．无法确定

【答案】A

【解析】当直接用数组名初始化指针时，表示把数组首地址的值赋给指针，p1=a，表示把 a[0]地址赋给 p1，p2=&a[5]，表示将数组第 5 个元素地址赋给 p2，p2-p1=5，故答案选择 A 选项。

16．设有如下定义语句：

```
int m[ ]={2,4,6,8},*k=m;
```

以下选项中，表达式的值为 6 的是（　　）。

A．*(k+2)

B．k+2

C．*k+2

D．*k+=2

【答案】A

【解析】本题中 k 指向数组 m 的首地址。*(k+2)先将指针后移两个元素，即为 m[2]的地址，然后取出 m[2]的值 6。B 项，k+2 是 m[2]的地址；C 项，*k 表示 m[0]的值，然后再加上 2，所以为 4；D 项，表示给 m[0]的值加 2，也为 4。答案选择 A 选项。

17．若有定义语句：

```
double x[5]={1.0,2.0,3.0,4.0,5.0}, *p=x;
```

则错误引用 x 数组元素的是（　　）。

A．*p

B．x[5]

C．*(p+1)

D．*x

【答案】B

【解析】直接引用一维数组元素的表达式为：数组名[下标]。数组大小为 n 时，下标的取值范围为 0～(n-1)，所以本题中 x 的下标为 0～4，x[5]溢出，B 项错误。还可以通过指针引用一维数组元素。指针 p 指向该数组，所以*p 表示 x[0]，A 项正确；指针 p+1 指向数组 x 的第二个元素的地址，*(p+1)表示 x[1]，C 项正确；数组名 x 为 x 数组元素的首地址，*x 表示对 x[0]的引用，D 项正确。答案选择 B 选项。

18．设有定义：

```
int a[10]={0,1,2,3,4,5,6,7,8,9}, *p=a,i;
```

若 0≤i≤9，则对 a 数组元素的引用错误的是（　　）。

A．a[10]

B．*(&a[i])

C．p[i]

D．a[p-a]

【答案】A

【解析】长度为 n 的数组其各个元素的下标应该是从 0 到 n-1，因此，长度为 10 的数组 a，第 10 个元素为 a[9]，而不是 a[10]，答案选择 A 选项。

19．设有定义

double a[10],*s=a;

以下能够代表数组元素 a[3]的是（　　）。

A．(*s)[3]

B．*(s+3)

C．*s[3]

D．*s+3

【答案】B

【解析】B 项，s+3 指向 a[3]的地址，*(s+3)取出数组元素 a[3]。A 项，小括号和方括号的优先级相同，从左向右结合，先取*s 的值，然后再进行方括号运算，编译错误；C 项，方括号的优先级高于*，s[3]等价于*(s+3)，代表 a[3]元素，对 a[3]元素进行间址运算，编译错误；D 项，间址操作符*的优先级高于加号，先取出 s 指向的数据，然后加 3，并不是数组元素 a[3]。答案选择 B 选项。

20．若有以下定义：

int x[10],*pt=x;

则对 x 数组元素的正确引用是（　　）。

A．*&x[10]

B．*(x+3)

C．*(pt+10)

D．pt+3

【答案】B

【解析】数组的下标是从 0 开始的，故 x[10]实际上具体为 x[0]，x[1]，…，x[9]，而若使用 x[10]就会溢出、编译出错，所以 AC 两项错误；D 项，pt+3 指向地址，并未引用到元素；B 项的表达正确，代表 x[3]中的元素，答案选择 B 选项。

21．执行以下程序段后，s 的值为（　　）。

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

for(k=0;k<8;k+=2) s+=*(a+k);

A．13

B．16

C．17

D．45

【答案】B

【解析】获取数组 A 中第 i 个元素时，有两种形式：一是 A[i]，二是*(A+i)。数组 a 中的元素为：a[0]=1，a[1]=2，…，a[8]=9。k=0 时，s = s+*(a+0) = 0+1 = 1；k=2 时，s = 1+*(a+2) = 1+3 = 4；k=4 时，s = 4+5 = 9；k=6 时，s = 9+7 = 16。答案选择 B 选项。

22．有以下程序

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

程序执行后的输出结果是（　　）。

A．25

B．20

C．45

D．36

【答案】A

【解析】*(t+i)等价于 t[i]，依次访问数组 t 中的偶数位元素，s = t[0]+t[2]+t[4]+t[6]+t[8] = 1+3+5+7+9 = 25。答案选择 A 选项。

23．有以下程序：

```
#include<stdio.h>
main()
{
   int a[]={10,20,30,40},*p=a,i;
   for(i=0;i<=3;i++)
   {
      a[i]=*p;
      p++;
    }
   printf("%d\n",a[2]);
}
```

程序运行后的输出结果是（　　）。

A．30

B．40

C．10

D．20

【答案】A

【解析】for 循环重新给数组 a 赋值，执行完成后数组 a 中的值没有发生变化，故最终输出的 a[2]还是原来的 a[2]，为 30。答案选择 A 选项。

24．有以下程序：

```
#include<stdio.h>
main()
{
   int a[]={1,2,3,4},y,*p=&a[3];
   --p;
   y=*p;
   printf("y=%d\n",y);
}
```

程序的运行结果是（　　）。

A．y=0

B．y=1

C．y=2

D．y=3

【答案】D

【解析】数组 a 定义结果为，a[0]=1，a[1]=2，a[2]=3，a[3]=4；指针 p 指向 a[3]。执行--p；语句后，p 指向数组元素 a[2]；y=*p，y 的值为指针 p 所指向的地址的存储值，为 3。答案选择 D 选项。

25．fun 函数的功能是：通过键盘输入给 x 所指的整型数组所有元素赋值。在下划线处应填写的是（　　）。

```
#include <stdio.h>
#define N 5
void fun(int x[N])
{
   int m;
   for(m=N-1;m>=0;m--)scanf("%d",_____);
}
```

A．&x[++m]
B．&x[m+1]
C．x+(m++)
D．x+m
【答案】D
【解析】数组 x 中包含 5 个元素，分别为 x[0]，x[1]，x[2]，x[3]，x[4]。AB 两项，第一次循环时，m=4，x[++m]和 x[m+1]均越界；C 项，m++与 for 循环中的表达式 m--构成死循环。答案选择 D 选项。

26．以下函数的功能是：通过键盘输入数据，为数组中的所有元素赋值。

```
#include <stdio.h>
#define N 10
void fun(int x[N])
{
   int i=0;
   while(i<N)scanf("%d",_____);
}
```

在程序中下画线处应填入的是（　　）。
A．x+i
B．&x[i+1]
C．x+(i++)
D．&x[++i]
【答案】C
【解析】划线处需要完成两个功能：①给 x[i]元素赋值；②i 变量加一。A 项错误，仅能对 x[0]赋值，且会是程序陷入死循环；B 项错误，仅能对 x[1]赋值，程序陷入死循环；D 项错误，不能实现为 x[0]赋值，而且还会导致数组越界。答案选择 C 选项。

27．以下程序中给数组所有元素输入数据，请从选项中选择正确的答案填入下划线处。

```
#include <stdio.h>
main()
{
   int a[10],i=0;
   while(i<10)scanf("%d",_____);
   …
}
```

A．a+(i++)
B．&a[i+1]
C．a+i
D．&a(i++)
【答案】A

【解析】A 项正确，a 为数组首地址，i++先取 i 值再加 1，scanf 读入的数据依次存放在数组 a 中；B 项错误，程序会进入死循环；C 项错误，控制变量 i 没有依次加 1，无法结束循环，也无法对整个数组赋值；D 项错误，数组元素引用为[]，不是()。答案选择 A 选项。

28．有以下程序：

```
#include <stdio.h>
main()
{
   int a[10]={11,12,13,14,15,16,17,18,19,20},*p=a,i=9;
   printf("%d,%d,%d\n",a[p-a],p[i],*(&a[i]));
}
```

程序运行后的输出结果是（　　）。

A．11,19,19

B．12,20,20

C．11,20,20

D．12,19,20

【答案】C

【解析】a 表示数组 a 的第一个元素的地址，*p=a 则将 p 指向了数组 a 的第一个元素的地址，p-a=0，所以 a[p-a]即为 a[0]，p[i]相当于 a[i]即 a[9]，输出第一个结果为 11、第二个结果为 20。取地址运算符&和取值运算符*互为逆运算，所以*(&a[i]))的值仍为 a[i]即 a[9]，输出第三个结果为 20，答案选择 C 选项。

29．有以下程序：

```
#include <stdio.h>
void fun(int *p)
{
   printf("%d\n",p[5]);
}
main()
{
   int a[10]={1,2,3,4,5,6,7,8,9,10};
   fun(&a[3]);
}
```

程序运行后的输出结果是（　　）。

A．5

B．6

C．8

D．9

【答案】D

【解析】fun 函数的功能是在屏幕上打印输入参数 p 后的第五个元素。在 main 中，a[3]的值为 4，&a[3]即为取 a[3]所指内容的地址，即指针 p 指向 a[3]的地址，故 p[5]应为 a[3+5]=a[8]=9。答案选择 D 选项。

30．有以下程序

```
#include <stdio.h>
main()
{
   int a[]={2,4,6,8,10},x,*p,y=1;
   p=&a[1];
   for(x=0;x<3;x++)y+=*(p+x);
   printf("%d\n",y);
}
```

程序的输出结果是（　　）。

A．19

B．13

C．11

D．15

【答案】A

【解析】p=&a[1]是把 p 指向 a[1]，即*p=a[1]=4；在 for 循环中，通过指针 p 依次访问 a[1]、[2]、a[3]，所以 y=1+4+6+8=19。答案选择 A 选项。

31．有如下程序：

```
#include <stdio.h>
main()
{
   int i,*ptr;
   int array[3]={8,2,4};
   for(ptr=array,i=0;i<2;i++)
      printf("%d,",*ptr++);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．8,2,

B．8,8,

C．2,4,

D．4,8,

【答案】A

【解析】*的优先级低于++，即先运算 ptr++再运算*ptr。程序执行过程为：定义指针 ptr 与数组 array，执行 for 循环，使指针指向数组首地址，i=0，输出指针指向的元素 array[0]=8，然后指针加 1，指向数组下一个元素；i=1，输出 array[1]=2，指针指向数组下一个元素；i=2 退出 for 循环。程序依次输出 8 和 2。答案选择 A 选项。

32．有以下程序：

```
#include <stdio.h>
main()
{
   int i,*ptr;
   int array[4]={1,1,3,4};
   for(ptr=array,i=0;i<3;i++)
      printf("%d,",*ptr++);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．1,2,4,

B．1,3,4,

C．1,1,3,

D．1,1,4,

【答案】C

【解析】*的优先级低于++，即先运算 ptr++再运算*ptr。定义指针 ptr 与数组 array，执行 for 循环，使指针指向数组首地址，i=0，输出指针指向的元素 array[0]=1，然后指针加 1，指向数组下一个元素；i=1，输出 array[1]=1，指针指向数组下一个元素；i=2，输出指针指向的元素 array[2]=3，然后指针加 1，指向数组下一个元素；i=3 退出 for 循环。程序依次输出 1、1、3。答案选择 C 选项。

33．有如下程序：

```
#include <stdio.h>
main()
{
   int i, *ptr;
   int array[5] = {5,3,1};
   for (ptr=array, i=0; i<5; i++, ptr++)
   {
      if (*ptr == 0)
         putchar('X');
      else
         putchar('A' + *ptr);
   }
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．FDBXX

B．FFFXX

C．FDBBB

D．ABCDE

【答案】A

【解析】对数组进行初始化时，如果在说明数组时给出了长度，但没有给所有的元素赋予初始值，而只依次给前面的几个数组元素赋予初值，那么 C 语言将自动对余下的元素赋初值，即 array={5,3,1,0,0}。程序执行过程为：执行 for 循环，将数组首地址赋给指针 ptr，依次遍历每一个元素，如果数组元素为 0 则输出'X'，如果不为 0 则按照字母表输出字符'A'后第 array[i]个字符。程序运行后的输出结果为：FDBXX，答案选择 A 选项。

34．有以下程序：

```
#include <stdio.h>
main()
{
   int x[]={8,2,6,12,5,15},f1,f2;
   int *p=x;
   f1=f2=x[0];
   for(;p<=x+5;p++)
   {
      if(f1<*p)f1=*p;
      if(f2>*p)f2=*p;
   }
   printf("%d,%d\n",f1,f2);
}
```

程序的运行结果是（　　）。

A．15,2

B．15,15

C．2,15

D．8,8

【答案】A

【解析】本题求数组的最大值和最小值，首先把数组 x 首地址的值赋给指针 p，故*p 初始值为 8，而 p 是地址值，p+1 相当于数组中下一元素的地址，在 for 循环，是求数组 x 对应的最大值和最小值，f1 为最大值，f2 为最小值。因此输出为 15 和 2。答案选择 A 选项。

35．若有函数声明：

void fun(float array[],int*ptr);

以下叙述正确的是（　　）。

A．函数参数 array，ptr 都是指针变量

B．函数参数 array 不是指针变量，ptr 是指针变量

C．调用函数时，实参数组的值将一一复制给 array 数组

D．调用函数时，array 是按值传送，ptr 是按地址传送

【答案】A

【解析】数组名为数组的首地址，也是指向数组的指针，所以 array 和 ptr 都是指针，A 项正确，B 项错误。调用函数时，将实参数组首地址赋给指针 array，而不是传递整个数组元素，C 项错误。调用函数时，array 与 ptr 都是按地址传送的，D 项错误。答案选择 A 选项。

36．若有函数

void fun(double a[],int *n)

{……}

以下叙述中正确的是（　　）。

A．调用 fun 函数时只有数组执行按值传送，其他实参和形参之间执行按地址传送

B．形参 a 和 n 都是指针变量

C．形参 a 是一个数组名，n 是指针变量

D．调用 fun 函数时将把 double 型实参数组元素一一对应地传送给形参 a 数组

【答案】C

【解析】A 项，用数组名（指向数组首元素的一个指针常量）作为实参可以实现实参和形参之间执行按地址传送；B 项，形参 a 是一个数组名，是指针常量而不是变量；D 项，调用 fun 函数时只是将实参数组的首地址传送给形参 a。答案选择 C 选项。

37．若主函数中有定义语句：

int a[10],b[10],c;

在主函数前定义的 fun 函数首部为：

void fun(int x[])

则以下选项中错误的调用语句是（　　）。

A. fun(b);

B. fun(&c);

C. fun(&a[3]);

D. fun(b[11]);

【答案】D

【解析】fun 函数的形式参数为一个数组，需要实参为一个地址，而 b[11]是一个整型元素，参数类型不一致，且 b[11]已经溢出，所以 D 项错误。答案选择 D 选项。

38. 有以下程序

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

程序运行后的输出结果是（　　）。

A. 1,2,6,8,10,12,7,8,9,10,

B. 1,2,3,4,5,6,7,8,9,10,

C. 1,2,3,4,10,12,14,16,9,10,

D. 1,2,6,8,10,12,14,16,9,10,

【答案】A

【解析】数组名用作函数参数的情况类似于指针，属于传地址，故对形参数组元素的修改会同时修改实参。f 函数通过一个 for 循环语句，将传入数组的元素从下标 2 到下标 5 每个各自乘以 2。因此，调用结束时，数组 a 中的内容为{1,2,6,8,10,12,7,8,9,10}。答案选择 A 选项。

39. 有以下程序：

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

程序运行的结果是（　　）。

A．345678

B．876543

C．1098765

D．321678

【答案】D

【解析】在 main()函数中定义了一个有 10 个元素的数组 k，并且赋初值为{1,2,3,4,5,6,7,8,9,10}。执行函数 fun(k,5)，把 k 的首地址赋给形参 a，把 5 赋给形参 n。在 fun()函数中，for 循环执行了 2 次：第一次，i 的值为 0，循环体中将 a[0]与 a[4]的值互换；第二次，i 值为 1，这次是将 a[1]与 a[3]的值互换。所以，fun()函数执行完后，数组 k 中的内容为{5,4,3,2,1,6,7,8,9,10}。接下来，循环输出 a[2]～a[7]的值，故输出结果是 321678。答案选择 D 选项。

40．有以下程序：

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

程序的运行结果是（　　）。

A．2,4,6,8,10,9,7,5,3,1,

B．10,8,6,4,2,1,3,5,7,9,

C．1,2,3,4,5,6,7,8,9,10,

D．1,3,5,7,9,10,8,6,4,2,

【答案】A

【解析】程序的执行过程为：调用函数 fun，将数组 c 地址与 n 数值传入函数，此函数实现，将数组偶数位置元素从中间向前依次放入数组前半段，数组奇数位置元素从中间向后依次放入数组后半段，调用结果为 c[10]={2,4,6,8,10,9,7,5,3,1}，之后依次输出。答案选择 A 选项。

41．有以下程序：

```
#include <stdio.h>
void fun(int a[], int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        if(i%2==0)
            a[i] += n;
        else
            a[i] -= n;
    }
}
main()
{
    int c[5]={5,4,3,2,1},i;
    fun(c,5);
    for(i=0;i<5;i++)
        printf("%d,",c[i]);
    printf("\n");
}
```

程序运行后的输出结果是（　　）。

A. 10,-1,8,-3,6,

B. 5,4,3,2,1,

C. 10,2,8,4,6,

D. 5,-1,3,-3,1,

【答案】A

【解析】fun 函数功能是：对数组中，下标为偶数的元素累加 5，下标为奇数的元素减去 5，所以执行完 fun 函数后，数组 c 变为{10,-1,8,-3,6}。答案选择 A 选项。

42. 有以下程序：

```
#include <stdio.h>
void fun(int a[],int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        if(i % 3==0)
            a[i]-=n;
        else
            a[i]+=n;
    }
}
main()
{
    int c[5]={6,7,8,9,10},i;
    fun(c,5);
    for(i=0;i<5;i++)printf("%d,",c[i]);
    printf("\n");
}
```

程序运行后的输出结果是（　　）。

A. 1,12,13,4,15,

B. 10,9,8,7,6,

C. 1,7,13,9,15,

D. 10,12,8,4,6,

【答案】A

【解析】fun 函数功能是：将数组 c 中下标对 3 求余为 0 的元素减去 5，下标对 3 求余不为 0 的元素累加 5，所以执行完 fun 函数后，数组 c 变为{1,12,13,4,15}。答案选择 A 选项。

43. 若有以下程序

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

则程序的输出结果是（　　）。

A．2,9,4,7,6,5,8,3,10,1,

B．10,9,8,7,6,5,4,3,2,1,

C．10,1,8,3,6,5,4,7,2,9,

D．1,10,3,8,5,6,7,4,9,2,

【答案】C

【解析】fun 函数的作用是把数组 a 中的奇数位置元素从小到大排序。答案选择 C 选项。

44．以下程序中函数 f 的功能是：当 flag 为 1 时，进行由小到大排序；当 flag 为 0 时，进行由大到小排序。

```
#include <stdio.h>
void f(int b[],int n,int flag)
{
   int i,j,t;
   for(i=0;i<n-1;i++)
      for(j=i+1;j<n;j++)
         if(flag?b[i]>b[j]:b[i]<b[j])
         {
            t=b[i];
            b[i]=b[j];
            b[j]=t;
         }
}
main()
{
   int a[10]={5,4,3,2,1,6,7,8,9,10},i;
   f(&a[2],5,0);
   f(a,5,1);
   for(i=0;i<10;i++)
      printf("%d,",a[i]);
}
```

程序运行后的输出结果是（　　）。

A. 1,2,3,4,5,6,7,8,9,10,

B. 3,4,5,6,7,2,1,8,9,10,

C. 5,4,3,2,1,6,7,8,9,10,

D. 10,9,8,7,6,5,4,3,2,1,

【答案】B

【解析】第一次排序：将 a[2]～a[6]的 5 个元素按从小到大排序，排序后数组为 5,4,7,6,3,2,1,8,9,10；第二次排序：将 a[0]～a[4]的 5 个元素按从大到小排序，排序后数组为 3,4,5,6,7,2,1,8,9,10。答案选择 B 选项。

45．有以下程序

```
#include<stdio.h>
void f(int *p);
main()
{
   int a[5]={1,2,3,4,5},*r=a;
   f(r);
   printf("%d\n",*r);
}
void f(int *p)
{
   p=p+3;
   printf("%d,",*p);
}
```

程序运行后的输出结果是（　　）。

A. 1,4

B. 4,4

C. 3,1

D．4,1

【答案】D

【解析】在调用 f 函数时，形参 p 通过传递参数指向了数组 a 的起始地址，执行语句“p=p+3;”后指向了 a[3]的地址，所以输出 4；主函数中，指针 r 仍指向数组 a 的首地址，所以输出 1。答案选择 D 选项。

46．有以下程序

```
#include <stdio.h>
void f(int *q)
{
   int i=0;
   for(;i<5;i++)(*q)++;
}
main()
{
   int a[5]={1,2,3,4,5},i;
   f(a);
   for(i=0;i<5;i++)
      printf("%d,",a[i]);
}
```

程序运行后的输出结果是（　　）。

A．2,2,3,4,5,

B．6,2,3,4,5,

C．1,2,3,4,5,

D．2,3,4,5,6,

【答案】B

【解析】函数 f 中(*q)++的表示的是 q 指向的存储空间的值自增，但是 q 的指向不变，f 的作用就是 q 指向的值加 5。数组作为实参进行函数调用时，以指针的方式传递，形参 q 指向数组的第一个元素 a[0]，因为 q 的指向没有发生变化，所以除 a[0]外，a 中其他元素不发生变化。答案选择 B 选项。

47．有以下程序：

```
#include <stdio.h>
#define N 8
void fun(int *x,int i)
{
   *x=*(x+i);
}
main()
{
   int a[N]={1,2,3,4,5,6,7,8},i;
   fun(a,2);
   for(i=0;i<N/2;i++)
   {
      printf("%d,",a[i]);
   }
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．1,3,1,3,

B. 2,2,3,4,
C. 3,2,3,4,
D. 1,2,3,4,
【答案】C
【解析】函数 fun 的功能是把整型指针 x 后面第 i 个内存单元中的内容复制到 x 所指向的内存单元中。在主函数中先调用 fun 函数，即令 a[0]=a[2]，然后输出数组 a 的前 4 个值。答案选择 C 选项。

48. 有如下程序：

```
#include <stdio.h>
void change(int *array,int len)
{
   for(;len>=0;len--)
      array[len]+=2;
}
main()
{
   int i,array[5]={1,2};
   change(array,4);
   for(i=0;i<4;i++)
      printf("%d,",array[i]);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。
A. 2,3,4,5,
B. 3,4,5,6,
C. 3,4,2,2,
D. 1,2,0,0,
【答案】C
【解析】本题程序执行过程为：调用 change 函数，将数组 array={1,2,0,0,0}首地址传入函数，函数实现将数组每个元素加 2，arrray={3,4,2,2,2}。依次输出数组前 4 个元素为 3、4、2、2。答案选择 C 选项。

49. 有以下程序：

```
#include <stdio.h>
void change(int * array,int len)
{
   for(;len>=0;len--)array[len]-=1;
}
main()
{
   int i, array[5]={2,2};
   change(array,4);
   for(i=0;i<5;i++)printf("%d,",array[i]);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。
A. 1,1,-1,-1,-1,
B. 1,0,-1,1,-1,
C. 1,1,1,1,1,

D．1,-1,1,-1,1,

【答案】A

【解析】在 main()函数中，首先给一维数组 array 赋初值[2,2,0,0,0]，再调用 change 函数，对 array 数组中的每一个数进行减 1 处理，最后使用 for 循环语句输出数组元素的值，答案选择 A 选项。

50．有以下程序

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

程序的运行结果是（　　）。

A．1,2,3,4,9,8,7,6,5,0,

B．0,9,8,7,6,5,1,2,3,4,

C．0,9,8,7,6,5,4,3,2,1,

D．1,2,3,4,5,6,7,8,9,0,

【答案】A

【解析】在本题中，主函数在调用 fun 函数进行排序时，传递的参数是 c+4 和 6，fun 函数实现的功能是将数组 c 的第 5 个元素开始的 6 个元素依次进行从大到小的顺序排列。排序之后，数组 c 的内容变为{1,2,3,4,9,8,7,6,5,0}。答案选择 A 选项。

51．有以下程序（函数 fun 只对下标为偶数的元素进行操作）

```
#include<stdio.h>
void fun(int *a,int n)
{
   int i,j,k,t;
   for(i=0;i<n-1;i+=2)
   {
      k=i;
      for(j=i;j<n;j+=2) if(a[j]>a[k])k=j;
      t=a[i];
      a[i]=a[k];
      a[k]=t;
   }
}
main()
{
   int aa[10]={1,2,3,4,5,6,7},i;
   fun(aa,7);
   for(i=0;i<7;i++)printf("%d,",aa[i]);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A. 7,2,5,4,3,6,1,

B. 1,6,3,4,5,2,7,

C. 7,6,5,4,3,2,1,

D. 1,7,3,5,6,2,1,

【答案】A

【解析】函数 fun 的功能是对 a 数组中的偶数位置的元素降序排列。主函数中调用 fun 函数后，将 aa[0]、aa[2]、aa[4]、aa[6]对应的元素降序排列，答案选择 A 选项。

52．有以下程序：

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

程序的运行结果是（　　）。

A. 0123456345

B. 0123453456

C. 0123456666

D．0123454569

【答案】B

【解析】程序的执行过程为：定义数组 m，并为其赋初值，数组长度为 10。调用函数 fun(m,6,3)。函数实现对数组 m[6]~m[9]的重新赋值，令 m[9]=m[6]，m[8]=m[5]，m[7]=m[4]，m[6]=m[3]，for 循环结束之后数组为 m={0,1,2,3,4,5,3,4,5,6}。依次输出数组元素，答案选择 B 选项。

53．有以下程序：

```
#include <stdio.h>
void fun(int *x,int s, int e)
{
   int i,j,t;
   for(i=s,j=e;i<j;i++,j--)
   {
      t=*(x+i);
      *(x+i)=*(x+j);
      *(x+j)=t;
   }
}
main()
{
   int m[]={0,1,2,3,4,5,6,7,8,9},k;
   fun(m,0,3);
   fun(m+4,0,5);
   fun(m,0,9);
   for(k=0;k<10;k++)printf("%d",m[k]);
}
```

程序的运行结果是（　　）。

A．4567890123

B．3210987654

C．9876543210

D．0987651234

【答案】A

【解析】程序的执行过程为：定义数组 m，并为其赋初值，数组长度为 10。调用函数 fun(m,0,3)将数组首地址传入函数，函数实现将数组下标值从 0 到 3 的元素首尾倒置，for 循环结束之后数组为 m={3,2,1,0,4,5,6,7,8,9}。调用函数 fun(m+4,0,4)将数组下标值为 4 的元素地址传入函数，函数实现将数组下标值从 4 到 9 的元素首尾倒置，for 循环结束之后数组为 m={3,2,1,0,9,8,7,6,5,4}。调用函数 fun(m,0,9)将数组首地址传入函数，函数实现将数组下标值从 0 到 9 的元素首尾倒置，for 循环结束之后数组为 m={4,5,6,7,8,9,0,1,2,3}。依次输出数组元素，结果为 4567890123。答案选择 A 选项。

54．有以下程序：

```
#include <stdio.h>
void fun(int *s,int n1,int n2)
{
   int i,j,t;
   i=n1;
   j=n2;
   while(i<j)
   {
```

```
            t=s[i];
            s[i]=s[j];
            s[j]=t;
            i++;
            j--;
        }
    }
    main()
    {
        int a[10]={1,2,3,4,5,6,7,8,9,0},k;
        fun(a,0,3);
        fun(a,4,9);
        fun(a,0,9);
        for(k=0;k<10;k++)
            printf("%d",a[k]);
        printf("\n");
    }
```

程序运行的结果是（　　）。

A．0987654321

B．4321098765

C．5678901234

D．0987651234

【答案】C

【解析】函数 fun 的功能是：将数组 s 中从 n1 至 n2 的元素首尾互换，主函数中，依次对数组 a 的 0～3、4～9、0～9 位进行了三次调换，第一次调用后得到数组 a 为：{4,3,2,1,5,6,7,8,9,0}，第二次调用后得到数组 a 为：{4,3,2,1,0,9,8,7,6,5}，第三次调用后得到数组 a 为：{5,6,7,8,9,0,1,2,3,4}。答案选择 C 选项。

55．有以下程序：

```
#include <stdio.h>
void fun(int *s,int n1,int n2)
{
   int i,j,t;
   i=n1;
   j=n2;
   while(i<j)
   {
      t=*(s+i);
      *(s+i)=*(s+j);
      *(s+j)=t;
      i++;
      j--;
   }
}
main()
{
   int a[10]={1,2,3,4,5,6,7,8,9,0},i,*p=a;
   fun(p,0,3);
   fun(p,4,9);
   fun(p,0,9);
   for(i=0;i<10;i++)
      printf("%d",*(a+i));
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．5678901234

B．0987654321

C．4321098765

D．0987651234

【答案】A

【解析】程序执行过程为：调用函数 fun(p,0,3)，将数组 a 首地址传入函数，循环执行 2 次，将数组前 4 个元素前后倒置，数组 a = {4,3,2,1,5,6,7,8,9,0}；调用函数 fun(p,4,9)，将数组 a 首地址传入函数，循环执行 3 次，将数组第 5 个到第 10 个元素前后倒置，数组 a = {4,3,2,1,0,9,8,7,6,5}；调用函数 fun(p,0,9)，将数组 a 首地址传入函数，循环执行 5 次，将数组元素前后倒置，数组 a = {5,6,7,8,9,0,1,2,3,4}；依次输出数组元素。答案选择 A 选项。

56．有以下程序：

```
#include <stdio.h>
void fun(int *s,int t,int *k)
{
   int p;
   for(p=0,*k=p;p<t;p++)
      if(s[p]>s[*k])*k=p;
}
main()
{
   int a[10] = {11,12,13,14,15,16,20,18,19,10}, k;
   fun(a,10,&k);
   printf("%d,%d\n",k,a[k]);
}
```

程序运行后的输出结果是（　　）。

A. 6,20

B. 10,9

C. 7,20

D. 10,10

【答案】A

【解析】调用函数 fun，将数组的地址、数组长度、变量 k 的地址传入函数。for 函数实现查找数组中最大值，将其下标值存入变量 k 标志的内存单元。输出 k=6，a[6]=20。答案选择 A 选项。

57. 有以下程序：

```
#include <stdio.h>
int fun(char s[])
{
   int n=0;
   while(*s<='9'&&*s>='0')
   {
      n=10*n+*s-'0';
      s++;
   }
   return (n);
}
main()
{
   char s[10]={'6','1','*','4','*','9','*','0','*'};
   printf("%d\n",fun(s));
}
```

程序的运行结果是（　　）。

A. 61490

B. 61

C. 9

D. 5

【答案】B

【解析】f 函数将依次遍历字符数组，将其中的数字字符转换为对应的整数，并按照其出现的顺序将其构造成一个整数，直至遇见非数字字符循环停止。本题中，依次遍历到'6'，'1'，构造成整数 61，循环遇到'*'时结束，返回 61。答案选择 B 选项。

58．设有定义

int x[2][3];

则以下关于二维数组 x 的叙述中错误的是（　　）。

A．x[0]可看作是由 3 个整型元素组成的一维数组

B．x[0]和 x[1]是数组名，分别代表不同的地址常量

C．数组 x 包含 6 个元素

D．可以用语句 x[0]=0;为数组所有元素赋初值 0

【答案】D

【解析】在二维数组中，不能使用 x[0]=0 这样的赋值语句，x[0]代表一个一维数组。答案选择 D 选项。

59．以下定义数组的语句中错误的是（　　）。

A．int num[]={1,2,3,4,5,6};

B．int num[][3]={{1,2},3,4,5,6};

C．int num[2][4]={{1,2},{3,4},{5,6}};

D．int num[][4]={1,2,3,4,5,6};

【答案】C

【解析】A 项，赋值后下标自动取 6；B 项，赋值后下标取 3 行 3 列{{1,2,0}，{3,4,0}，{5,6,0}}；C 项错误，给 2 行 4 列的数组赋于 3 行 4 列的值；D 项，赋值后下标取 2 行 4 列，等价于 num[2][4] = {{1,2,3,4}, {5,6,0,0}}。答案选择 C 选项。

60．以下错误的定义语句是（　　）。

A．int x[][3]={{0},{1},{1,2,3}};

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

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

D．int x[][3]={1,2,3,4};

【答案】C

【解析】通过赋值定义二维数组的大小时，只可以省略第一个方括号中的常量表达式，而不能省略第二个方括号中的常量表达式。答案选择 C 选项。

61．以下数组定义中错误的是（　　）。

A．int x[2][3]={{1,2},{3,4},{5,6}};

B．int x[][3]={0};

C．int x[][3]={{1,2,3},{4,5,6}};

D．int x[2][3]={1,2,3,4,5,6};

【答案】A

【解析】A 项错误，应该是 x[3][2]；B 项正确，对于二维数组，第一维的大小可以省略，计算规则是元素的个数除以第二维的大小向上取整，定义 1 行 3 列的数组，里面的元素全部是零；C 项正确，第一维的大小由所赋初值的行数来决定，定义 2 行 3 列的数组；D 项正确，在给二维数组赋初值时可以不用行花括号对。答案选择 A 选项。

62．若有定义语句：

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

则与该语句等价的是（　　）。

A．int m[ ][3]={{1,2,3},{4,5,6},{7}};

B．int m[ ][3]={{1,2},{3,4},{5,6,7}};

C．int m[ ][3]={{1,2,3},{4,5},{6,7}};

D．int m[ ][3]={{1},{2,3,4},{5,6,7}};

【答案】A

【解析】考查二维数组的初始化。二维数组的初始化有两种方式，①分行初始化，方式为：数据类型数组名

[行下标表达式][列下标表达式]={{第 0 行初值},{第 1 行初值表},…{最后 1 行初值表}};，如果初值表只对部分元素赋初值，没有被赋初值的元素将被自动赋值为 0。②不分行将所有数据依次列在一个花括号里，即数据类型数组名[行下标表达式][列下标表达式]={初值表};，这种方式的赋值就是将初值表的数据依次赋予数组的每个元素，其中赋值是按照数组元素在内存中的位置进行的。题目中的初始化语句是第二种方法，如果用第一种方法应该是 int m[][3] = {{1,2,3}, {4,5,6}, {7}};，这里应该保证除 r 最后一行，每一行都满列有 3 个元素，答案选择 A 选项。

63．若有定义：

int a[2][3];

以下选项中对 a 数组元素正确引用的是（　　）。

A．a[2][!1]

B．a[2][3]

C．a[0][3]

D．a[1>2][!1]

【答案】D

【解析】a[2][3]表示 2 行 3 列的数组，即 a[0][0]，a[0][1]，a[0][2]，a[1][0]，a[1][1]，a[1][2]，行下标最大值为 1，列下标最大值为 2。D 项，其中 1>2 为假即 0，!1 也为 0，即访问 a[0][0]。答案选择 D 选项。

64．有如下程序：

```
#include <stdio.h>
main()
{
   int i,k;
   int array[4][2]={{1,2},{4,9},{6}};
   for(i=0;i<2;i++)
      for(k=0;k<4;k++)
      {
         printf("%d,",array[k][i]);
      }
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．1,2,4,9,6,

B．1,4,6,0,2,9,0,0,

C．2,9,0,0,1,4,6,0,

D．2,9,6,1,4,

【答案】B

【解析】二维数组分行初始化如果初值表只对部分元素赋初值，没有被赋初值的元素将被自动赋值为 0，相当于 array[4][2]={{1,2},{4,9},{6,0},{0,0}}。执行两次 for 语句，实现按列依次输出数组中元素，答案选择 B 选项。

65．有以下程序：

```
#include<stdio.h>
main()
{
   int i,k;
   int array[4][2]={{1,0},{0},{2,9},{3}};
   for(i=0;i<2;i++)
      for(k=0;k<3;k++)
      {
         printf("%d,",array[k][i]);
      }
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A. 1,0,2,0,0,9,

B. 1,2,4,9,6,0,

C. 2,9,0,0,1,4,

D. 1,2,0,1,4,1,

【答案】A

【解析】本题定义一个 4 行 2 列数组，其中赋值 6 个数，其余自动赋值为 0，根据 for 循环，第一次输出第一列前三行数，分别是 1、0、2；第二次输出第二列前三行数，分别是 0、0、9，答案选择 A 选项。

66. 有以下程序

```
#include<stdio.h>
main()
{
   int b[3][3] = {0,1,2,0,1,2,0,1,2},i,j,t = 1;
   for(i = 0;i<3;i++)
      for(j = i;j <= i;j++)t+=b[i][b[j][i]];
   printf("%d\n",t);
}
```

程序运行后的输出结果是（　　）。

A. 4

B. 3

C. 1

D. 9

【答案】A

【解析】本题中，两层 for 循环等价于 t = t + b[0][b[0][0]] + b[1][b[1][1]] + b[2][b[2][2]] = 1+ b[0][0] + b[1][1] + b[2][2] = 1+ 0+ 1+ 2 = 4。答案选择 A 选项。

67. 有以下程序

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

程序运行后的输出结果是（　　）。

A．8,7,3,1,

B．1,6,5,7,

C．4,7,5,2,

D．1,6,2,1,

【答案】B

【解析】先对二维数组的每列从小到大排序，然后输出对角线元素。答案选择 B 选项。

68．有以下程序

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

程序运行后的输出结果是（　　）

A．6,4,3,2,

B．6,2,1,1,

C．1,1,2,6,

D．2,3,4,6,

【答案】A

【解析】第一个嵌套的 for 循环功能将对角线上的数据从大到小排序，其他数据不变，排序完成后，a[0][0]

值为 6，输出第一行，答案选择 A 选项。

69．有以下程序

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

程序执行后的输出结果是（　　）。

A．3,5,7,

B．7,5,3,

C．3,6,9,

D．7,5,1,

【答案】A

【解析】依次输出 t[2][0]，t[1][1]，t[0][2]。答案选择 A 选项。

70．有以下程序：

```
#include<stdio.h>
main()
{
   char a[4][4]={' '};
   int i,j;
   for(i=0;i<4;i++)
   {
      a[i][0]=a[i][3]='#';
      for(j=1;j<3;j++)
      {
         a[0][j]=a[3][j]='#';
         if((i!=0)&&(i!=3))a[i][j]='o';
      }
   }
   for(i=1;i<3;i++)
   {
      for(j=0;j<4;j++)
         printf("%2c",a[i][j]);
      printf("\n");
   }
}
```

程序的运行结果是（　　）。

A．　# o o #<换行># o o #

B．　# # # #<换行># o o #

C．　# o o #<换行># # # #

D．　# # # #<换行># # # #

【答案】A

【解析】程序的执行过程为：定义 4 行 4 列二维数组字符 a 并且初始化为 a[0][0]=' '，其他元素均为 0。执行 for 循环将数组第一列和第四列赋值为'#'。执行 for 循环将数组第一行和第四行的第二个到第三个元素赋值为'#'，

然后将剩余元素全部赋值为'o'。此时矩阵为{{#,#,#,#},{#,o,o,#},{#,o,o,#},{#,#,#,#}}。之后输出矩阵第二行与第三行元素，答案选择 A 选项。

71．若有定义语句：

char s[3][10],(*k)[3],*p;

则以下赋值语句正确的是（　　）。

A．p=s;

B．p=k;

C．p=s[0];

D．k=s;

【答案】C

【解析】A 项，将字符型二维数组首地址赋值给了一个字符型指针，类型不匹配；B 项，将指针数组的首地址赋值给一个字符型指针，类型不匹配；C 项，s 是二维字符数组，s[0]表示一个含有 10 个元素的一维数组，即将一维字符数组首地址赋值给了一个字符型指针，这是允许的；D 项，将二维字符数组赋值给了指向一维字符数组的指针，类型不匹配。答案选择 C 选项。

72．有定义语句：

int *p[4];

以下选项中与此语句等价的是（　　）。

A．int p[4];

B．int **p;

C．int *(p[4]);

D．int (*p)[4];

【答案】C

【解析】题目中定义语句的含义是定义了一个包含 4 个整数指针元素的数组 p。A 项，仅定义了一个含有 4 个整数元素的数组；B 项，定义了一个指向整数指针的指针；D 项，声明了一个指针变量，它指向的是一个含 4 个元素的一维数组。答案选择 C 选项。

73．若有定义

int(*pt)[3];

则下列说法正确的是（　　）。

A．定义了基类型为 int 的三个指针变量

B．定义了基类型为 int 的具有三个元素的指针数组 pt

C．定义了一个名为*pt、具有三个元素的整型数组

D．定义了一个名为 pt 的指针变量，它可以指向每行有三个整数元素的二维数组

【答案】D

【解析】int(*pt)[3];语句定义了一个指向一维数组的指针 pt，该一维数组具有三个 int 型元素。D 项，按照 C 语言中二维数组的定义知，二维数组先进行行排列，再进行列排列，故 pt 也可以指向每行有三个整数元素的二维数组。答案选择 D 选项。

74．若有定义语句：

int a[4][10],*p,*q[4];

且 0≤i<4，则错误的赋值是（　　）。

A．p=a

B．q[i]=a[i]

C．p=a[i]

D．p=&a[2][1]

【答案】A

【解析】二维数组名是指向指针的指针，所以 a 和 q 都为指向指针的指针，而 p 为指向 int 类型的指针，p 和 a 不同类型，故 A 选项中 p=a 赋值语句错误。其余选项可以正确赋值，其中 D 项是用取地址符&返回整数的

地址，然后赋值给 p。所以答案选择 A 选项。

75．若有定义语句：

int a[2][3],*p[3];

则以下语句中正确的是（　　）。

A．p[0]=&a[1][2];

B．p[0]=a;

C．p=a;

D．p[1]=&a;

【答案】A

【解析】声明了 a 是一个二维数组，p 是长度为 3 的 int *数组。A 项正确，p[0]是 int 类型指针，可以指向二维数组中的元素。B 项错误，p[0]是 int 类型指针，不能指向二维数组；C 项错误，p 是 int 类型指针的数组，一旦定义，不能再次赋值；D 项错误，p[1]是 int 类型的指针，&a 是二维数组的地址。答案选择 A 选项。

76．若有定义：

int w[3][5];

则以下不能正确表示该数组元素的表达式是（　　）。

A．*(w+1)[4]

B．*(*w+3)

C．*(*(w+1))

D．*(&w[0][0]+1)

【答案】A

【解析】a 数组元素可用五种表达式来引用：①a[i][j]；②*(a[i]+j)；③*(*(a+i)+j)；④(*(a+i))[j]；⑤*(&a[0][0]+N*i+j)。A 项，C 语言中规定，方括号的优先级高于*，相当于*((w+1)[4])，不能表示数组元素。B 项，*w 是 w[0]的地址，*w+3 是 w[0][3]的地址，表示数组元素 w[0][3]；C 项，*(w+1)是 w[1]的地址，表示数组元素 w[1][0]；D 项，&w[0][0]是元素 w[0][0]的地址，&w[0][0]+1 是元素 w[0][1]的地址，表示数组元素 w[0][1]。答案选择 A 选项。

77．设有以下说明和语句：

int x[3][4],(* p)[4];p=a;

则与表达式*(*p+2)等价的选项是（　　）。

A．a[0][2]

B．*(a+2)[0]

C．(*a+2)[0]

D．a[2][0]

【答案】A

【解析】(*p)[4]是指向有 4 个元素的数组指针，a 为二位数组首地址，赋值给数组指针 p。题目中表达式*(*p+2)是对数组 a 第 1 行第 3 列元素 a[0][2]的引用。a[0][2]是数组第 1 行第 3 列元素，与题目中等价，A 项正确。*(a+2)[0]是对数组第 3 行第 1 列元素的引用，B 项错误。(*a+2)[0]是对数组第 1 行第 3 列元素的引用，C 项错误。a[2][0]是数组第 3 行第 1 列元素，D 项错误。答案选择 A 选项。

78．有以下程序

```
#include<stdio.h>
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（　　）。

A.　2　0　4

B.　2　0　0

C.　2　4　0

D.　2　4　6

【答案】A

【解析】在二维数组 x 中，x[0]，x[1]，x[2]其值依次为二维数组每行第一个元素的地址，其基类型就是数组元素的类型，即 x[0]=&[0][0]，x[1]=&x[1][0]，x[2]=&x[2][0]，完成输入后，数组 x={{2,0},{4,0},{6,0}}。答案选择 A 选项。

79．有以下程序：

```
#include<stdio.h>
main()
{
   int x[3][3] = {{2},{4},{6}}, i, *q = &x[0][0];
   for(i=0;i<2;i++)
   {
      if(i==0)
         x[i][i+1]=*q+1;
      else
         ++q;
      printf("%d",*q);
   }
   printf("\n");
}
```

程序的运行结果是（　　）。

A．23

B．26

C．33

D．36

【答案】A

【解析】题中把数组首元素地址赋值给指针变量 q，在 for 循环体内，当 i=0 时，满足 if 语句，把指针 q 指向的存储单元的值加 1，即将 3 赋给 x[0][1]，接着输出 q 指针指向存储单元的值 2；i=i+1 后再次进人循环体，此时 i 不等于 0，指针 q 地址加 1，指向 x[0][1]，此时*q=3，因此最后函数输出的分别是 2 和 3。答案选择 A 选项。

80．有以下程序：

```
#include<stdio.h>
main()
{
   int s[3][2] = {1,2,3,4,5,6}, *ps[3],k;
   for(k=0;k<3;k++)
   {
      ps[k]=s[k];
      printf("%d",*(ps[k]+1));
   }
}
```

程序的运行结果是（　　）。

A．246

B. 135
C. 123
D. 456
【答案】A
【解析】程序执行过程为：定义二维数组 s 与指针数组 ps，for 循环实现将二维数组每一行的首地址赋值给对应的指针数组元素，然后输出二维数组每行的第二个元素。程序的运行结果是 246。答案选择 A 选项。

81. 有以下程序：

```
#include<stdio.h>
main()
{
   int x[3][4]={1,3,5,7,9,11,2,4,6,8,10,12};
   int (*p)[4]=x,k=1,m,n=0;
   for(m=0;m<2;m++)n+=*(*(p+m)+k);
   printf("%d\n",n);
}
```

程序的运行结果是（　　）。
A. 10
B. 20
C. 14
D. 16
【答案】C
【解析】(*(p+m)+k)等价于 x[m][k]，因此，在 for 循环中，n+=*(*(p+m)+k);执行的是将数组 x[0][1]和 x[1][1]累加后赋值给 n，最后输出，值为 14。答案选择 C 选项。

82. 有以下程序

```
#include<stdio.h>
main()
{
   int a[3][4] = {1,3,5,7,9,11,13,15,17,19,21,23}, (*p)[4]=a,i,j,k=0;
   for(i=0;i<3;i++)
      for(j=0;j<2;j++) k=k+=*(*(p+i)+j);
   printf("%d\n",k);
}
```

程序运行后的输出结果是（　　）。
A. 99
B. 68
C. 60
D. 108
【答案】C
【解析】定义了二维数组 a，p 是指针，它指向长度为 4 的 int 数组，初始化时，p 执行 a[0]，*(*(p+i)+j)相当于 a[i][j]。main 函数的作用是计算数组 a 中前两列的和，结果是 60。答案选择 C 选项。

83. 有以下程序：

```
#include <stdio.h>
#define N 4
void fun(int a[][N])
{
   int i;
   for(i=0;i<N;i++)
      a[0][i]=a[N-1][N-1-i];
}
main()
{
   int x[N][N] = {{1,2,3,4}, {5,6,7,8}, {9,10,11,12}, {13,14,15,16}} ,i;
   fun(x);
   for(i=0;i<N;i++)
      printf("%d,",x[i][i]);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A. 16,6,11,16,

B. 1,6,11,16,

C. 4,7,10,13,

D. 17,17,17,17,

【答案】A

【解析】fun 函数给数组的第一列重新赋值，主函数调用 fun(x)后，x[0][0] = x[3][3] = 16，x[0][1] = x[3][2] = 15，x[0][2] = x[3][1] = 14，x[0][3] = x[3][0] = 13.程序最后输出 x[0][0]，x[1][1]，x[2][2]，x[3][3]，答案选择 A 选项。

84．有以下程序：

```
#include <stdio.h>
#define N 4
void fun(int a[][N])
{
   int b[N][N],i,j;
   for(i=0;i<N;i++)
      for(j=0;j<N;j++)
         b[i][j]=a[N-1-j][i];
   for(i=0;i<N;i++)
      for(j=0;j<N;j++)
         a[i][j]=b[i][j];
}
main()
{
   int x[N][N] = {{1,2,3,4}, {5,6,7,8}, {9,10,11,12}, {13,14,15,16}},i;
   fun(x);
   fun(x);
   for(i=0;i<N;i++)
      printf("%d,",x[i][i]);
   printf("\n");
}
```

程序的运行结果是（　　）。

A．16,11,6,1,

B．1,6,11,16,

C．4,7,10,13,

D．13,10,7,4,

【答案】A

【解析】程序的执行过程为：调用函数 fun，将二维数组 x 地址传入函数，此函数实现将矩阵转置，然后将每一列首尾倒置，调用结果为 x[N][N] = {{13,9,5,1}, {14,10,6,2}, {15,11,7,3}, {16,12,8,4}}。再次调用函数 f，调用结果为 x[N][N] = {{16,15,14,13}, {12,11,10,9}, {8,7,6,5}, {4,3,2,1}}。输出结果对角线元素，答案选择 A 选项。

85．有以下程序：

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

程序运行的结果是（　　）。

A．1,2,3,4,

B．1,0,7,0,

C．1,4,5,9,

D．3,4,8,0,

【答案】B

【解析】数组 x 有 4 列。fun 函数中 for 循环的作用是将二维数组的对角线元素赋给 b[0]，b[1]，b[2]，b[3]。在主函数中，二维数组初始化为{{1,2,3}, {4}, {5,6,7,8}, {9,10}}，对角线元素为 1,0,7,0。答案选择 B 选项。

86．有以下程序：

```
#include<stdio.h>
#define N 3
void fun(int a[ ][N],int b[ ])
{
   int i,j;
   for(i=0;i<N;i++)
   {
      b[i]=a[i][0];
      for(j=i;j<N;j++)
         if(b[i]<a[i][j])b[i]=a[i][j];
   }
}
main()
{
   int x[N][N]={1,2,3,4,5,6,7,8,9},y[N],i;
   fun(x,y);
   for(i=0;i<N;i++) printf("%d,",y[i]);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A. 2,4,8,

B. 3,6,9,

C. 3,5,7,

D. 1,3,5,

【答案】B

【解析】fun 函数的功能是将二维数组 a 中每一行元素的最大值存在数组 b 中，x[3][3] = {1,2,3,4,5,6,7,8,9}，即 x[0] = {1,2,3}，第一行最大值为 3，x[1] = {4,5,6}，第二行最大值为 6，x[2] = {7,8,9}，第三行最大值为 9。答案选择 B 选项。

87. 有以下程序：

```
#include <stdio.h>
#define N 4
void fun(int a[][N],int b[])
{
   int i;
   for(i=0;i<N;i++)
      b[i]=a[i][i]-a[i][N-1-i];
}
main()
{
   int x[N][N] = {{1,2,3,4}, {5,6,7,8}, {9,10,11,12}, {13,14,15,16}}, y[N],i;
   fun(x,y);
   for(i=0;i<N;i++)
      printf("%d,",y[i]);
   printf("\n");
}
```

程序运行后输出的结果是（　　）。

A. 12,-3,0,0,

B. -3,-1,1,3,

C. 0,1,2,3,

D. -3,-3,-3,-3,

【答案】B

【解析】在函数参数传递时，一维数组和二维数组都是以指针的形式。函数 fun 的作用是通过二维数组 a 来给数组 b 赋值，a[i][i]表示 a 上对角线元素，a[i][N-1-i]表示 a 中第 i 行倒数第 i 个元素（从倒数 0 个开始）。所以，b[0] = a[0][0]-a[0][3] = -3，b[1] = a[1][1]-a[1][2] = -1，b[2] = a[2][2]-a[2][1] = 1，b[3] = a[3][3]-a[3][0] = 3。答案选择 B 选项。

88. 有以下程序：

```
#include<stdio.h>
#define N 4
void fun(int a[ ][N],int b[ ])
{
   int i;
   for(i=0;i<N;i++)b[i]=a[i][N-1-i];
}
main()
{
   int x[N][N] = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16}, y[N],i;
   fun(x,y);
   for(i=0;i<N;i++) printf("%d,",y[i]);
   printf("\n");
}
```

程序的运行结果是（　　）。

A. 1,2,3,4,

B. 3,6,9,12,

C. 4,7,10,13,

D. 1,5,9,13,

【答案】C

【解析】函数 fun 的功能是将参数 a 的二维数组中反斜对角对应的数依次赋值给参数 b 的一维数组。调用 fun 函数，y[i] = x[i][N-1-i]，N=4，x 是 4 行 4 列的二维数组。i=0 时，y[0] = x[0][4-1] = x[0][3] = 4；i=1 时，y[1] = x[1][4-1-1] = x[1][2] = 7；i=2 时，y[2] = x[2][4-1-2] = x[2][1] = 10；i=3 时，y[3] = x[3][0] = 13。答案选择 C 选项。

89. 若有以下程序

```
#include <stdio.h>
#define N 4
void fun(int a[][N],int b[],int flag)
{
   int i,j;
   for(i=0;i<N;i++)
   {
      b[i] = a[i][0];
      for(j=1;j<N;j++)
         if (flag?(b[i] > a[i][j]):(b[i] < a[i][j]))
      b[i] = a[i][j];
   }
}
main()
{
   int x[N][N] = {1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12,13,14,15,16}, y[N],i;
   fun(x,y,1);
   for(i=0;i<N;i++) printf("%d,",y[i]);
   fun(x,y,0);
   for (i=0;i<N;i++) printf("%d,",y[i]);
   printf("\n");
}
```

则程序的输出结果是（　　）。

A．1,2,3,4,13,14,15,16,

B．4,8,12,16,1,5,9,13,

C．1,5,9,13,4,8,12,16,

D．13,14,15,16,1,2,3,4,

【答案】C

【解析】考查数组名作为函数参数。函数 fun 的作用是如果 flag 为 1 则用 b[i]记录 a[i][0]至 a[i][3]（即当前行）中最小的值，若 flag 为 0，则用 b[i]记录 a[i][0]至 a[i][3]（即当前行）中最大的值。答案选择 C 选项。

90．有以下程序：

```
#include<stdio.h>
int fun(int(*s)[4],int n,int k)
{
   int m,i;
   m=s[0][k];
   for(i=1;i<n;i++)
      if(s[i][k]>m)
         m=s[i][k];
   return m;
}
main()
{
   int a[4][4] = {{1,2,3,4}, {11,12,13,14}, {21,22,23,24}, {31,32,33,34}};
   printf("%d\n",fun(a,4,0));
}
```

程序运行的结果是（　　）。

A．4
B．34
C．31
D．32
【答案】C
【解析】在 fun 函数中，首先将 s[0][k]的值赋给 m，然后通过 for 循环，遍历 s[1][k]～s[n-1][k]之中的数，将遇到的比 m 大的数保存到 m 中。这是搜索最大值的算法，搜索的是 s[0][k]～s[n-1][k]之间的最大值。所以，主函数中 fun(a,4,0)的功能就是搜索 a[0][0]～a[3][0]之间的最大值，这 4 个值分别是 1、11、21、31，所以最终输出结果是 31。答案选择 C 选项。

91．有以下程序：

```
#include<stdio.h>
void fun(char(*p)[6])
{
   int i;
   for(i=0;i<4;i++) printf("%c",p[i][i]);
   printf("\n");
}
main( )
{
   char s[6][6] = {"ABCDE", "abcde", "12345", "FGHIJ", "fghij", "54321"};
   fun(s);
}
```

程序的运行结果是（　　）。
A．Aa1F
B．Ab3I
C．ABCD
D．fghij
【答案】B
【解析】函数 fun 的功能是输出 s[i][i]，二维数组的对角线元素，所以程序会输出 s[0][0]、s[1][1]、s[2][2]、s[3][3]，即 Ab3I。答案选择 B 选项。

92．有以下程序：

```
#include <stdio.h>
#include <string.h>
void fun(char *s[],int n)
{
   char *t;
   int i,j;
   for(i=0;i<n-1;i++)
      for(j=i+1;j<n;j++)
         if(strlen(s[i])>strlen(s[j]))
         {
            t=s[i];
            s[i]=s[j];
            s[j]=t;
         }
}
main()
{
   char *ss[] = {"bcc", "bbcc", "xy", "aaaacc", "aabcc"};
   fun(ss,5);
   printf("%s,%s\n",ss[0],ss[4]);
}
```

程序的运行结果是（　　）。

A. xy,aaaacc

B. aaaacc,xy

C. bcc,aabcc

D. aabcc,bcc

【答案】A

【解析】函数 fun 的功能：将字符串数组 s 中前 n 个字符串按照字符串的长度由小到大进行排序，要求输出数组的第一个和第五个字符串的内容，答案选择 A 选项。

93. 有以下程序：

```
#include <stdio.h>
#include <stdlib.h>
void fun(int **s, int x[2][3])
{
   **s =*(x[1]+1);
}
void main()
{
   int a[2][3]={1,2,3,4,5,6},*p;
   p = (int *)malloc(sizeof(int));
   fun(&p,a);
   printf("%d\n",*p);
}
```

程序的运行结果是（　　）。

A. 5

B. 2

C. 6

D．3

【答案】A

【解析】malloc()函数的返回值是新分配的存储区的首地址，将该地址赋值给 p 指针，然后取 p 指针的地址，调用 fun 函数。fun 函数中对指向指针的指针 s 指向的内存单元赋值，其中 x[1]表示二维数组 x 的第二行的行地址，x[1]+1 表示二维数组 x 的第二行第二列的地址，所以 fun 函数的功能是将二维数组 x 的第二行第二列的值，即 x[1][1]赋值给**s，*s 为 p，所以**s 代表*p，*p 即为 a[1][1]，答案选择 A 选项。

94．下面是有关 C 语言字符数组的描述，其中错误的是。

A．不可以用赋值语句给字符数组名赋字符串

B．可以用输入语句把字符串整体输入给字符数组

C．字符数组中的内容不一定是字符串

D．字符数组只能存放字符串

【答案】D

【解析】C 语言中，字符数组可以用来存放单个的字符或者字符串。答案选择 D 选项。

95．以下叙述中正确的是（　　）。

A．语句“char str[10]="string!";”和“char str[10]={"string!"};”并不等价

B．对于字符串常量“string!”，系统已自动在最后加入了'\0'字符，表示串结尾

C．对于一维字符数组，不能使用字符串常量来赋初值

D．在语句“char str[]="string!";”中，数组 str 的大小等于字符串的长度

【答案】B

【解析】B 项正确，C 编译程序将自动在字符串常量最后添加'\0'。AC 两项错误，可以直接用字符串常量给一维字符数组赋初值，习惯上均省略花括号；D 项错误，'\0'会占用一个不属于字符串的存储单元，因此数组 str 的长度大于字符串的长度。答案选择 B 选项。

96．以下正确的字符串常量是（　　）。

A．""

B．'abc'

C．Olympic Games

D．"\\\"

【答案】A

【解析】字符串常量需要用双引号括起来，A 项正确；"\\\"中第三个\和"和在一起是一个字符，D 项错误。答案选择 A 选项。

97．以下能正确定义字符串的语句是（　　）。

A．char str[]="\0";

B．char str="kx43";

C．char str=";

D．char str[]={'\064'};

【答案】A

【解析】字符串末尾一定要有结束标记，双引号括起来的字符串系统会默认添加结束标记。A 项正确，定义了空字符串 str。B 项错误，str 是 char 类型变量，不能用字符串为其赋值；C 项错误，字符串应该用双引号括起来；D 项错误，如果一个字符数组用来作为字符串使用，一定要人为加入'\0'，str 字符串中 str[0]='\064'='4'，但是 str 缺少结束符。答案选择 A 选项。

98．设有以下定义：

char s1[]="0123";

char s2[]={'0','1','2','3'};

则以下叙述正确的是（　　）。

A．数组 s1 的长度小于 s2 的长度

B. 数组 s1 和 s2 的长度相同

C. 数组 s1 的长度大于 s2 的长度

D. 数组 s1 和 s2 完全等价

【答案】C

【解析】字符数组 s1 赋值字符串"0123"，字符串中字符依次放入数组中，在最后一个字符后要添加一个结束字符'\0'，数组 s1 长度为 5。数组 s2 只需要 4 个单元即可放下所有初始化值，故数组长度为 4。所以数组 s1 的长度大于 s2 的长度，C 项正确，AB 错误。两个数组等价是指两个数组长度和类型以及对应每个元素值均相同，所以 s1 和 s2 不是完全等价，D 项错误。答案选择 C 选项。

99. 以下选项中，合法的是（　　）。

A. char str3[]={'d', 'e', 'b', 'u', 'g', '\0'}

B. char str4; str4="hello world";

C. char name[10]; name="china";

D. char str1[5]="pass", str2[6]; str2=str1;

【答案】A

【解析】A 项是在赋初值时直接赋字符串常量。B 项，str4 是一个字符型变量，不能直接赋值字符串；C 项，数组名 name 是一个地址常量，不能直接被赋值；D 项，str1 和 str2 代表两个大小不同的字符数组的首地址，不能互相赋值。答案选择 A 选项。

100. 以下能正确进行字符串赋值的语句组是

A. char*ch; ch="abc";

B. char ch[]={'a', 'b', 'c'};

C. char ch[3]= "abc";

D. char ch[4]; ch="abc";

【答案】A

【解析】一个字符串包括字符部分和字符串结束标志。将字符指针变量初始化为一个字符串有两种方式，①通过赋值运算使一个字符指针指向一个字符串常量：char*ch; ch="abc"；②定义一个字符指针变量，并且初始化为一个字符串的首地址：char*ch="abc"。A 项正确。B 项定义一个字符数组并为其赋初值，因为没有字符串结束标志，所以字符数组不是一个字符串，B 项错误。C 项定义字符数组，长度为 3，用字符串"abc"初始化，但是字符串的结束标志由于数组长度不够并未放入数组，所以数组不是一个完整的字符串，C 项错误。D 项正确定义了数组，长度为 4，ch 表示数组首地址，不可以修改，更不能被赋值，D 项错误。答案选择 A 选项。

101. 以下涉及字符串数组、字符指针的程序段，不会产生编译错误的是（　　）。

A. char*str,name[10]; str="Hello World";

B. char*str,name[10]; name="Hello  World";

C. char str1[10]="prog.c", str2[10]; str2=str1;

D. char head_line[]; head_line="== == == == == == =";

【答案】A

【解析】将字符指针变量初始化为一个字符串有两种方式，一种通过赋值运算使一个字符指针指向一个字符串常量：char*str; str="Hello World"，A 选项正确。另一种是定义一个字符指针变量，并且初始化为一个字符串的首地址：char*str="Hello World"。字符指针变量与字符数组都可以用来实现字符串的存储和运算，但二者是有区别的：①数组名是一个地址常量，而字符指针变量是一个变量，所以不能给一个数组名赋值，如 name = "Hello world"错误，B 选项错误；②字符型的指针变量的值是可以改变的；③字符指针变量和字符数组在内存中的存储情形不同，数组指的是按照数组长度分配的若干存储空间，而指针只是一个存储字符串首地址的存储空间，所以相同类型指针可以相互赋值，而数组不能直接用“=”赋值，C 选项错误。不能给数组名赋值.Head_line="== == == == == =="不合法，只有在数组初始化时可以直接赋值，如 char head_line[]="==== == == == == ="合法，并且定义数组时必须指定数组长度，char head_line[]不合法，故 D 选项错误。答案选择 A 选项。

102. 有以下程序：

```
#include <stdio.h>
#include <string.h>
main()
{
   char str[12]={'s', 't', 'r', 'I', 'n', 'g'};
   printf("%d\n",strlen(str));
}
```

程序运行后的输出结果是（　　）。

A．6

B．7

C．11

D．12

【答案】A

【解析】str[12]是字符数组，初始化时未赋初始值的自动赋值'\0'，即 str[12]={'s', 't', 'r', 'i', 'n', 'g', '\0', '\0', '\0', '\0', '\0', '\0'}。strlen 函数求字符串长度，统计长度时以'\0'结束，即统计到长度为 6 时结束，strlen 函数返回值为 6。答案选择 A 选项。

103．有以下程序：

```
#include <stdio.h>
main()
{
   char s[] = "012xy\08s34f4w2";
   int i,n = 0;
   for(i = 0;s[i]!='0';i++)
      if(s[i]>'0'&& s[i]<='9')n++;
      printf("%d\n",n);
}
```

程序运行后的输出结果是（　　）。

A．0

B．3

C．7

D．8

【答案】A

【解析】此题有陷阱，考查的是 for 的执行顺序，①i=0，初始化初值；②s[i]!='0'进行判断，如果条件为真，则继续执行；③执行循环体代码；④i++变量 i 自增；⑤回到第②步，一直循环下去，直到第②步为假，退出循环。本题 i= 0 时，s[i]=='0'，for 循环终止，接着执行 printf 语句，n 的值没有发生变化，故输出 0，答案选择 A 选项。

104．有以下程序：

```
#include <stdio.h>
main()
{
   char s[]={"012xy"};
   int i,n=0;
   for(i=0;s[i]!='\0';i++) if(s[i]>='a'&&s[i]<='z')n++;
   printf("%d\n",n);
}
```

程序运行后的输出结果是（　　）。

A．0

B．2

C．3

D．5

【答案】B

【解析】程序中 main 函数的作用就是判断字符串 s 中小写字母的个数，显然结果为 2。答案选择 B 选项。

105．有以下程序：

```
#include <stdio.h>
main()
{
   char name[10] = {'S','T','R'};
   name[2]='#';
   name[6]=0;
   printf("%s\n",name);
}
```

程序运行后的输出结果是（　　）。

A．ST#

B．STR#

C．STR#0

D．STR0

【答案】A

【解析】程序首先对 name 数组进行初始化，再对元素 name[2]赋值为'#'，由于用“%s”格式输出字符串时，以'\0'作为结束标志，'\0'对应于 0，在输出时，系统认为 name[2]后面已经结束，最后输出 ST#。答案选择 A 选项。

106．有如下程序：

```
#include <stdio.h>
main()
{
   char name[10] = {'S','T','R','I','N','G'};
   name[3]='E';
   name[5]=0;
   printf("%s\n",name);
}
```

程序运行后的输出结果是（　　）。

A．STRENG

B．STRIEG

C．STREN

D．STREN0

【答案】C

【解析】printf 函数按照格式符说明输出对应的数据。%s 控制符用于输出字符串，输出时从给定地址开始依次输出字符，直到遇到'\0'结束。给字符变量赋值 0，相当于赋值'\0'。初始化时 name = "STRING"，改动数组第 4 个和第 6 个元素之后字符串变为"STREN"。调用 printf 输出字符串为 STREN，答案选择 C 选项。

107．有以下程序：

```
#include <stdio.h>
main()
{
   int i,j=0;
   char a[] = "How are you!",b[10]={0};
   for(i=0;a[i];i++)
      if(a[i]==' ')
         b[j++]=a[i-1];
   printf("%s\n",b);
}
```

程序运行后的输出结果是（　　）。

A. we

B. How are you!

C. ay

D. we!

【答案】A

【解析】for 语句循环体执行功能是，每次当 a[i]元素为' '时，自动将 a[i-1]上一个单元内容赋值给 b[j]，赋值完成后，j 值加 1，因此 a[i]的'w'和'e'分别赋值给 b[0]，b[1]。答案选择 A 选项。

108．有以下程序：

```
#include <stdio.h>
main()
{
   int i,j=0;
   char a[] = "How are you",b[10]={0};
   for(i=0;a[i];i++)
      if(a[i]==' ')
         b[j++]=a[i+1];
   printf("%s\n",b);
}
```

程序运行后的输出结果是（　　）。

A. Hay

B. How are you

C. we

D. ay

【答案】D

【解析】for 语句循环体执行功能是，每次当 a[i]元素为' '时，自动将 a[i]下一个元素赋值给 b[j]，赋值完成后，j 值加 1，因此 a[i]的'a'和'y'分别赋值给 b[0]，b[1]，输出结果为 ay。答案选择 D 选项。

109．以下选项中正确的语句组是（　　）。

A. char s[];s="BOOK!";

B. char *s;s={"BOOK!"};

C. char s[10];s="BOOK!";

D. char *s;s="BOOK!";

【答案】D

【解析】AC 两项错误，因为字符串常量在赋值过程中给出的是这个字符串在内存中所在的一串连续存储单元的首地址，即 s 是数组首地址，并非字符串变量，故错误；B 项，字符串变量的赋值方式有 char *s;s="BOOK!";。数组赋值的方式有：char s[];s={"BOOK!"};或是 char s[];s="BOOK!";。s 为字符串变量，赋值方式应该是前者，故

错误。D 中先定义一个字符串变量，然后将变量指向一个字符串常量，语句正确。答案选择 D 选项。

110．以下使指针指向一个字符串的选项错误的是（　　）。

A．char str[]="string"; char * ps; *ps= str;

B．char str[]="string"; char *ps; ps= str;

C．char str[]="strinh",* ps=str;

D．char * ps; ps=str; ps="strinh";

【答案】A

【解析】A 项中定义了指针 ps 后，*ps 表示指针 ps 指向地址的内容，而不是地址，*ps=str 无法将 str 的首地址赋给一个常量，语法错误。答案选择 A 选项。

111．下列语句中，正确的是（　　）。

A．char *s; s = "Olympic";

B．char s[7]; s= "Olympic";

C．char *s; s= {"Olympic"};

D．char s[7]; s= {"Olympic"};

【答案】A

【解析】A 项，将存放字符串常量的无名存储区的首地址赋给指针变量；BD 两项错误，C 语言中不可以用赋值语句对数组名一字符串，数组名代表数组首地址，不可更改，可以逐个给数组元素赋字符值，并在最后加入字符串结束标志；C 项，赋值格式错误。答案选择 A 选项。

112．下面选项中的程序段，没有编译错误的是（　　）。

A．char *sp,s[10]; sp = "Hello";

B．char *sp,s[10]; s = "Hello";

C．char str1[10] = "conputer", str2[10]; str2=str1;

D．char mark[]; mark = "PROGRAM";

【答案】A

【解析】A 项正确，首先定义了字符串指针 sp，然后给 sp 赋值字符串常量"Hello"。BCD 三项错误，数组只有在初始化是可以被赋值，不能通过赋值语句将字符串常量或其他字符数组中的字符串直接赋给数组名。答案选择 A 选项。

113．以下叙述中正确的是（　　）。

A．不能用字符串常量对字符数组名进行整体赋值操作

B．字符串常量"Hello"会被隐含处理成一个无名字符型数组，它有 5 个元素

C．“char str[7] = "string!";”在语法上是合法的，运行也是安全的

D．“char *str = "Hello";”与“char  str[]; str = "Hello";”效果是一样的

【答案】A

【解析】A 项正确，不可以用赋值语句给字符数组整体赋一串字符，但是可以直接用字符串常量给一维字符数组赋初始值。B 项错误，编译器会在字符串常量最后添加'\0'，它共有 6 个元素；C 项错误，在语法上是合法的，但是没有字符串尾标识符，系统将要在其后的内存中找一个距它最近的'\0'作为其结束标志，运行时不安全；D 项错误，不能给数组名 str 赋值。答案选择 A 选项。

114．设有如下程序段：

char s[20]="Bejing",*p;

p=s;

则执行 p=s;语句后，以下叙述正确的是（　　）。

A．可以用*p 表示 s[0]

B．s 数组中元素的个数和 p 所指字符串长度相等

C．s 和 p 都是指针变量

D．数组 s 中的内容和指针变量 p 中的内容相等

【答案】A

【解析】A 项正确，p=s;后，指针 p 指向 s 的首地址，*p=s[0]；B 项错误，p 是字符串指针所指向的字符串为"Bejing"，其长度为 7，而数组 a 中元素的个数为 20；C 项错误，s 为字符数组名；D 项，数组 s 中有 20 个元素，其中 s[6]～s[19]都为 0，p 中只有 7 个元素。答案选择 A 选项。

115．设有定义：

char *c;

以下选项中能够使字符型指针 c 正确指向一个字符串的是（　　）。

A．char str[]="string";c=str;

B．scanf("%s",c);

C．c=getchar();

D．*c="string";

【答案】A

【解析】B，C 选项均为输入函数，其表达意思为输入字符串 c。B 项中 scanf()函数是将 c 定义为一个字符数组的数组名；C 项中是将 c 定义为一个字符型变量；D 项中是需要在指针定义时为它赋值，因此是不合法的，所以 BCD 三项都不正确。A 项定义字符数组 str，再将字符数组 str 的首地址赋给字符型指针 c，正确。所以答案选择 A 选项。

116．有以下说明语句：

char *s = "\"Name\\Address\n";

指针 s 所指字符串的长度是（　　）。

A．17

B．15

C．14

D．说明语句不合法

【答案】C

【解析】以“\”开头的转义字符也是字符常量。用一对双引号括起来的多个字符为字符串。“\"”是“"”的转义字符，“\\”是“\”的转义字符，“\n”是换行的转义字符，指针 s 指向的字符串中字符依次是："、N、a、m、e、\、A、d、d、r、e、s、s、\0。所以字符串长度为 14。答案选择 C 选项。

117．若有说明和语句：

char str[]="Hello", *p; p=str;

则此时*(p+5)中的值为（　　）。

A．'\0'

B．'o'

C．'o'的地址

D．不确定的值

【答案】A

【解析】定义字符数组并为其初始化，str 数组前 5 个元素为 Hello，第六个元素自动赋值为'\0'。定义指针使等于字符串首地址，p+5 指向字符串第六个元素。答案选择 A 选项。

118．有以下程序

```
#include<stdio.h>
main()
{
   char s[]="rstuv";
   printf("%c\n",*s+2);
}
```

程序运行后的输出结果是（　　）。

A．tuv
B．字符 t 的 ASCII 码值
C．t
D．出错
【答案】C
【解析】"*"的优先级高于"+"，因此先对 s 取内容，然后将 s 中的内容加 2。s=&s[0]，*s 是字符 r，*s+2 表示字符 r 后的第二个字符 t。答案选择 C 选项。

119．有以下程序

```
#include<stdio.h>
main()
{
   char ch[]="uvwxyz",*pc;
   pc=ch;
   printf("%c\n",*(pc+5));
}
```

程序运行后的输出结果是（　　）。
A．0
B．z
C．元素 ch[5]的地址
D．字符 y 的地址
【答案】B
【解析】将字符数组的首地址赋给 pc，那么 pc = &ch[0]，pc+5 = &ch[5]，*(pc+5)等价于 ch[5]。答案选择 B 选项。

120．有以下程序（字符 a 的 ASCII 码值为 97）

```
#include <stdio.h>
main()
{
   char *s={"abc"};
   do
   {
      printf("%d",*s%10);
      ++s;
   }while(*s);
}
```

程序运行后的输出结果是（　　）。
A．abc
B．789
C．7890
D．979899
【答案】B
【解析】a、b、c 的 ASCII 值分别为 97、98、99。程序中执行输出 s 中字符对应的 ASCII 码与 10 进行模运算后的值，s 是一个指针，首先指向字符 a，先执行 97%10，结果为 7；然后++s，指针指向下一个字符 b，执行 98%10，结果为 8，直到 s 所指为空，故最后输出的结果为 789。答案选择 B 选项。

121．有以下程序：

```
#include<stdio.h>
main()
{
   char s[10] = "verygood", *ps = s;
   ps += 4;
   ps = "nice";
   puts(s);
}
```

程序的运行结果是（　　）。

A. nice

B. verynice

C. nicegood

D. verygood

【答案】D

【解析】通过赋值运算使一个字符指针指向一个字符串常量：char*ps; ps="nice";这种形式只是将字符串"nice"首地址赋给指针，而不是将指针原本指向的单元元素改变成"nice"。程序执行过程为：定义字符数组并且初始化为"verygood"常量，定义指针 ps 使其指向数组，使 ps 指向数组第 5 个元素，使指针指向字符串"nice"，这并不影响 s 数组任何元素值，故输出字符串 s 为"verygood"。答案选择 D 选项。

122．有如下程序：

```
#include<stdio.h>
int disp(char *str)
{
   while(*str) putchar(*str++);
   return *str;
}
main()
{
   printf("%d\n",disp("NAME"));
}
```

程序运行后的输出结果是（　　）。

A. NAME0

B. NAMEE

C. NAME

D. NAME\0

【答案】A

【解析】程序执行过程为：调用函数 disp，将字符串“NAME”首地址传给指针 str，在 while 循环中，依次判断字符串中字符是否为'\0'，不是'\0'则输出字符，否则结束循环。当(*str) = '\0'时，结束循环，返回字符'\0'的 ASCII 码 0，并输出。程序运行后的输出结果是：NAME0。答案选择 A 选项。

123．有以下程序：

```
#include<stdio.h>
int disp(char *str)
{
   while(*str) putchar(*str++);
   putchar('#');
   return *str;
}
main()
{
   printf("%d\n",disp("C##123"));
}
```

程序运行后的输出结果是（　　）。

A. C##123#0

B. C##1230

C. C##0

D. C##123#\0

【答案】A

【解析】disp 函数中，存在 while 循环，当指针指向地址单元存储的字符不等于空字符'\0'，输出此字符，否则退出循环，输出'#'。在主函数中调用 disp 函数时，字符串"C##123"先全部输出，当指针指向字符串最后的空字符时，退出循环，输出'#'，并且再将'\0'返回输出。由于 disp 函数返回类型为 int，所以返回'\0'的 ASCII 码 0，答案选择 A 选项。

124．有以下程序：

```
#include <stdio.h>
int fun(char *s)
{
   char *p=s;
   while(*p++!='\0');
   return(p-s);
}
main()
{
   char *p="01234";
   printf("%d\n",fun(p));
}
```

程序的运行结果是（　　）。

A. 6

B. 5

C. 4

D. 3

【答案】A

【解析】程序执行过程为：定义字符串指针 p 并为其初始化为"01234"，调用函数 fun(p)，将指针传入函数。fun 函数功能即返回字符串首地址与结束符下一个地址之差，也即是字符串长度加 1。输出地址差为 6，答案选择 A 选项。

125．有以下程序：

```
#include<stdio.h>
main()
{
   int password;
   char *p,old_str[10]="wind";
   scanf("%d",&password);
   p = old_str;
   while(*p)
   {
      printf("#%c",*p+password);
      p++;
   }
}
```

程序运行时，从键盘输入 2<回车>，输出结果是（　　）。

A. #y#k#p#f

B. #wi#nd#

C. xj#oe

D. #2222#

【答案】A

【解析】首先定义了一个指针 p，使其指向数组的首地址，在 while 语句中，如果当前指针指向地址单元的字符不等于空字符'\0'，则输出'#'和指针对应地址元素的值在 ASCII 码加 2 后变换的字符常量，答案选择 A 选项。

126．有以下程序：

```
#include <stdio.h>
main()
{
   char s1[]="programe",s2[]="Language";
   char *p1=s1,*p2=s2;
   int k;
   for(k=0;k<8;k++)
      if(*(p1+k)==*(p2+k))
         printf("%s ",(p1+k));
}
```

程序的运行结果是（　　）。

A. grame ame e

B. g a e

C. programe

D. 无输出字符

【答案】A

【解析】定义两个指针变量 p1、p2，分别指向数组 s1、s2 首地址，在 for 循环中，比较两个指针对应地址的元素是否相等，如果相等，则输出当前指针指向地址的字符串，当 k=3 时，*(p1+k)和*(p2+k)相等都为'g'，输出 p1+k 对应的字符串，即"grame"；当 k=5 时，输出"ame"；当 k=7 时，输出"e"。答案选择 A 选项。

127．若要求从键盘读入含有空格字符的字符串，应使用函数（　　）。

A. gets()

B. getc()

C. getchar()

D. scanf()

【答案】A

【解析】A 项，gets 函数用来从终端键盘读入字符串（包括空格符），直到读入一个换行符为止。B 项错误，C 语言标准库中没有 getc 函数；C 项，getchar 函数是从键盘读入单个字符，空格、回车符都将作为字符读入；D 项，scanf 函数用%s 格式符输入字符串时，空格和回车符都作为输入数据的分隔符而不能被读入。答案选择 A 选项。

128．设有定义：

char s[81];int i=0;

以下不能将一行（不超过 80 个字符）带有空格的字符串正确读入的语句或语句组是（　　）。

A．gets(s);

B．while((s[i++]=getchar())!='\n');s[i]= '\0';

C．scanf("%s",s);

D．do{scanf("%c",&s[i]);}while(s[i++]!='\n'); s[i]='\0';

【答案】C

【解析】字符串的输入有两种方式：①scanf()函数；②get()函数。A 项，gets 函数用来从终端键盘读入字符串（包括空格符），直到读入一个换行符为止；B 项，getchar()函数从终端读入一个字符作为函数值；D 项，%c 格式读入单个字符，空格、回车符和 Tab 键都将作为字符读入。C 项中，s 代表输入一个字符数组而非地址，而且遇到空格时会默认字符串输入结束，所以不能读入带有空格的字符串。答案选择 C 选项。

129．以下不能将键盘输入的字符串：This is a string<回车>读入到 str 中的程序段是（　　）。

A．char str[80]; scanf("%s",str);

B．char str[80]; int i=0; while((str[i++]=getchar())1='\n'); str[i]=0;

C．char str[80]; gets(str);

D．char str[80], *ps=str; do{ scanf("%c",ps); }while(* ps++!='\n'); *(ps)=0;

【答案】A

【解析】在使用 scanf 函数时，在输入数据时，遇到空格，或按“回车”键或按“跳格”（Tab）键时该数据认为结束。A 项中输入 This 后有一个空格，认为数据输入结束，以后的数据不能再读入到 str 中。B 项用 getchar 读人字符，键盘输入的所有字符均被认为是有效字符读入，用 while 循环控制读入，当读入字符为回车时结束读入，字符串被正确的读入到 str 中。C 项用 gets 函数读入字符串，它读入键盘输入的所有字符，遇到回车自动结束读入，所以也能正确读入字符串到 str 中。D 项用 do…while 和指针循环控制字符输入，依次判断输入字符是否为回车符，若不是回车符则放人字符数组中，直到读入回车符，结束循环，数组 str 中正确存放人指定的字符串。答案选择 A 选项。

130．有定义语句：

char s[10];

若要从终端给 s 输入 5 个字符，错误的输入语句是（　　）。

A．gets(&s[0]);

B．scanf("%s",s+1);

C．gets(s);

D．scanf("%s",s[1]);

【答案】D

【解析】采用 scanf 函数输入时，输入项为变量的地址。gets 函数的输入项为存放字符串的首地址。A 项，&s[0]为数组的首地址；B 项，s+1 为数组中第二个元素的地址；C 项，s 也为数组的首地址；D 项，s[1]不是地址。答案选择 D 选项。

131．有以下程序

```
#include <stdio.h>
main()
{
   char a[20],b[20],c[20];
   scanf("%s%s",a,b);
   gets(c);
   printf("%s%s%s\n",a,b,c);
}
```

程序运行时从第一列开始输入：

This is a cat!<回车>

则输出结果是（　　）。

A. Thisisacat!

B. Thisis a

C. Thisis a cat!

D. Thisisa cat!

【答案】C

【解析】用 scanf()函数的%s 格式符输入字符串时，空格和回车符都作为输入数据的分隔符而不能被读入；而 gets 函数可以一次接收一行输入字符串，其中可以有空格。在本题中，字符数组 a 的赋值到输入的第一个空格即结束，a 的值为“This”，同理 b 的值为“is”；而 gets(c)函数可以接收该行剩余的所有字符，所以 c 被赋值为“a cat!”。故输出的结果为 Thisis a cat!。答案选择 C 选项。

132．有以下程序：

```
#include <stdio.h>
main()
{
   char a[30],b[30];
   scanf("%s",a);
   gets(b);
   printf("%s\n%s\n",a,b);
}
```

程序运行时若输入：

how are you? I am fine<回车>

则输出结果是（　　）。

A. how are you?<换行> I am fine

B. how<换行> are you? I am fine

C. how are you? I am fine

D. how are you?

【答案】B

【解析】scanf 语句接收字符串时遇到空格就认为字符串读入结束，但是 gets 函数遇到回车才认为结束。本题中将第一个空格前面的部分赋值给 a，将后面的部分赋值给 b。答案选择 B 选项。

133．有以下程序：

```
#include <stdio.h>
main()
{
   char a,b,c,d;
   scanf("%c%c",&a,&b);
   c=getchar();
   d=getchar();
   printf("%c%c%c%c\n",a,b,c,d);
}
```

当执行程序时，按下列方式输入数据（从第一列开始，<CR>代表回车，注意：回车是一个字符）

12<CR>

34<CR>

则输出结果是（　　）。

A. 1234

B. 12

C. 12<CR>3

D. 12<CR>34

【答案】C

【解析】scanf()函数的一般调用形式为：scanf（格式控制，输入地址列表）。其中，格式控制是用双引号括起来的字符串，包括格式字符和普通字符，格式是由“%”和格式字符组成。getchar()函数的功能是从标准输入设备上读入一个字符。根据程序中的格式控制可知，接收输入时分别把 1 赋给了 a，2 赋给了 b，然后 getchar()函数提取一个换行符赋给 c，再提取一个字符 3 赋给了 d。所以程序的输出结果为：12<CR>3。答案选择 C 选项。

134．有以下程序

```
#include <stdio.h>
char fun(char *c)
{
   if(*c<='Z'&&*c>='A')
      *c-='A'-'a';
   return *c;
}
main()
{
   char s[81],*p=s;
   gets(s);
   while(*p)
   {
      *p=fun(p);
      putchar(*p);
      p++;
   }
   printf("\n");
}
```

若运行时从键盘上输入 OPEN THE DOOR<回车>，程序的输出结果是（　　）。

A. OPEN THE DOOR

B. OPEN tHE dOOR

C. open the door

D. Open The Door

【答案】C

【解析】在 ASCII 码表中，同一字母的小写编码比大写编码大 32。fun 函数中对大字字母执行*c-= 'A'-'a'，即*c = *c-(-32)，把*c 从大写变成了小写。该程序的功能就是先将数组 s 中的大写字符转变为小写字符，然后输出 s。答案选择 C 选项。

135．有以下程序：

```
#include <stdio.h>
main()
{
   char A,B,C;
   B='1';C='A';
   for(A=0;A<6;A++)
   {
      if(A%2)putchar(B+A);
      else putchar(C+A);
   }
}
```

程序运行后输出的结果是（　　）。

A．1

B．3D5FBABCDFE

C．A2C4E6

D．1123456

【答案】C

【解析】第一次循环 A 的值为 0，A%2 也为 0，if 条件不满足，所以执行 putchar(C+A)，输出字符'A'。第二次 A 的值为 1，条件 A%2 的值为 1，if 条件满足，执行 putchar(B+A)，输出字符'2'。第三次 A 的值为 2，if 条件不满足，执行 putchar(C+A)，输出字符'C'。同理后续打印字符依次为'4'、'E'、'6'。答案选择 C 选项。

136．有如下程序：

```
#include<stdio.h>
main()
{
   char *p,old_str[10]="wind";
   int password;
   scanf("%d",&password);
   p = old_str;
   while(*p)
   {
      printf("%c",*p+password);
      p++;
   }
   printf("\n");
}
```

程序运行时，从键盘输入 2<回车>，输出结果是（　　）。

A．ykpf

B．wind

C．xjoe

D．2222

【答案】A

【解析】程序执行过程为：定义字符数组 str 并且初始化为："wind"，定义 password 并通过 scanf 函数从键盘为其赋值 2。定义指针 p 并使其指向字符串。通过 while 循环，将字符数组中每一个字符的 ASCII 码加 2，并且按字符格式输出每一个字符。程序运行后输出结果是：ykpf。答案选择 A 选项。

137．以下选项中有语法错误的是（　　）。

A．char *str[] = {"guest"};

B．char str[][10] = {"guest"};

C．char *str[3]; str[1] = "guest";

D．char str[3][10]; str[1] = "guest";

【答案】D

【解析】考查指针数组的概念，选项 D 中 str 为二维字符数组，str[1]指的是第 1 行第 0 列的首地址，即 str[1]=&str[1][0]，不能对地址常量赋值。答案选择 D 选项。

138．以下语句中存在语法错误的是（　　）。

A．char ss[6][20];ss[1]="right?";

B．char ss[][20]={"right?"};

C．char*ss[6];ss[1]="right?";

D．char *ss[]={"right?"};

【答案】A

【解析】A 项错误，数组定义后，不能对数组整体赋值，ss 是二维数组，ss[1]是一维字符数组，即字符串，字符串赋值可以使用 strcpy(ss[1],"right");的形式，但不能使用赋值的形式。BD 两项正确，在定义时对数组同时进行初始化；C 项正确，将常量字符串在内存中的首地址赋给指针数组的一个元素。答案选择 A 选项。

139．若有定义：char*ps[]={"aa","bb","cc","dd"};，则以下叙述正确的是（　　）。

A．ps[0]是字符串"aa"

B．*ps[0]是字符串"aa"的首地址

C．ps[0]是字符串"aa"的首地址

D．*ps[0]是字符串"aa"

【答案】C

【解析】定义一个字符串数组指针 ps 后，ps 是指针变量，ps[0]指向的是数组首个元素的地址，即字符串"aa"的首地址，答案选择 C 选项。

140．若有以下程序段

```
char str[4][12] = {"aa","bbb","ccccc","d"}, *strp[4];
int i;
for(i=0;i<4;i++)strp[i]=str[i];
```

不能正确引用字符串的选项是（　　）。

A．*strp

B．str[0]

C．strp[3]

D．strp

【答案】D

【解析】strp 是 char *类型、长度为 4 的数组，4 个指针分别指向字符串数组 str 中的 4 个字符串。D 项错误，strp 是 char *的数组，不能引用字符串。A 项正确，引用 strp 数组中第一个指针指向的内容，即"aa"；B 项正确，str 是字符串数组，引用数组中的"aa"；C 项正确，strp[3]等价于*(strp+3)，访问 strp 中第四个指针指向的内容，即"d"。答案选择 D 选项。

141．有以下程序：

```
#include <stdio.h>
main()
{
    char ch[3][5] = {"AAAA","BBBB","CC"};
    printf("%s\n",ch[1]);
}
```

程序运行后的输出结果是（　　）。

A. AAAA

B. CC

C. BBBCC

D. BBBB

【答案】D

【解析】程序中声明一个 3 行 4 列的字符型数组。要求输出 ch[1]，由于数组下标默认从 0 开始，故 ch[1]为第二行的 BBBB。答案选择 D 选项。

142．有以下程序：

```
#include <stdio.h>
main()
{
    char b[4][10];
    int i;
    for(i=0;i<4;i++)
        scanf("%s",b[i]);
    printf("%s%s%s%s\n", b[0], b[1], b[2], b[3]);
}
```

执行时若输入：Fig flower is red.<回车>，则输出结果是（　　）。

A. Figflowerisred.

B. Figflowefis red.

C. Figflower is red.

D. Fig flower is red.

【答案】A

【解析】scanf 函数从输入设备按照指定的类型输入对应类型的若干个数据，遇到空格、制表符和回车时读取结束。可知读到第一个空格时 b[0]赋值结束，为"Fig"，从之后的第一个有效字符'f'读到第二个空格时 b[1]赋值结束，为"flower"，可知 b[2]为 is，b[3]为"red."。printf 函数格式控制符%s 输出字符串，遇到'\0'结束输出。输出结果为“Figflowerisred.”。答案选择 A 选项。

143．有以下程序：

```
#include <stdio.h>
main()
{
   char b[3][10],c;
   int i;
   for(i=0;i<2;i++) scanf("%s",b[i]);
   i=0;
   while((c=getchar())!='\n') b[2][i++]=c;
   b[2][i] = '\0';
   printf("%s%s%s\n", b[0], b[1], b[2]);
}
```

执行时若输入以下字符串：

Peach flower is pink.<回车>

则输出结果是（　　）。

A. Peachflower is pink.

B. Peachfloweris pink.

C. Peachflowerispink.

D. Peach flower is pink.

【答案】A

【解析】由于用“%s”格式输出字符串时，遇到空格、制表符和回车时读取结束，因此，b[0] = "Peach"，b[1] = "flower"，而 while 循环中，用 getchar 函数一直读入字符直到'\n'为止，b[2]=" is pink."，答案选择 A 选项。

144．有以下程序：

```
#include <stdio.h>
main()
{
   char b[4][10],c;
   int i,j;
   for(i=0;i<4;i++)
   {
      j=0;
      while((c=getchar())!=' '&& c!='\n') b[i][j++]=c;
      b[i][j] = '\0';
   }
   printf("%s%s%s%s\n", b[0], b[1], b[2], b[3]);
}
```

程序运行时从第一列开始输入：Peach flower is pink.<回车>

则输出结果是（　　）。

A. Peachflowefispink.

B. Peachflowefis pink.

C. Peachflower is pink.

D. Peach flower is pink.

【答案】A

【解析】for 循环语句中，通过执行 while 语句，while 的判断条件是输入的字符不为空格和回车，当遇到空格或回车符时，i 加一，分别保存在二维数组 b 的 4 行中，最终 b[0] = "Peach"，b[1] = "flower"，b[2] = "is"，b[3] = "pink."，再通过%s 格式控制符，将二维数组 b 的四行字符串无空格连接输出。答案选择 A 选项。

145．有以下程序：

```
#include<stdio.h>
#include<string.h>
main()
{
   char w[20], a[5][10] = {"abcdef", "ghijkl", "mnopq", "rstuv", "wxyz"};
   int i,j;
   for(i=0;i<5;i++)
   {
      j=0;
      while(a[i][j]!='\0')j++;
      w[i]=a[i][j-2];
   }
   w[5]='\0';
   puts(w);
}
```

程序运行后的输出结果是（　　）。

A. agmrw

B. ekpuy

C. djotx

D. flqvz

【答案】B

【解析】本题程序功能是依次将 a[0]、a[1]、a[2]、a[3]对应的字符串中的倒数第二个字符赋给 w[0]，w[1]，w[2]，w[3]，答案选择 B 选项。

146. 有以下程序：

```
#include <stdio.h>
main()
{
   char *a[]={"abcd","ef","gh","ijk"};
   int i;
   for(i=0;i<4;i++)printf("%c",*a[i]);
}
```

程序运行后的输出结果是（　　）。

A. aegi

B. dfhk

C. abcd

D. abcdefghijk

【答案】A

【解析】char *a[]定义了一个指向数组的指针。由题中初始化结果得：*a[0]= "abcd"，a[1]= "ef"，a[2]= "gh"，a[3]= "ijk"，但是在输出语句时要求输出的格式是%c，为一个字符，所以每个元素输出第一个字符，即 aegi。答案选择 A 选项。

147. 有以下程序：

```
#include<stdio.h>
#include<string.h>
main( )
{
   char *mm[4]= {"abcd", "1234", "mnop", "5678"};
   char **pm= mm;
   int i;
   for(i=0;i<4;i++) printf("%s",pm[i]+i);
   printf("\n");
}
```

程序的运行结果是（　　）。

A. abcd1234mnop5678

B. abcd234op8

C. a2o8

D. a1m5

【答案】B

【解析】程序执行过程为：定义指针数组，长度为 4，并初始化为 4 个字符串。定义指向指针的指针变量，初始化为指针数组首地址。for 循环依次输出 4 个字符串，第 i 个字符串从第 i 个下标值处开始输出直到字符串结束，即第一个字符串输出 abcd，第二个字符串输出 234，第三个字符串输出 op，第四个字符串输出 8。程序的运行结果是 abcd2340p8，答案选择 B 选项。

148. 有以下程序：

```
#include <stdio.h>
main()
{
   char *s[6] = {"ABCD", "EFGH", "IJKL", "MNOP", "QRST", "UVWX"}, **p;
   int i;
   p=s;
   for(i=0;i<4;i++)printf("%s",p[i]);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A. ABCDEFGHIJKLMNOP

B. ABCDEFGHHKL

C. ABCD

D. AEIM

【答案】A

【解析】程序定义数组指针 s 以及指针的指针 p，p 指向数组指针 s。p[i]为 s 的第 i 行字符串，因此程序输出 s 的前四行字符串。答案选择 A 选项。

149. 有以下程序

```
#include <stdio.h>
main()
{
   char c[2][5] = {"6938","8254"}, *p[2];
   int i,j,s=0;
   for(i=0;i<2;i++)
      p[i]=c[i];
   for(i=0;i<2;i++)
      for(j=0;p[i][j]>0;j+=2)
         s=10*s+p[i][j]-'0';
   printf("%d\n",s);
}
```

程序运行后的输出结果是（　　）。

A. 9284

B. 9824

C. 6982

D. 6385

【答案】D

【解析】p 定义了两个字符串指针，p[0]指向 c[0]，p[1]指向 c[1]。所以，p[i][j]等价于 c[i][j]，s = 10*s+ p[i][j]-'0'，作用是把字符串 p[i]转化成 10 进制数，注意到内层循环中 j+=2，结果取 c[0]中的第 0 个，第 2 个元素，取 c[1]中的第 0 个，第 2 个元素。i=0 时，输出 63；i=1 时，输出 85。答案选择 D 选项。

150．有以下程序

```
#include <stdio.h>
void fun(char **p)
{
   ++p;
   printf("%s\n",*p);
}
main()
{
   char *a[] = {"Morning", "Afternoon", "Evening",  "Night"};
   fun(a);
}
```

程序的运行结果是（　　）。

A. Afternoon

B. flemoon

C. Morning

D. orning

【答案】A

【解析】用字符串数组 a 给形参 p 赋初始值，p 指向 a[0]，在函数 fun 中执行++p，则 p 指向 a[1]，程序运行结果是 Afternoon。答案选择 A 选项。

151．以下关于字符串的叙述中正确的是（　　）。

A. C 语言中有字符串类型的常量和变量

B. 两个字符串中的字符个数相同时才能进行字符串大小的比较

C. 可以用关系运算符对字符串的大小进行比较

D. 空串比空格打头的字符串小

【答案】D

【解析】A 项错误，C 语言中没有字符串类型，而是通过字符数组的形式保存字符串。B 项错误，字符串比较的方法是：依次对 s1 和 s2 中对应位置上的字符两两进行比较，当出现第一对不相同的字符时，即由这两个字符决定所在串的大小（比较字符大小的依据是其 ASCII 码值）。空格是一种字符，所以空串肯定比空格打头的字符串小。C 项错误，比较字符串大小时，通过库函数 strcmp(s1,s2)或者自定义函数，不能使用关系运算符。D 项正确：空串的长度为 0，而以空格打头的字符串的长度至少为 1。答案选择 D 选项。

152．若有定义语句：

char s[10]="1234567\0\0";

则 strlen(s)的值是（　　）。

A．7

B．8

C．9

D．10

【答案】A

【解析】C 语言规定以字符'\0'作为字符串结束的标识符。strlen 函数返回的是字符串的长度，不包含字符'\0'，所以值是 7。答案选择 A 选项。

153．下列选项中，能够满足“若字符串 s1 等于字符串 s2，则执行 ST”要求的是（　　）。

A．if(strcmp(s2,s1)==0) ST;

B．if(s1==s2) ST;

C．if(strcpy(s1,s2)==1) ST;

D．if((s1-s2)==0) ST;

【答案】A

【解析】A 项，函数 strcmp(s2,s1)的作用是比较字符串大小；BD 两项，都是比较的字符串 s1 与 s2 的地址是否一致而不是比较字符串内容是否一致；C 项，函数 strcpy(s1,s2)的作用是进行字符串复制。答案选择 A 选项。

154．字符数组 a 和 b 中存储了两个字符串，判断字符串 a 和 b 是否相等，应当使用的是（　　）。

A．if(strcmp(a,b)==0)

B．if(strcpy(a,b))

C．if(a==b)

D．if(a=b)

【答案】A

【解析】C 语言中，判断字符串是否相等，使用字符串比较函数 strcmp()，不能使用相等操作符“==”。strcmp(s1,s2)函数比较 s1 和 s2 所指字符串的大小时，若串 s1＞串 s2，函数值大于 0（正数）；若串 s1=串 s2，函数值等于 0；若串 s1＜串 s2，函数值小于 0（负数）。答案选择 A 选项。

155．若有定义语句

char*s1="OK",*s2="ok";

以下选项中能够输出"OK"的语句是（　　）。

A．if(strcmp(s1,s2)==0)puts(s1);

B．if(strcmp(s1,s2)!=0)puts(s2);

C．if(strcmp(s1,s2)==1)puts(s1);

D．if(strcmp(s1,s2)!=0)puts(s1);

【答案】D

【解析】strcmp 用于比较两字符串：当 s1<s2 时，返回值小于 0；当 s1=s2 时，返回值为 0；当 s1>s2 时，返回值大于 0。题中 s1 与 s2 两个字符串不相等且 s1 小于 s2，AC 两项错误。B 项，输出的小写的 ok，只有 D 项输出的是大写的 OK。答案选择 D 选项。

156．若有定义语句：

char str1[] = "string", str2[8], *str3, str4[10] = "string";

库函数 strcpy 的功能是复制字符串，以下选项中错误的函数调用是（　　）。

A. strcpy(str3, "HELLO!");

B. strcpy(str2, "HELLO!");

C. strcpy(str1, "HELLO!");

D. strcpy(str4, "HELLO!");

【答案】A

【解析】题目中字符指针 str3，没有为指针开辟内存，不能作为函数参数。答案选择 A 选项。

157. 以下不能将 s 所指字符串正确复制到 t 所指存储空间的是（　　）。

A. while(*t=*s){t++;s++;}

B. for(i=0;t[j]=s[i];i++);

C. do{*t++=*s++;}while(*s);

D. for(i=0,j=0;t[i++]=s[j++];);

【答案】C

【解析】C 项，*t++=*s++;能够实现将 s 中除了字符串结束标志的字符'\0'以外的所有字符复制到 t 中，字符串 t 是不完整的。答案选择 C 选项。

158. 若有以下定义和语句：

char s1[10]="abcd!", *s2="n123\\";

printf("%d%d\n", strlen(s1), strlen(s2));

则输出结果是（　　）。

A. 55

B. 105

C. 107

D. 58

【答案】A

【解析】strlen 函数是计算字符串长度的函数，求字符串的实际字符个数，不包括字符'\0'在内，在字符'\0'之后的所有字符均不计入长度中。所以 strlen(s1)=5。而以"\"开头的字符序列是转义字符，"\\"的含义是一个字符'\'，所以 s2 所指向的内容实际上是"n123\"，所以 strlen(s2)=5。答案选择 A 选项。

159. 有以下程序

```
#include <stdio.h>
#include <string.h>
main()
{
   printf("%d\n",strlen("ATS\n012\1"));
}
```

程序运行后的输出结果是（　　）。

A. 3

B. 8

C. 4

D. 9

【答案】B

【解析】strlen 函数是计算字符串 s 的长度，不包括结束符'\0'在内，其中由“\”代表的转义字符也算作一个字符。本题中，转义字符'\n'和'\1'分别代表一个字符。答案选择 B 选项。

160. 以下语句的输出结果是（　　）。

printf("%d\n",strlen("\t\"\065\xff\n"));

A．5
B．8
C．14
D．输出项不合法，无正常输出
【答案】A
【解析】字符常量是用一对单引号括起来的单个字符，还有一些特殊字符常量，即以'\'开头的转义字符。'\'后可以为某些单个字符也可以为八进制或十六进制数字。't'表示制表符，'\"'表示字符串中包含字符"，'\065'中 65 是八进制数，对应十进制的 53，ASCII 码为 53 的是字符'5'；'\xff'中 ff 是十六进制数，对应十进制的-1；'\n'表示换行符，共 5 个字符，答案选择 A 选项。

161．有如下程序：

```
#include <stdio.h>
#include <string.h>
main()
{
   printf("%d\n", strlen("0\t\nA011\1"));
}
```

程序运行后的输出结果是（　　）。
A．8
B．9
C．7
D．10
【答案】A
【解析】strlen 是求字符串长度的函数，"0\t\nA011\1"是一个长度为 8 的字符串，其中\t、\n、\1 均为转义字符常量，但是长度也是 1，因此输出结果是 8，答案选择 A 选项。

162．有以下程序：

```
#include <stdio.h>
#include <string.h>
main()
{
   char a[10]="abcd";
   printf("%d,%d\n", strlen(a), sizeof(a));
}
```

程序运行后的输出结果是（　　）。
A．7,4
B．4,10
C．8,8
D．10,10
【答案】B
【解析】在 C 语言中，strlen()用来统计字符串中字符的个数（不包含字符串结束标志'\0'），sizeof()用来求分配给数组的存储空间大小。题目中字符串 a 中字符个数为 4，但由于数组 a 定义含有 10 个字符，所以所占空间大小为 10。所以答案选择 B 选项。

163．有以下程序：

```
#include<stdio.h>
#include<string.h>
main()
{
   char str[]={"Hello,Beijing"};
   printf("%d,%d\n",strlen(str),sizeof(str));
}
```

程序的运行结果是（　　）。

A. 13,13

B. 13,14

C. 13,15

D. 14,15

【答案】B

【解析】strlen 返回字符串的长度，不包含字符串末尾的结束字符'\0'，结果为 13；sizeof 返回字符串所占存储空间的大小，由于字符串最后要加上一个'\0'，所以结果为 13+1=14。答案选择 B 选项。

164. 有以下程序

```
#include<stdio.h>
#include<string.h>
main()
{
   char x[]="STRING";
   x[0]=0;
   x[1]='\0';
   x[2]='0';
   printf("%d %d\n",sizeof(x),strlen(x));
}
```

程序运行后的输出结果是（　　）。

A. 6 1

B. 7 0

C. 6 3

D. 7 1

【答案】B

【解析】sizeof 是返回字符串在内存中所占用的空间，是字符数组真正的长度。strlen 是返回字符串的长度，strlen 遇到'\0'（ASCII 码值为 0）就结束，而且不包括'\0'。题目中数组 x 初始化为字符串"STRING"，6 个字符加上字符串结束符共占用 7 个字节长度，所以 sizeof(x)返回值为 7。数组 x 在初始化后，又重新赋值 x[0] = 0，即 x[0] = '\0'，所以 strlen(x)返回值为 0。答案选择 B 选项。

165. 有如下程序：

```
#include<stdio.h>
#include<string.h>
main()
{
   char a[]="THIS", *b="OK";
   printf("%d,%d,%d,%d\n", strlen(a), sizeof(a), strlen(b), sizeof(b));
}
```

程序运行后的输出结果是（　　）。

A．4,5,2,4
B．4,4,2,1
C．5,5,3,3
D．4,5,2,3
【答案】A
【解析】strlen 函数统计字符串长度，遇到'\0'统计结束。sizeof 用来获取类型或数据对象的长度，也即是一个这种数据类型的变量在内存中所占字节数。a 数组'\0'之前有效字符有 4 个，由于字符串有效字符之后有一个'\0'也会被放入数组，char 类型占一个字节，所以数组 a 所占字节数为 1*5=5。b 为指向字符串的指针，字符串长度为 2，指针类型变量所占字节数为 4。答案选择 A 选项。

166．有如下程序：

```
#include <stdio.h>
#include <string.h>
main()
{
   char name[10]="c-book";
   char *str=name;
   printf("%d,%d,%d,%d\n", sizeof(name), strlen(name), sizeof(str), strlen(str));
}
```

程序运行后的输出结果是（　　）。
A．10,6,4,6
B．11,6,11,6
C．11,6,1,6
D．10,7,1,7
【答案】A
【解析】由一维数组初始化的知识可知，name[10] = "c-book"，即 name[10]="c-book\0\0\0\0"。sizeof 函数用来获取类型或数据对象的长度，也即是一个这种数据类型的变量在内存中所占字节数。strlen 函数统计字符串长度，遇到'\0'统计结束。字符指针变量和字符数组在内存中的存储情形不同，数组指的是按照数组长度分配的若干存储空间，在内存中字符类型变量占 1 个字节；而指针只是一个存储字符串首地址的存储空间，指针占 4 个字节。所以 sizeof(name)=1×10=10，strlen(name)=6，sizeof(str)=4，strlen(str)=6。答案选择 A 选项。

167．有以下程序：

```
#include <stdio.h>
#include <string.h>
main()
{
   char name[9]="c##line";
   char *str=name;
   printf("%d,%d,%d,%d\n", sizeof(name), strlen(name), sizeof(str), strlen(str));
}
```

程序运行后的输出结果是（　　）。
A．9,7,4,7
B．8,6,9,6
C．8,6,3,6
D．10,8,5,8
【答案】A
【解析】由于 name 是一个长度为 9 的一维数组，故在内存中占用 9 字节长度，而其中字符串"c##line"只有 7 个字符，strlen 函数返回的是该字符串的长度，不包含结束符，str 是一个指针变量，占用 4 字节长度，但是由

于 name 首地址赋给了 str 指针变量，在调用 strlen 函数时，返回的是指针对应地址单元的字符串的长度 7，答案选择 A 选项。

168．有如下程序：

```
#include <stdio.h>
main()
{
   char *p1 = 0;
   int *p2 = 0;
   float *p3 = 0;
   printf("%d,%d,%d\n", sizeof(p1), sizeof(p2), sizeof(p3));
}
```

程序运行后的输出结果是（　　）。

A．4,4,4

B．1,4,8

C．0,0,0

D．1,2,4

【答案】A

【解析】sizeof 函数用来获取类型或数据对象的长度，也即是一个这种数据类型的变量在内存中所占字节数。由于一个变量的地址也是一个值，因此就可以把这个地址值存放到另一个变量里保存。这种专门用来存放变量地址的变量，称为“指针变量”。所有类型的指针变量都是地址，所占字节数均为 4，答案选择 A 选项。

169．有以下程序（strcpy 为字符串复制函数，strcat 为字符串连接函数）：

```
#include <stdio.h>
#include <string.h>
main()
{
   char a[10] ="abc",b[10]="012",c[10]="xyz";
   strcpy(a+1,b+2);
   puts(strcat(a,c+1));
}
```

程序运行后的输出结果是（　　）。

A．a12xyz

B．12yz

C．a2yz

D．bc2yz

【答案】C

【解析】先执行 strcpy，将 b 数组中第 2 个及之后位置上的字符复制到 a 数组中第 1 个及之后的位置上的字符位置，即此时数组 a 中字符为 a2，再执行 strcat 连接函数，将 a 中字符与 c 数组中第 2 个及之后位置上的字符连接，即为 a2yz。注意，数组下标从 0 开始。答案选择 C 选项。

170．有以下程序：

```
#include <stdio.h>
#include <string.h>
main()
{
   char a[20]="ab",b[20]="cdef";
   int k=0;
   strcat(a,b);
   while(a[k]!='\0')
   {
      b[k]=a[k];
      k++;
   }
   puts(b);
}
```

程序的运行结果是（　　）。

A. abcdef

B. cbcdef

C. cdef

D. ab

【答案】A

【解析】strcat 把字符串 a，b 连接起来放在数组 a 中，while 语句再将字符串 a 赋给字符串 b。答案选择 A 选项。

171. 有以下程序（其中的 strstr()函数头部格式为：char* strstr(char* p1,char* p2)确定 p2 字符串是否在 p1 中出现，并返回 p2 第一次出现的字符串首地址）：

```
#include <stdio.h>
#include <string.h>
char *a="you";
char *b="Welcome you to Beijing!";
main()
{
   char *p;
   p=strstr(b,a)+strlen(a)+1;
   printf("%s\n",p);
}
```

程序的运行结果是（　　）。

A. to Beijing!

B. you to Beijing!

C. Welcome you to Beijing!

D. Beijing!

【答案】A

【解析】调用 strstr 函数，返回值为 a 指向的字符串在 b 指向的字符串中第一次出现的位置，并将此地址赋给指针 p。strlen()函数求字符串的实际长度（不包含结束标志）。strstr 函数返回的地址下标值为 8，加上 a 长度 3，再加 1，指针 P 指向的地址下标值为 12，输出：to Beijing!，答案选择 A 选项。

172. 有以下程序：

```
#include <stdio.h>
#include <string.h>
char *a="you";
char *b="Welcome you to Beijing!";
main()
{
   char *p;
   p=b;
   while(*p != *a)p++;
   p+=strlen(a)+1;
   printf("%s\n",p);
}
```

程序运行后的输出结果是（　　）。

A. Beijing!

B. you to Beijing!

C. Welcome you to Beijing!

D. to Beijing!

【答案】D

【解析】while 函数判断 p 指针指向地址的内容是否和 a 指针指向地址的内容相等，如果不相等，则往后移动指针 p，当 p 指向*b 字符串中的'y'时，和*a 首地址元素相等，退出 while 语句后，p 再向右移动 a 字符串长度+1 个单位地址，此时 p 指向't'，输出't'及剩余的字符串，答案选择 D 选项。

173．有以下程序：

```
#include <stdio.h>
#include <string.h>
main()
{
   char s[]="Beijing";
   printf("%d\n", strlen(strcpy(s,"China")));
}
```

程序运行后的输出结果是（　　）。

A. 5

B. 7

C. 12

D. 14

【答案】A

【解析】在存储字符串常量时，由系统在字符串的末尾自动加一个'\0'作为字符串的结束标志。strcpy 函数将"China"复制给字符数组 s，内存存储情况为"China\0"。strlen 函数统计字符串长度时，遇到'\0'结束，s 数组长度为 5。答案选择 A 选项。

174．有以下程序：

```
#include<stdio.h>
#include<string.h>
main( )
{
   char w[20], a[5][10] = {"abcdef", "ghijkl", "mnopq", "rstuv", "wxyz."};
   int i;
   for(i=0;i<5;i++) w[i]=a[i][strlen(a[i])-1];
   w[5]='\0';
   puts(w);
}
```

A. flqv.
B. agmrw
C. ekpuy
D. flqvz

【答案】A

【解析】puts 函数将数组中存放的字符串输出，用'\n'取代字符串结束符'\0'。strlen 函数求字符串的实际长度（不包含结束标志）。程序中 for 循环将第 i 个字符串最后一个字符放入 w 数组下标值为 i 的位置。最后将数组 w 的最后一个字符赋值为'\0'，数组 w 变成"flqv."。答案选择 A 选项。

175．有以下函数：

```
int fun(char *x,char *y)
{
   int n=0;
   while((*x==*y)&&*x!='\0')
   {
      x++;
      y++:
      n++;
   }
}
```

函数的功能是（　　）。

A. 查找 x 和 y 所指字符串中是否有'\0'
B. x、y 所指字符串最前面连续相同的字符个数
C. 将 y 所指字符串赋值给 x 所指存储空间
D. 统计 x 和 y 所指字符串中相同的字符个数

【答案】B

【解析】由程序可知，定义语句中的 x 和 y 是用户标识符，在每个变量前的*是一个说明符，只有在 x 和 y 所指字符串中的首字符相同且非结束符时，字符串变为相应的字串，n 加一继续循环；一旦所指字符串的首字符不相同，则终止循环而返回相同个数 n。答案选择 B 选项。

176．有以下函数：

```
int fun(char *ps)
{
   char *p;
   p=ps;
   if(*ps==NULL)return 0;
   while(*++p);
   return(p-ps);
}
```

该函数的功能是（　　）。

A. 计算字符串的长度

B. 实现字符串的赋值

C. 将字符串逆序存放

D. 计算字符串所占字节数

【答案】A

【解析】在 fun 函数中定义了字符指针 P，首先把形参 ps（相当于字符串的首地址）赋值给指针 P，再通过 while 循环移动 p 指针，当 p 指针指向的存储单元的内容为 0 时，退出 while 循环，返回此时 p 地址和 ps 地址差值，即字符串的长度。答案选择 A 选项。

177．有以下函数

```
int aaa(char *s)
{
   char *t=s;
   while(*t++);
   t--;
   return (t-s);
}
```

以下关于 aaa 函数功能叙述正确的是（　　）。

A. 求字符串 s 的长度

B. 比较两个串的大小

C. 将串 s 复制到串 t

D. 求字符串 s 所占字节数

【答案】A

【解析】aaa 函数中，首先定义了一个字符指针 t 指向形参 s，然后通过一个 while 循环让指针 t 不断递增，直到 t 指向字符串结束标志处。当 t 指向结束标志处时，由于后缀++运算符的原因，它还会被再递增 1，此时 t 指向字符串结束符'\0'后面一个单元，所以接下来的 t--；语句让它回到结束标志处。最后返回 t-s，因为 s 还是指向字符串第 1 个字符处，而 t 指向字符串结束符，故返回值为字符串的长度值。答案选择 A 选项。

178．有以下函数

```
int fun(char *s)
{
   char *t=s;
   while(*t++);
   return(t-s);
}
```

该函数的功能是（　　）。

A. 计算 s 所指字符串占用内存字节的个数

B. 比较两个字符串的大小

C．计算 s 所指字符串的长度

D．将 s 所指字符串复制到字符串 t 中

【答案】A

【解析】本题中，首先让 t 指向形参 s，然后通过一个循环体为空的 while 循环，将 t 逐次后移，直到其所指内容为'\0'（字符串结束标志）。此时 t 仍然会被增 1，所以从循环出来，t 指向的是 s 所指字符串的结束标志的后一个字节。因此，返回的 t-s 是 s 所指字符串占用内存字节的个数，A 项正确。而 C 项所说的长度并不包括字符串结束标志位，错误。答案选择 A 选项。

179．有以下函数：

```
void fun(char*p,char*q)
{
   while((*p++=*q++)!='\0');
}
```

该函数的功能是（　　）。

A．计算字符串的长度

B．计算字符串所占字节数

C．将字符串逆序存放

D．实现字符串的复制

【答案】D

【解析】while 循环语句中，实现功能是把 q 指针指向地址单元的值赋给 p 指针指向的地址中，且每次完成赋值后，移动指针，进行下一次赋值，直到 q 指针指向字符串结束符时，退出 while 循环，功能为把 q 指向的字符串复制到 p 指向的地址中。答案选择 D 选项。

180．下列函数的功能是（　　）。

```
fun(char * a,char * b)
{
   while((*b= *a)!='\0')
   {
      a++;
      b++;
   }
}
```

A．将 a 所指字符串赋给 b 所指空间

B．使指针 b 指向 a 所指字符串

C．将 a 所指字符串和 b 所指字符串进行比较

D．检查 a 和 b 所指字符串中是否有'\0'

【答案】A

【解析】函数 fun 中 a 和 b 是两个字符型指针，在 while 语句的表达式中将指针 a 所指向的字符赋给指针 b 所指向的内存单元，再判断指针 b 所指向的字符是不是字符串中的结尾符，若不是，则字符指针 a 和 b 分别自增，再执行循环语句，直至 b 所指向的字符为字符串中的空字符。所以答案选择 A 选项。

181．有以下函数：

```
int fun(char *s,char *t)
{
   while((*s)&&(*t)&&(*t++==*s++));
   return (*s-*t);
}
```

函数的功能是（　　）。

A．求字符串的长度

B．比较两个字符串的大小

C．将字符串 s 复制到字符串 t 中

D．连接字符串 s 和字符串 t

【答案】B

【解析】函数体执行过程为：将两个字符串首地址传入函数，分别赋给指针 s 与 t，在函数体内 s 与 t 所指向的字符串的字符不为'\0'时，判断两个指针指向的字符是否相同，若相同则两个指针分别加一指向下一个字符，若不同则退出 while 循环，返回不相同的字符的 ASCII 码值之差。返回值大于 0 表示字符串 s>t；返回值小于 0 表示 s<t；返回值为 0，表示 s=t。函数实现了比较两个字符串大小的功能。答案选择 B 选项。

182．有以下程序：

```
#include <stdio.h>
#include <string.h>
main()
{
   int i;
   char a[]="How are you!";
   for(i=0;a[i];i++)
   {
      if(a[i]==' ')
      {
         strcpy(a,&a[i+1]);
         i=0;
      }
   }
   printf("%s\n",a);
}
```

程序的运行结果是（　　）。

A．you!

B．Howareyou!

C．areyou!

D．are you!

【答案】A

【解析】复制字符串函数 strcpy(str1,str2)，将 str2 完整的（包括'\0'）复制到 str1 中，str1 中原有的内容被覆盖。程序执行过程为：在 a[i]不等于'\0'的情况下，判断当前元素是否为空格，若为空格将下一个元素以及之后的所有字符复制到 a 数组中。当 i=3 时，if 条件成立，a 以及之后单元中元素为"are you!\0"，i=0，进行下一次循环，之后当 i=3 时，if 条件成立，a 以及之后单元中元素为"you!\0"，i=0，之后的元素中在'\0'前没有空格出现，输出 you!。答案选择 A 选项。

183．有以下程序：

```
#include <stdio.h>
#include <string.h>
main()
{
   char a[5][10]= {"china", "beijing",  "you", "tiananmen",  "welcome"};
   int i,j;
   char t[10];
   for(i=0;i<4;i++)
      for(j=i+1;j<5;j++)
         if(strcmp(a[i],a[j])>0)
         {
            strcpy(t,a[i]);
            strcpy(a[i],a[j]);
            strcpy(a[j],t);
         }
   puts(a[3]);
}
```

程序运行后的输出结果是（　　）。

A．beijing

B．china

C．welcome

D．tiananmen

【答案】C

【解析】strcmp 用于比较两字符串：当 s1<s2 时，返回值<0；当 s1=s2 时，返回值=0；当 s1>s2 时，返回值>0。本程序就是排序算法中的冒泡排序，将 5 个字符串按字典序从小到大排列。最后数组 a 的内容依次为"beijing"、"china"、"tiananmen"、"welcome"、"you"。答案选择 C 选项。

184．有以下程序（strcat 函数用以连接两个字符串）：

```
#include <stdio.h>
#include <string.h>
main()
{
   char a[20]="ABCD\0EFG\0",  b[]="IJK";
   strcat(a,b);
   printf("%s\n",a);
}
```

程序运行后的输出结果是（　　）。

A．ABCDE\0FG\0IJK

B．ABCDIJK

C．IJK

D．EFGIJK

【答案】B

【解析】在 C 语言中，系统在每个字符串的最后自动加入一个字符'\0'作为字符串的结束标志。char a[20] = "ABCD\0EFG\0"中，当遇到'\0'就结束初始化，因此 a[] = "ABCD"，b[] = "IJK"，连接这两个字符串 strcat(a,b)得到 ABCDIJK。答案选择 B 选项。

185．有以下程序（程序中库函数 islower(ch)用以判断 ch 中的字符是否为小写字母）：

```
#include <stdio.h>
#include <ctype.h>
void fun(char *p)
{
   int i=0;
   while(p[i])
   {
      if(p[i]==' '&&islower(p[i-1]))p[i-1]=p[i-1]-'a'+'A';
      i++;
   }
}
main()
{
   char s1[100]="ab cd EFG!";
   fun(s1);
   printf("%s\n",s1);
}
```

程序运行后的输出结果是（　　）。

A. ab cd EFG!

B. Ab Cd EFg!

C. aB cD EFG!

D. ab cd EFg!

【答案】C

【解析】fun 函数实现的功能是将字符串中空格前面的小写字母转换为对应的大写字母。答案选择 C 选项。

186. 有以下程序：

```
#include <stdio.h>
#include<string.h>
main()
{
   char str[][20]={"One*World","One*Dream!"},*p=str[1];
   printf("%d,",strlen(p));
   printf("%s\n",p);
}
```

程序运行后的输出结果是（　　）。

A. 9,One*World

B. 9,One*Dream!

C. 10,One*Dream!

D. 10,One*World

【答案】C

【解析】程序将两个字符串常量赋值给一个二维字符数组，然后 p 指向第二个字符串。strlen 统计字符串中有效字符的个数，可知"One*Dream!"中共有 10 个字符。所以答案选择 C 选项。

187. 有以下程序

```
#include <stdio.h>
#include <string.h>
main()
{
   char p[20]= {'a','b','c','d'}, q[]="abc", r[]="abcde";
   strcat(p,r);
   strcpy(p+strlen(q),q);
   printf("%d\n",strlen(p));
}
```

程序运行后的输出结果是（　　）。

A. 6

B. 9

C. 11

D. 7

【答案】A

【解析】首先定义了 3 个字符数组 p、q、r，并分别被初始化。数组 p 指定的大小为 20，初始化列表为{'a','b','c','d'}，即只指定了前 4 个元素的内容，根据 C 语言的规定，初始化列表不足时，其余元素均自动初始化为 0。然后通过 strcat 函数，将字符串 r 连接到字符串 p 之后，即执行后 p 中的内容为"abcdabcde"。"strlen(q)"表示求字符串 q 的长度，不包括最后的字符串结束符，得到 3，所以语句"strcpy(p+strlen(q),q); "的作用就是：将字符串 q 复制到数组 p 的第 4 个元素位置处，数组 p 变成{'a','b','c', 'a','b','c'}，所以字符串 p 的长度是 6。答案选择 A 选项。

二、填空题

请根据以下各小题的要求设计 C 应用程序（包括界面和代码）。

函数 fun 的功能是：把形参 a 所指数组中的最大值放在 a[0]中，接着求出 a 所指数组中的最小值放在 a[1]中；再把 a 所指数组元素中的次大值放在 a[2]中，把 a 数组元素中的次小值放在 a[3]中；其余依此类推。例如：若 a 所指数组中的数据最初排列为：1，4，2，3，9，6，5，8，7，则按规则移动后，数据排列为：9，1，8，2，7，3，6，4，5。形参 n 中存放 a 所指数组中数据的个数。

注意：部分源程序给出如下。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的横线上填入所编写的若干表达式或语句。

试题程序如下：

```
#include <stdio.h>
#define N 9
/************found************/
void fun(int  ①_____,int n)
{
   int i,j,max,min,px,pn,t;
   /************found************/
   for(i=0;i<N-1;i+=②_____)
   {
      max=min=a[i];
      px=pn=i;
      /************found************/
      for(j=③_____;j<N;j++)
      {
         if(max<a[j])
         {
            max=a[j];
            px=j;
         }
         if(min>a[j])
         {
            min=a[j];
            pn=j;
         }
      }
      if(px!=i)
      {
         t=a[i];
         a[i]=max;
         a[px]=t;
         if(pn==i)
            pn=px;
      }
      if(pn!=i+1)
      {
         t=a[i+1];
         a[i+1]=min;
         a[pn]=t;
      }
   }
}
```

【答案】

①*a 或 a[]

②2

③i+1

【解析】

填空 1：由后边的程序可知，形参 a 应定义为整型数组，所以此处应填：*a 或 a[]。

填空 2：由于循环每次对两个数组元素进行赋值，所以外 for 循环每次增量应该加 2。

填空 3：由题意可知，此处是找出从 i+1 到 n 的数的最大值和最小值，所以内 for 循环的初始值应为：i+1。

三、改错题

给定程序 MODI1.C 中函数 fun 的功能是：先将 s 所指字符串中的字符按逆序存放到 t 所指字符串中，然后把 s 所指串中的字符按正序连接到 t 所指串的后面。

例如：当 s 所指的字符串为：“ABCDE”时，则 t 所指的字符串应为：“EDCBAABCDE”。

请改正程序中的错误，使它能得出正确的结果。

注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构！

试题程序如下：

```
#include <stdio.h>
#include <string.h>
void fun(char *s, char *t)
{
   /*********found*********/
   int i;
   i=0;
   s1=strlen(s);
   for(;i<s1;i++)
      /*********found*********/
      t[i]=s[s1-i];
   for(i=0;i<s1;i++)
      t[s1+i]=s[i];
   t[2*s1]='\0';
}
main()
{
   char s[100],t[100];
   printf("\nPlease enter string s:\n");
   scanf("%s",s);
   fun(s,t);
   printf("The result is:%s\n",t);
}
```

【答案】

（1）错误：int i;

正确：int i,s1;

（2）错误：t[i]=s[s1-i];

正确：t[i]=s[s1-i-1];

【解析】

错误 1：变量 s1 没有定义。

错误 2：第一个循环实现将 s 串中的字符逆序存入 t 串中，si 为 s 的长度，数组 s 的下标值范围是 0～s-1，t[i]对应 s 串中的 s[s1-i-1]。

四、设计题

请根据以下各小题的要求设计 C 应用程序（包括界面和代码）。

请编写函数 fun()，它的功能是：将 3 行 4 列矩阵 x 乘以 4 行 3 列矩阵 y，结果放在 3 行 3 列矩阵 xy 中。矩阵相乘的基本方法是：矩阵 xy 中行列下标分别为 i，j 的元素的值，是矩阵 x 中第 i 行上 4 个元素与矩阵 y 第 j 列上 4 个元素对应相乘再相加的和。

注意：部分源程序给出如下。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入所编写的若干语句。

试题程序如下：

```
#include <stdio.h>
#include <string.h>
void fun(int a[3][4],int b[4][3],int ab[3][3])
{

}
main()
{
   int x[3][4] = {{1,0,1,1}, {2,1,0,1}, {1,2,0,3}};
   int y[4][3] = {{1,1,1}, {0,0,0}, {2,1,1}, {1,1,3}};
   int xy[3][3] = {0},i,j;
   fun(x,y,xy);
   printf("a × b = ab:(3,3):\n");
   for(i=0;i<3;i++)
   {
      for(j=0;j<3;j++)
         printf("%d",xy[i][j]);
      printf("\n");
   }
}
```

答：

```
void fun(int a[3][4],int b[4][3],int ab[3][3])
{
   int j,k,l;
   for(k=0;k<3;k++)
      for(l=0;l<3;l++)
         for(j=0;j<4;j++)
            ab[k][l]+=a[k][j]*b[j][l];
}
```

【解析】本题首先要明确矩阵 xy 中行列下标分别为 i，j 的元素的值，是矩阵 x 中第 i 行上 4 个元素与矩阵 y 第 j 列上 4 个元素对应相乘再相加的和。因此，每个元素的求解是循环控制来实现的，即 ab[k][l]+=a[k][j]*b[j][l]，矩阵 ab 的每个元素的表示可用一个二重循环，整个函数是一个三重循环的嵌套。

# 第 7 章　用函数实现模块化程序设计

一、选择题

1．在源程序的开始处加上“#include<stdio.h>”进行文件引用的原因，以下叙述正确的是（　　）。

A．stdio.h 文件中包含标准输入输出函数的函数说明，通过引用此文件以便能正确使用 printf、scanf 等函数

B．将 stdio.h 中标准输入输出函数链接到编译生成的可执行文件中，以便能正确运行

C．将 stdio.h 中标准输入输出函数的源程序插入到引用处，以便进行编译链接

D．将 stdio.h 中标准输入输出函数的二进制代码插入到引用处，以便进行编译链接

【答案】A

【解析】“stdio.h”文件中包含标准输入输出函数的函数说明，预处理指令#include<stdio.h>是指程序可以在该文件中找到 printf，scanf 等函数，答案选择 A 选项。

2．有以下程序：

```
#include<stdio.h>
main()
{
   printf("%d\n",NULL);
}
```

程序运行后的输出结果是（　　）。

A．0

B．变量无定义，输出不确定

C．-1

D．1

【答案】A

【解析】在 C 语言中，NULL 和 0 的值都是一样的，但是为了目的和用途及容易识别的原因，NULL 用于指针和对象，0 用于数值。答案选择 A 选项。

3．若有代数式 $\sqrt{|n^x+e^x|}$（其中 e 仅代表自然对数的底数，不是变量），则以下能够正确表示该代数式的 C 语言表达式是（　　）。

A．sqrt(fabs(pow(n,x)+exp(x)))

B．sqrt(fabs(pow(n,x)+pow(x,e)))

C．sqrt(abs(n^x+e^x))

D．sqrt(fabs(pow(x,n)+exp(x)))

【答案】A

【解析】math.h 是数学运算库函数的头文件名：double sqrt(double)：平方根函数；double fabs(double)：绝对值函数；double pow(double x, double y)：返回 $x^y$；exp(double x)：返回 $e^x$。答案选择 A 选项。

4．以下叙述正确的是（　　）。

A．C 语言函数可以嵌套调用，例如：fun(fun(x))

B．C 语言程序是由过程和函数组成的

C．C 语言函数不可以单独编译

D．C 语言中除了 main 函数，其他函数不可作为单独文件形式存在

【答案】A

【解析】一个函数的返回值可以作为参数然后传给另一个函数，因此函数是可以嵌套调用的。B 项错误，C 语言程序只有函数构成，没有过程；C 项错误，编译系统的任务在于检查语法错误，只要符合语法规则的 C 程序都可以通过编译，就算是单独的函数也可以；D 项错误，在 C 语言中除 main()函数以外的其他函数可以和 main()函数在同一个 C 文件中，也可以单独处于其他的 C 文件，只要在使用到这些函数的 main()函数的 C 文件中用预

编译指令“#include”包含进来即可。答案选择 A 选项。

5．若函数调用时的实参为变量时，以下关于函数形参和实参的叙述中正确的是（　　）。

A．函数的形参和实参分别占用不同的存储单元

B．形参只是形式上的存在，不占用具体存储单元

C．同名的实参和形参占同一存储单元

D．函数的实参和其对应的形参共占同一存储单元

【答案】A

【解析】函数体中，变量（包括形参）只在函数被调用时才临时开辟存储单元，当退出函数时，这些临时开辟的存储单元全被释放掉。C 语言中函数调用可以分成传值和传引用，传值调用，形参是实参的数据拷贝；传引用调用，形参是实参的指针拷贝；所以，形参和实参占用不同的存储单元。答案选择 A 选项。

6．以下叙述中错误的是（　　）。

A．当在程序的开头包含头文件 stdio.h 时，可以给指针变量赋 NULL

B．函数可以返回地址值

C．改变函数形参的值，不会改变对应实参的值

D．可以给指针变量赋一个整数作为地址值

【答案】D

【解析】A 项正确，NULL 是在头文件 stdio.h 中定义的符号常量；B 项正确，函数的返回值可以是地址，即指针；C 项正确，函数调用中形参值的变化不会传递给实参；D 项错误，不能将一个整数直接赋给指针变量作为地址，只能用取地址运算符“&”把该整型变量的地址赋值给该指针变量。答案选择 D 选项。

7．以下叙述中错误的是（　　）。

A．C 程序必须由一个或一个以上的函数组成

B．函数调用可以作为一个独立的语句存在

C．若函数有返回值，必须通过 return 语句返回

D．函数可以通过实际参数和形式参数之间进行数据传递

【答案】C

【解析】C 项错误，比如 main 函数中有 exit(0)，则可以通过 exit 函数返回状态。A 项正确，C 程序至少有一个 main 函数；B 项正确，C 语言中的函数可以仅进行某些操作而不返回函数值，这时函数的调用可作为一条独立的语句；D 项正确，当函数调用为传引用时，形参指针和实参指针指向同一块内存，修改形参的同时也就修改了实参。答案选择 C 选项。

8．以下叙述中正确的是（　　）。

A．不同函数的形式参数不能使用相同名称的标识符

B．用户自己定义的函数只能调用库函数

C．实用的 C 语言源程序总是由一个或多个函数组成

D．在 C 语言的函数内部，可以定义局部嵌套函数

【答案】C

【解析】一个 C 程序可以有一个或多个程序文件，所以也可以有一个或多个函数，选项 C 正确；除了 main 函数不能被其他函数调用之外，用户自定义函数可以调用其他任意函数，包括库函数和用户自定义函数，选项 B 错误；函数是一个独立的模块，不同函数之间的定义相互没有影响，所以标识符可以相同，选项 A 错误；函数体内部只可以调用但是不可以定义其他函数，选项 D 错误；答案选择 C 选项。

9．C 语言程序中，若函数无返回值，则应该对函数说明的类型是（　　）。

A．int

B．double

C．char

D．void

【答案】D

【解析】A 项，int 表示返回值是整型；B 项，double 表示返回值是双精度型；C 项，char 表示返回值是字符型；D 项，void 表示无返回值。答案选择 D 选项。

10．以下叙述中错误的是（　　）。

A．函数的返回值类型不能是结构体类型，只能是简单类型

B．函数可以返回指向结构体变量的指针

C．可以通过指向结构体变量的指针访问所指结构体变量的任何成员

D．只要类型相同，结构体变量之间可以整体赋值

【答案】A

【解析】函数的返回值类型可以是结构体类型，也可以是指向结构体变量的指针类型，相同类型结构体变量之间可以整体赋值，可以通过指针变量引用结构体成员。答案选择 A 选项。

11．以下关于 return 语句的叙述中正确的是（　　）。

A．一个自定义函数中必须有一条 return 语句

B．一个自定义函数中可以根据不同情况设置多条 return 语句

C．定义成 void 类型的函数中可以有带返回值的 return 语句

D．没有 return 语句的自定义函数在执行结束时不能返回到调用处

【答案】B

【解析】B 项，一个自定义函数可以根据不同的情况设置多条 return 语句，但注意函数的返回值必须只有一个；A 项，一个自定义函数中可以没有返回值，只要定义为 void 型即可；C 项，定义成 void 类型的函数没有返回值，所以其中不可以有 return 语句；D 项，程序执行结束后无论是否有 return 语句都自动返回到调用处。答案选择 B 选项。

12．以下叙述中错误的是（　　）。

A．用户定义的函数中可以没有 return 语句

B．用户定义的函数中可以有多个 return 语句，以便可以调用一次返回多个函数值

C．用户定义的函数中若没有 return 语句，则应当定义函数为 void 类型

D．函数的 return 语句中可以没有表达式

【答案】B

【解析】用户定义的函数有两种：①void 函数，可以没有 return 语句，如果有 return 语句，也不可以返回任何表达式；②指定返回类型函数，至少有一个返回语句。在一个函数内，可以根据需要在多处出现 return 语句，但无论有多少个 return 语句，return 语句只会被执行一次然后退出函数，并且只能返回一个函数值。AC 两项，在没有返回值的函数中可以没有 return 语句，函数类型定义为 void 即可；D 项，在没有返回值的函数中如果有 return 语句，该语句必须不带任何表达式；B 项，用户定义的函数可以有多个 return 语句，但是只能返回一个函数值。答案选择 B 选项。

13．以下关于函数的叙述中正确的是（　　）。

A．函数调用必须传递实参

B．函数必须要有形参

C．函数必须要有返回值

D．函数形参的类型与返回值的类型无关

【答案】D

【解析】函数如果没有形参也就不必传递实参，A 项错误；函数可以没有形参，B 项错误；函数可以没有返回值，C 项错误；函数形参的类型与返回值的类型无关，D 项正确。答案选择 D 选项。

14．下面的函数调用语句中 func 函数的实参个数是（　　）。

```
func(f2(v1,v2), (v3,v4,v5), (v6,max(v7,V8)));
```

A．3

B．4

C．5

D．8

【答案】A

【解析】函数在被调用时，传入的实参以逗号分隔，实参可以是一个变量，也可以是一个表达式。在本题中，func()函数传入的参数是被逗号分隔的 3 个表达式，它们分别是 f2(v1,v2)、(v3,v4,v5）和(v6,max(v7,v8))，所以它的实参个数是 3。其中，f2(v1,v2)是一个函数调用；(v3,v4,v5)是一个逗号表达式；(v6,max(v7,v8))也是一个逗号表达式，它里面还包含了一个函数调用，但它仍只代表 1 个实参。答案选择 A 选项。

15．函数调用语句：fun((exp1,exp2),(exp1,exp2,exp3));含有的实参个数是（　　）。

A．1

B．4

C．5

D．2

【答案】D

【解析】函数 fun 参数列表中有两个参数，多个参数之间用逗号分隔，分别是逗号表达式“(exp1,exp2)”的值与“(exp1,exp2,exp3)”的值。答案选择 D 选项。

16．avg 函数的功能是求整型数组中的前若干个元素的平均值，设数组元素个数最多不超过 10，则下列函数说明语句错误的是（　　）。

A．int avg(int *a,int n);

B．int avg(int a[10],int n);

C．int avg(int a,int n);

D．int avg(int a[],int n);

【答案】C

【解析】C 项，函数第一个形参的数据类型是 int 型，只能传递单个整型数值。而 a 表示整型数组的首地址，不能直接传递给它，因此不符合题目要求，声明不正确。答案选择 C 选项。

17．以下程序的主函数中调用了在其前面定义的函数 fun：

```
#include <stdio.h>
…
main()
{
   double a[15],k;
   k=fun(a);
   …
}
```

则以下选项中错误的 fun 函数首部是（　　）。

A．double fun(double a[15])

B．double fun(double*a)

C．double fun(double a[])

D．double fun(double a)

【答案】D

【解析】由题目可知，fun 函数中定义的应该是一个 double 型的指针变量。D 项定义的是双精度型浮点数变量。答案选择 D 选项。

18．在 C 语言程序中，下列说法中正确的是（　　）。

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．函数的定义和调用均不可以嵌套

D．函数的定义和调用均可以嵌套

【答案】B

【解析】C 语言规定，不能在函数的内部定义函数，但函数的调用可以嵌套。答案选择 B 选项。

19．设有函数定义：

void sub(int k,char ch){…}

则以下对函数 sub 的调用语句中，正确的是（　　）。

A．sub(1,97);

B．sub(2,'97');

C．h=sub(3,'a');

D．sub(4,"a");

【答案】A

【解析】函数的参数有两个，第一个是整型，第二个是字符类型，在调用函数时，实参必须一个是整型，另一个是字符型相容的类型（整型）。A 项中，97 为字符'a'的 ASCII 码，可以作为字符传入函数，A 项正确。B 项中，'a'不是合法字符，字符单引号里面只能有一个字符，B 项错误。函数 sub 返回类型为空，即不返回任何值，故 C 项中将函数返回值赋给 h 是不正确的调用方法，C 项错误。D 项，"a"为字符串，也不是合法的字符，D 项错误。答案选择 A 选项。

20．有以下程序：

```
#include <stdio.h>
int f(int x);
main()
{
   int a,b=0;
   for(a=0;a<3;a++)
   {
      b=b+f(a);
      putchar('A'+b);
   }
}
int f(int x)
{
   return x*x+1;
}
```

程序运行后的输出结果是（　　）。

A．ABE

B．BDI

C．BCF

D．BCD

【答案】B

【解析】第一次调用函数 f，返回 0*0+1=1，'A'+1 为 B，输出 B；第二次调用函数 f，返回 f(1)=2，b=1+2=3，输出 D；第三次为调用函数 f，返回 f(2)=5，b=3+5=8，输出 I。答案选择 B 选项。

21．有以下函数：

```
#include <stdio.h>
void func(int n)
{
   int i;
   for(i=0;i<=n;i++)printf("*");
   printf("#");
}
main()
{
   func(3);
   printf("????");
   func(4);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A. ****#????***#

B. ***#????*****#

C. **#????*****#

D. ****#????*****#

【答案】D

【解析】从 main 函数开始，执行 func(3)函数，for 循环执行 4 次，连续输出四个“*”，然后输出一个“#”；输出“????”；再次调用 func(4)，for 循环执行 5 次，连续输出五个“*”，然后输出一个“#”；最后输出换行符。答案选择 D 选项。

22．有以下函数：

```
#include <stdio.h>
void exch(int t[])
{
   t[0] = t[5];
}
main()
{
   int x[10] = {1,2,3,4,5,6,7,8,9,10},i=0;
   while(i <= 4)
   {
      exch(&x[i]);i++;
   }
   for(i = 0;i < 5;i++)printf("%d",x[i]);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A. 2 4 6 8 10

B. 1 3 5 7 9

C. 1 2 3 4 5

D. 6 7 8 9 10

【答案】D

【解析】函数 exch 的作用就是把数组某个元素的值变为这个元素位置加 5 后对应的元素值，所以执行完之后输出数组 x 的前五个元素的值，其实就是数组 x 的后五个元素的值，答案选择 D 选项。

23．有如下程序：

```
#include <stdio.h>
int sum(int a,int b)
{
    return a+b-2;
}
main()
{
    int i;
    for(i = 0;i < 5;i++)
        printf("%d",sum(i,3));
    printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．54321

B．01234

C．45678

D．12345

【答案】D

【解析】本题程序执行过程为：在 i=0，1，2，3，4 时依次输出 sum(i,3)，即调用 sum 函数后依次输出 1、2、3、4、5，答案选择 D 选项。

24．有以下程序

```
#include <stdio.h>
int f(int n)
{
    int t=0,a=5;
    if(n/2)
    {
        int a=6;
        t+=a++;
    }
    else
    {
        int a=7;
        t+=a++;
    }
    return t+a++;
}
main()
{
    int s=0,i=0;
    for(;i<2;i++) s+=f(i);
    printf("%d\n",s);
}
```

程序运行后的输出结果是（　　）。

A．24

B. 28
C. 32
D. 36
【答案】A
【解析】本题中需要注意在函数 f 中，最后的 return 语句中的变量 a 是函数初始定义的 a，它的值始终是 5，与 if 和 else 的执行语句中的 a 不是同一个变量。要注意区分变量的有效范围。执行 for 循环，i=0 时，s=s+f(0)=0+f(0)，f(0)=12，s=12；i=1 时，s=s+f(1)=12+f(1)，f(1)=12，s=24；i=2 时循环结束。答案选择 A 选项。

25．有以下程序：

```
#include <stdio.h>
double f(double x);
main()
{
   double a=0;
   int i;
   for(i=0;i<30;i+=10) a+=f((double)i);
   printf("%5.0f\n",a);
}
double f(double x)
{
   return x*x+1;
}
```

程序运行后的输出结果是（　　）。
A.　503
B.　401
C.　500
D.　1404
【答案】A
【解析】函数 f 的功能是计算传入的实参 x 的平方加一，然后将结果返回。i 初始值为 0，根据 x*x+1，即第一次返回值为 1，然后执行 i=i+10，第二次 i 的值为 10，即返回值为 101+1=102，第三次 i 的值为 20，即返回值为 401+102=503。注意 5 位输出，位数不够的，补空格。答案选择 A 选项。

26．有以下程序：

```
#include <stdio.h>
int fun(int x, int y)
{
   if(x==y)return(x);
   else return((x+y)/2);
}
main()
{
   int a=4,b=5,c=6;
   printf("%d\n", fun(2*a,fun(b,c)));
}
```

程序运行后的输出结果是（　　）。
A. 3
B. 6
C. 8

D．12

【答案】B

【解析】fun 的功能是求两数的平均数，并且参数和结果都为整型。题中是用一个函数作为另一个函数的参数，所以应该先做里面嵌套的函数即 fun(b,c)，值为 5。fun(2*a,fun(b,c))即为计算 fun(2*4,5)，所以结果为(5+8)/2=6。答案选择 B 选项。

27．有以下程序

```
#include <stdio.h>
int f(int x);
main()
{
   int n=1,m;
   m=f(f(f(n)));
   printf("%d\n",m);
}
int f(int x)
{
   return x*2;
}
```

程序运行后的输出结果是（　　）。

A．1

B．2

C．4

D．8

【答案】D

【解析】考查函数嵌套调用，第一次调用的是最内层的 f(n)，即 f(1)返回值 2。第二次调用中间的 f(f(n))，即 f(2)返回值 4。最后调用最外层的 f(f(f(n)))，即 f(4)返回值 8，最后打印输出。答案选择 D 选项。

28．有以下程序：

```
#include <stdio.h>
int f(int x, int y)
{
   return((y-x)*x);
}
main()
{
   int a=3,b=4,c=5,d;
   d=f(f(a,b),f(a,c));
   printf("%d\n",d);
}
```

程序运行后的输出结果是（　　）。

A．10

B．9

C．8

D．7

【答案】B

【解析】主函数中语句 d=f(f(a,b),f(a,c));用函数的返回值做参数，所以首先计算 f(a,b)代入 a=3，b=4，得 3，然后计算 f(a,c)，代入 a=3，c=5 得 6，所以 d=f(f(a,b),f(a,c));，即为计算 d=f(3，6)，值为 9。答案选择 B 选项。

29．有以下程序

```
#include <stdio.h>
int fun(int a,int b)
{
   return a+b;
}
main()
{
   int x = 6,y=7,z = 8,r;
   r = fun(fun(x,y),z--);
   printf("%d\n",r);
}
```

程序运行后的输出结果是（　　）。

A．15

B．21

C．20

D．31

【答案】B

【解析】函数的嵌套调用，fun 函数的作用是计算两个整数的和，r=fun(fun(x, y), z--)等价于：temp = fun(x,y)，r = fun(temp, z--)，则 temp = fun(6, 7) = 13，然后 r = fun(13,8) = 21，最后执行 z--，得到 z = 7。答案选择 B 选项。

30．有以下程序

```
#include <stdio.h>
int f(int x);
main()
{
   int n=1,m;
   m=f(f(f(n)));
   printf("%d\n",m);
}
int f(int x)
{
   return x*2;
}
```

程序运行后的输出结果是（　　）。

A．4

B．2

C．8

D．1

【答案】C

【解析】考查函数的递归调用。m = f(f(f(n))) = f(f(2)) = f(4) = 8，答案选择 C 选项。

31．有以下程序

```
#include <stdio.h>
int fun1(double a)
{
   return (int)(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);
   printf("%4.2f",w);
}
```

程序执行后输出结果是（　　）。

A．5

B．5.00

C．5.21

D．0.0

【答案】B

【解析】fun1(1.1)的值为 1，fun1(2.0)的值为 4，所以 fun2(1.1,2.0)函数返 5，%4.2f 要求输出的实数宽度为 4，其中小数点后有 2 位，按格式输出后为 5.00。答案选择 B 选项。

32．若有以下函数首部

int fun(double x[10],int*n)

则下面针对此函数的函数声明语句中正确的是（　　）。

A．int fun(double*,int*);

B．int fun(double,int);

C．int fun(double *x,int n);

D．int fun(double x,int*n);

【答案】A

【解析】函数声明应该和函数调用的参数保持一致，声明时的参数（形参）名字可以省略，数组在参数传递过程中变成指针。答案选择 A 选项。

33．若各选项中所用变量已正确定义，函数 fun 中通过 return 语句返回一个函数值，以下选项中错误的程序是（　　）。

A.

```
float fun(int a,int b){……}
main()
{……x=fun(i,j);……}
```

B.

```
main()
{……x=fun(2,10);……}
float fun(int a,int b){……}
```

C.

```
float fun(int,int);
main()
{……x=fun(2,10);……}
float fun(int a,int b){……}
```

D.

```
main()
{
   float fun(int i,int j);
   ……x=fun(i,j);……
}
float fun(int a,int b){……}
```

【答案】B

【解析】当一个函数调用另一个函数前，必须在该调用语句之前有被调用函数的声明或者定义，否则，调用函数无法识别被调用函数。B 项错误，main 函数不能识别 fun 函数。答案选择 B 选项。

34．有以下程序：

```
#include <stdio.h>
main()
{
   int findmax(int,int,int),m;
   …
   m=findmax(a,b,c);
   …
}
int findmax(int x,int y,int z)
{
   …
}
```

则以下叙述正确的是（　　）。

A．在 main 函数中声明了 findmax 函数

B．在 main 函数中定义了 findmax 函数

C．在 main 函数中两次调用了 findmax 函数

D．在 main 函数内、外重复定义了 findmax 函数

【答案】A

【解析】程序中函数 findmax 定义放在主函数之后，主函数中声明了 findmax 函数。“int findmax(int,int,int);”为函数声明，“m=findmax(a,b,c);”为函数调用，“int findmax(int x,int y,int z){}”为函数定义。B 选项是在 main 函数外定义的 findmax 函数，错误。在 main 函数中，出现了一次 findmax 函数的声明和一次 findmax 函数的调用，不是两次调用，C 选项错误。D 选项在 main 函数内不是定义 findmax 函数而是声明 findmax 函数。答案选择 A 选项。

35．以下关于 C 语言函数参数传递方式的叙述正确的是（　　）。

A．数据只能从实参单向传递给形参

B．数据可以在实参和形参之间双向传递

C．数据只能从形参单向传递给实参

D．C 语言的函数，参数既可以从实参单向传递给形参，也可以在实参和形参之间双向传递，可视情况选择使用

【答案】A

【解析】数据只能由实参单向传递给形参称为“值传递”，而不能由形参传给实参，A 项正确，B、C、D 错

误；数组名、指针等作参数，实参传递给形参的是地址值，这样形参和实参就指向同一段内存单元，在函数体内对形参数据的改变也将影响到实参。答案选择 A 选项。

36．有以下程序：

```
#include <stdio.h>
void fun(int p)
{
   int d=2;
   p=d++;
   printf("%d",p);
}
main()
{
   int a=1;
   fun(a);
   printf("%d\n",a);
}
```

程序运行后的输出结果是（　　）。

A．32

B．12

C．21

D．22

【答案】C

【解析】C 语言中函数参数传递满足“单向传递”，实现传递值的功能，实参能传给形参，形参却不能传回给实参。fun 函数体内输出 p 的值为 2，并不影响到 fun 函数外 a 的值，a 的值在 main 函数内依然为 1。答案选择 C 选项。

37．有以下程序：

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

程序运行的结果是（　　）。

A．1,2,3,4,5,6,7,8,9,0,

B．2,1,4,3,6,5,8,7,0,9,

C．0,9,8,7,6,5,4,3,2,1,

D．0,1,2,3,4,5,6,7,8,9,

【答案】A

【解析】在 C 语言中，函数参数传递的作用是“传值”，形参和实参是两个没有关系的变量。函数 fun 交换了参数值，但只是交换了形参的值，结果并不会传递给实参。所以数组 c 没有发生变化，原顺序输出。答案选择 A 选项。

38．设有定义语句 int(*f)(int);，则以下叙述中正确的（　　）。

A．f 是基类型为 int 的指针变量

B．f 是指向函数的指针变量，该函数具有一个 int 类型的形参

C．f 是指向 int 类型一维数组的指针变量

D．f 是函数名，该函数的返回值是基类型为 int 类型的地址

【答案】B

【解析】在 C 语言中函数名代表该函数的入口地址，因此可以定义一种指向函数的指针来存放这类地址，int(*f)(int);，其中 f 为指向函数的指针变量，指向有一个整型变量且返回值也为整型的函数，第一个 int 为函数返回值的类型，第二个 int 为函数的形参类型。答案选择 B 选项。

39．设有以下函数：

void fun(int n,char* s){……}

则下面对函数指针的定义和赋值均是正确的是（　　）。

A．void (* pf)();pf=fun;

B．void *pf();pf=fun;

C．void * pf(); *pf=fun;

D．void (* pf)(int,char);pf=&fun;

【答案】D

【解析】函数指针的一般定义形式为：

返回值类型 (* 指针变量名) ([形参列表]);

其中，“返回值类型”说明函数的返回类型，“(* 指针变量名)”中的括号不能省，括号改变了运算符的优先级。若省略整体则成为一个函数说明，说明了一个返回的数据类型是指针的函数，后面的“形参列表”表示指针变量指向的函数所带的参数列表。

以本题函数和函数指针为例，将函数的首地址赋给指针，可以是 pf=fun;或者 pf=&fun;

选项 A，参数列表与题干函数不符，错误。选项 B，函数指针定义格式错误。选项 C。函数指针定义格式错误，复制格式也错误。答案选择 D 选项。

40．有以下程序

```
#include <stdio.h>
int add(int a,int b)
{
   return (a+b);
}
main()
{
   int k,(*f)(),a=5,b=10;
   f=add;
   …
}
```

则以下函数调用语句错误的是（　　）。

A．k=*f(a,b);

B．k=add(a,b);

C．k=(*f)(a,b);

D．k=f(a,b);

【答案】A

【解析】int (*f)(int, int)声明了一个函数指针，它可以指向一个函数，该函数的形参是两个 int，返回值是 int；f=add，是把函数 add 的地址赋给指针 f，选项 B 正确；函数指针的调用有两种方式，(*f)()和 f()，选项 C、D 正确。f(a,b)已经表示调用函数 add，返回 15，k=*15 出现编译错误，选项 A 错误。答案选择 A 选项。

41．以下叙述中正确的是（　　）。

A．简单递归不需要明确的结束递归的条件

B．任何情况下都不能用函数名作为实参

C．函数的递归调用不需要额外开销，所以效率很高

D．函数既可以直接调用自己，也可以间接调用自己

【答案】D

【解析】D 项正确，C 语言中的函数可以直接或间接地自己调用自己，前者称简单递归，后者称间接递归。A 项错误，递归必定要有一个明确的结束递归的条件；B 项错误，递归就是把函数名作为实参的一种特殊情况；C 项错误，函数的递归调用过程中，系统要为每一层调用中的变量开辟存储单元，记住每一层调用后的返回点，要增加许多额外的开销，通常会降低程序的运行效率。答案选择 D 选项。

42．有以下程序：

```
#include <stdio.h>
int fun(int n)
{
   if(n==1)
      return 1;
   else
      return(n+fun(n-1));
}
main()
{
   int x;
   scanf("%d",&x);
   x=fun(x);
   printf("%d\n",x);
}
```

执行程序时，给变量 x 输入 10，程序的输出结果是（　　）。

A．55

B．54

C．65

D．45

【答案】A

【解析】fun 函数是一个递归函数，用于求整数 1 到 n 的和。因此，主函数中最后输出 1 到 10 之间的整数和，即 55。答案选择 A 选项。

43．有以下程序：

```
#include<stdio.h>
int fun(int n)
{
   if(n)return fun(n-1)+n;
   else return 0;
}
main()
{
   printf("%d\n",fun(3));
}
```

程序的运行结果是（　　）。

A. 4

B. 5

C. 6

D. 7

【答案】C

【解析】fun 函数是一个递归函数，调用 f(3)，参数 n=3，返回 f(3-1)+3 = f(2)+3 = f(1)+2+3 = f(0)+1+2+3 = 0+6 = 6。答案选择 C 选项。

44. 有以下函数：

```
#include <stdio.h>
void fun(char c)
{
   if(c>'x')fun(c-1);
   printf("%c",c);
}
main()
{
   fun('z');
}
```

程序运行后的输出结果是（　　）。

A. xyz

B. wxyz

C. zyxw

D. zyx

【答案】A

【解析】函数 fun 是一个递归函数，递归的终止条件是输入的实参字符小于等于字符'x'。先从 main 函数运行，参数为'z'，满足 c>'x'，执行 fun 函数，参数变为'y'，满足 c>'x'，执行 fun 函数，此时，参数为'x'不再满足条件，开始逆序输出 z，返回上一级调用输出 y，返回最初调用输出 z。答案选择 A 选项。

45. 有如下程序：

```
#include <stdio.h>
void convert(char ch)
{
   if(ch< 'D')convert(ch+1);
   printf("%c",ch);
}
main()
{
   convert('A');
   printf("%\n");
}
```

程序运行后的输出结果是（　　）。

A．ABCDDCBA

B．ABCD

C．A

D．DCBA

【答案】D

【解析】本题程序的执行过程为：调用 convert('A')函数，ch='A'，if 条件成立，执行 convert('B') → convert('C') → convert('D')，ch= 'D'时，if 条件不成立，输出 D→执行 convert('C')中 if 语句后的输出语句，输出 C→执行 convert('B')中 if 语句后的输出语句，输出 B→执行 convert('A')中 if 语句后的输出语句，输出 A，函数调用完成。程序运行后的输出结果是 DCBA。答案选择 D 选项。

46．设有如下函数定义：

```
int fun(int k)
{
   if(k<1) return 0;
   else if(k==1) return 1;
   else return fun(k-1)+1;
}
```

若执行调用语句：n=fun(3);，则函数 fun()总共被调用的次数是（　　）。

A．2

B．3

C．4

D．5

【答案】B

【解析】函数 fun 为递归函数，递归结束条件是 k 为小于等于 1 的数。执行 fun(3)语句时会返回 fun(3-1)+1，即 fun(2)+1；执行 fun(2)时会返回 fun(2-1)+1，即 fun(1)+1；执行 fun(1)时会返回 1，所以函数 fun 总共被调用 3 次。答案选择 B 选项。

47．以下程序：

```
#include <stdio.h>
void fun(int x)
{
   if(x/2>1) fun(x/2);
   printf("%d",x);
}
main()
{
   fun(7);
   printf("\n");
}
```

程序运行后的结果是（　　）。

A. 137

B. 731

C. 73

D. 37

【答案】D

【解析】函数 fun 是递归函数，递归终止条件是 x 小于等于 3，当大于 3 时递归调用 fun(x/2)。第一次循环 x/2=3，调用 fun(3)，x/2=1，跳出 if 循环，执行输出 x=3，然后回到第一次循环，执行 if 循环的语句，输出 x=7。答案选择 D 选项。

48. 有以下程序：

```
#include <stdio.h>
void fun(int n)
{
   int i;
   if((i=n/10)!=0)
      fun(i);
   putchar(n%10+'0');
}
main()
{
   fun(256);
}
```

程序运行后的输出结果是（　　）。

A. 256

B. 652

C. 2560

D. 52

【答案】A

【解析】程序的执行过程为：调用函数 fun(256)，i=25，即 i!=0，if 条件成立，调用 fun(25)，i=2，即 i!=0，if 条件成立，调用 fun(2)，i=0，if 条件不成立，输出字符'2'，返回 fun(25)，输出字符'5'，返回 fun(256)，输出字符'6'，函数调用结束。答案选择 A 选项。

49. 有如下程序：

```
#include<stdio.h>
void get_put()
{
   char ch;
   ch=getchar();
   if(ch!='\n')get_put();
   putchar(ch);
}
main()
{
   get_put();
   printf("\n");
}
```

程序运行时，输入 1234<回车>，则输出结果是（　　）。

A. 1234

B. 4321

C. 4444

D. 1111

【答案】B

【解析】get_put 为递归函数。本题程序的执行过程为：在输入 1234<回车>的情况下，调用 get_put 函数，getchar 读入 1，即 ch='1'，if 条件成立，调用 get_put 函数；读入 2，ch='2'，调用 get_put 函数；读入 3，ch='3'，调用 get_put 函数；读入 4，ch='4'，调用 get_put 函数；读入回车，ch='\n'，if 条件不成立，返回执行每次调用函数中 if 语句后的输出语句，即依次输出 4321。答案选择 B 选项。

50．有如下程序：

```
#include<stdio.h>
void get_put()
{
   char ch;
   ch=getchar();
   if(ch!='\n')get_put();
   putchar(ch);
}
main()
{
   get_put();
}
```

程序运行时，输入 ABCD<回车>，则输出结果是（　　）。

A. DCDC

B. DCBA

C. BABA

D. ABCD

【答案】B

【解析】本题考查了函数的递归调用，进入 get_put 函数后，字符变量 ch 每次保存读进来的字符，如果不是换行符，则继续读取下一个字符常量，当 ch 保存完'D'后，下一次读取不满足 if 条件，函数进行输出，依次输出'D'，'C'，'B'，'A'，答案选择 B 选项。

51．有以下程序：

```
#include<stdio.h>
void fac2(int);
void fac1(int n)
{
   printf("*");
   if(n>0)fac2(n-1);
}
void fac2(int n)
{
   printf("#");
   if(n>0)fac2(--n);
}
main()
{
   fac1(3);
}
```

程序的运行结果是（　　）。

A. *###

B. *##*

C. **##f

D. *#*#

【答案】A

【解析】函数 fac1 中嵌套函数 fac2，fac2 为递归函数。程序执行过程为：调用函数 fac1(3)，输出*，3>0 成立，调用函数 fac2(2)，输出#，2>0 成立，调用 fac2(1)，输出#，1>0 成立，调用 fac2(0)，输出#，0>0 不成立，返回 fac2(1)，再返回 fac2(2)，再返回 fac1(3)，函数调用结束。程序的运行结果是：*###，答案选择 A 选项。

52．有以下程序

```
#include<stdio.h>
int fun(int x)
{
   int p;
   if(x==0||x==1)
      return (3);
   p=x-fun(x-2);
   return p;
}
main()
{
   printf("%d\n",fun(7));
}
```

执行后的输出结果是（　　）。

A. 2

B. 3

C. 7

D. 0

【答案】A

【解析】函数的递归调用，调用过程：①函数调用 f(7)，返回 7-f(5)；②函数调用 f(5)，返回 5-f(3)；③函数调用 f(3)，返回 3-f(1)；函数调用 f(1)，返回 3。输出 f(7) = 7-f(5) = 7-(5-f(3)) = 7-(5-(3-f(1))) = 7-(5-(3-3)) = 2。答

案选择 A 选项。

53．有以下程序：

```
#include<stdio.h>
int f(int x)
{
   int y;
   if(x==0||x==1)
      return (3);
   y=x*x-f(x-2);
   return y;
}
main()
{
   int z;
   z=f(3);
   printf("%d\n",z);
}
```

程序的运行结果是（　　）。

A．0

B．9

C．6

D．8

【答案】C

【解析】函数 f 是递归函数，递归的终止条件是 x 为 1 或 2。f(3)=3*3-f(1)，当 x=0 或 x=1 时返回值为 3，即 f(0)=3，f(1)=3；所以 y=3*3-3=6。答案选择 C 选项。

54．有以下程序

```
#include<stdio.h>
int f(int t[],int n);
main()
{
   int a[4]={1,2,3,4},s;
   s=f(a,4);
   printf("%d\n",s);
}
int f(int t[],int n)
{
   if(n>0)return t[n-1]+f(t,n-1);
   else return 0;
}
```

程序运行后的输出结果是（　　）。

A．4

B．10

C．14

D．6

【答案】B

【解析】函数 f 的功能是通过递归计算数组 t 中元素的和。在主函数中调用了递归函数 f(a,4)，将递归函数

f(a,4)的递归式展开，s = f(a,4) = a[3] + f(a,3) = a[3] + a[2] + f(a,2) = a[3] + a[2] + a[1] + f(a,1) = a[3] + a[2] + a[1] + a[0] + f(a,0) = a[3] + a[2] + a[1] + a[0] + 0 = 4 + 3 + 2 + 1 + 0 = 10。答案选择 B 选项。

55．有以下程序：

```
#include<stdio.h>
int fun(int a,int b)
{
   if(b==0)return a;
   else return(fun(--a,--b));
}
main()
{
   printf("%d\n",fun(4,2));
}
```

程序运行的结果是（　　）。

A．1

B．2

C．3

D．4

【答案】B

【解析】fun 函数是一个递归函数，其功能是：当 b 的值为零时，返回此时 a 的值；否则，返回 fun(--a,--b)，即 a 和 b 的值分别减 1 后递归调用返回 fun 函数。当 b 不断递减时，a 也不断递减，直到 b 为零。执行过程为：执行 fun(4,2)，b=2，返回 fun(3,1)，此时 b=1，返回 fun(2,0)，b=0，返回 a 的值 2。答案选择 B 选项。

56．若有以下程序

```
#include<stdio.h>
int f(int a[],int n)
{
   if(n>1)
   {
      int t;
      t=f(a,n-1);
      return t>a[n-1]?t:a[n-1];
   }
   else
      return a[0];
}
main()
{
   int a[]={8,2,9,1,3,6,4,7,5};
   printf("%d\n",f(a,9));
}
```

则程序的输出结果是（　　）。

A．1

B．9

C．8

D．5

【答案】B

【解析】函数的递归调用，调用过程如下表所示。

| 函数调用 | t | a[n-1] | 返回值 |
|---|---|---|---|
| f(a,9) | 9 | 5 | 9 |
| f(a,8) | 9 | 7 | 9 |
| f(a,7) | 9 | 4 | 9 |
| f(a,6) | 9 | 6 | 9 |
| f(a,5) | 9 | 3 | 9 |
| f(a,4) | 9 | 1 | 9 |
| f(a,3) | 8 | 9 | 9 |
| f(a,2) | 8 | 2 | 8 |
| f(a,1) | 8 | 8 | 8 |
| f(a,0) | - | - | 8 |

该段代码的作用是返回数组 a 里面的最大值。答案选择 B 选项。

57．有以下程序：

```
#include<stdio.h>
int f(int x[],int n)
{
   if(n>1)
   {
      f(&x[1],n-1);
      printf("%d,",x[0]);
   }
   else
      printf("%d,",x[0]);
}
main()
{
   int z[6]={1,2,3,4,5,6};
   f(z,6);
   printf("\n");
}
```

程序的运行结果是（　　）。

A．6,5,4,3,2,1,

B．1,1

C．1,1,1,1,1,1,

D．1,2,3,4,5,6,

【答案】A

【解析】本题中递归函数调用执行过程为：f(z,6)→f(&z[1],5)→f(&z[2],4)→f(&z[3],3)→f(&z[4],2)→f(&z[5],1)，输出 x[0]，也即 z[5]=6，返回执行 f(&z[4],2)之后的输出 x[0]语句，即 z[4]=5，同理再返回上一个 f 函数，依次输出 4，3，2，1，答案选择 A 选项。

58．有以下程序

```
#include<stdio.h>
void fun(int n,int *s)
{
   int f;
   if(n == 1)*s = n + 1;
   else
   {
      fun(n-1,&f);
      *s = f;
   }
}
main()
{
   int x = 0;
   fun(4,&x);
   printf("%d\n",x);
}
```

程序运行后的输出结果是（　　）。

A．3

B．1

C．2

D．4

【答案】C

【解析】函数 void fun(int n, int *s)有两个形参，第一个形参是传值，第二个形参是传引用，在函数 fun 内部改变 s 的值，则在函数 fun 外部也会发生相应改变。f 是函数内部的局部变量，递归调用时，把局部变量 f 以指针的形式传入，在下一层函数中会对 f 进行修改，当下一层函数返回后，再把当前局部变量 f 赋值给形参 s。函数调用过程如下表所示。

| 函数调用 | f | *s |
|---|---|---|
| fun(4, &x) | 2 | 2 |
| fun(3, &f) | 2 | 2 |
| fun(2, &f) | 2 | 2 |
| funs(1, &f) | - | 2 |

答案选择 C 选项。

59．有如下程序：

```
#include<stdio.h>
int sum(int *array,int len)
{
   if(len == 0)
      return array[0];
   else
      return array[0]+sum(array+1,len-1);
}
main()
{
   int array[5] = {1,2,3,4,5};
   int res = sum(array,4);
   printf("%d\n",res);
}
```

程序运行后的输出结果是（　　）。

A. 15

B. 10

C. 8

D. 1

【答案】A

【解析】程序执行过程为：调用函数 sum(array,4)，len=4；len>0 递归调用 sum(array+1,3)，传入地址为 array+1，即数组第二个元素地址；len=3>0，递归调用 sum(array+1,2)，传入地址为数组第三个元素地址；len=2>0，递归调用 sum(array+1,1)，传入地址为数组第四个元素地址；len=1>0 递归调用 sum(array+1,0)，传入地址为数组第五个元素地址，此时 len=0，返回 5；返回上一层调用执行 array[0]+sum(array+1,1)，即 4+5，返回 9；再返回上一层调用执行 array[0]+sum(array+1,2)，即 3+9，返回 12；同理，最后返回 15，并输出。本题中需要注意的是每一层调用时 array[0]是不一样的，并不是指 1。答案选择 A 选项。

60．在一个 C 源程序文件中所定义的全局变量，其作用域为（　　）。

A. 所在文件的全部范围

B. 所在程序的全部范围

C. 所在函数的全部范围

D. 由具体定义位置和 extern 说明来决定范围

【答案】D

【解析】全局变量是在函数外部任意位置上定义的变量，它的作用域是从变量定义的位置开始，到整个源文件结束止。答案选择 D 选项。

61．以下叙述中正确的是（　　）。

A. 只要是用户定义的标识符，都有一个有效的作用域

B. 只有全局变量才有自己的作用域，函数中的局部变量没有作用域

C. 只有在函数内部定义的变量才是局部变量

D. 局部变量不能被说明为 static

【答案】A

【解析】A 项正确，标识符的“作用域”是指在程序中的某一部分中，标识符是有定义的，可以被 C 编译和连接程序所识别。在 C 语言中，由用户命名的标识符都有一个有效的作用域。B 项错误，局部变量的作用域是所在的函数体（或复合语句）；C 项错误，在函数内部或复合语句内部定义的变量，称为局部变量；D 项错误，在函数体（或复合语句）内部用 static 来说明一个变量时，变量为静态局部变量。答案选择 A 选项。

62．以下叙述中正确的是（　　）。

A. 在复合语句中不能定义变量

B. 对于变量而言，“定义”和“说明”这两个词实际上是同一个意思

C. 全局变量的存储类别可以是静态类

D. 函数的形式参数不属于局部变量

【答案】C

【解析】C 项正确，对于全局变量可使用 extern 和 static 两种说明符。A 项错误，在复合语句中定义的变量是局部变量；B 项错误，“定义”（definition）是指给变量分配确定的存储单元，“说明”（declaration）只是说明变量的性质，而并不分配存储空间；D 项错误，函数的形式参数只能在函数内部被识别，属于局部变量。答案选择 C 选项。

63. 在 C 语言中，只有在使用时才占用内存单元的变量，其存储类型是（　　）。

A. auto 和 register

B. extern 和 register

C. auto 和 static

D. static 和 register

【答案】A

【解析】C 语言中，动态存储区域中存放的变量在使用时才分配内存空间。auto 变量的存储单元是分配在内存的动态存储区中，每当进入函数体时自动分配存储单元。register 变量也是自动类变量。static 说明的变量为静态变量，静态变量在内存的静态存储中占据着永久的存储单元，直至程序运行结束。extern 说明的变量为外部变量，属于全局变量，全局变量在整个程序运行期间都占用内存空间。答案选择 A 选项。

64. 当没有指定 C 语言中函数形参的存储类别时，函数形参的存储类别是（　　）。

A. 外部（extern）

B. 静态（static）

C. 寄存器（register）

D. 自动（auto）

【答案】D

【解析】auto 变量又称为自动变量，函数定义变量时，如果没有指定存储类别，系统就认为所定义的变量具有自动类别，D 选项正确。static 变量又称为静态变量，编译时为其分配的内存在静态存储区中。register 变量又称为寄存器变量，变量的值保留在 CPU 的寄存器中，而不是像一般变量那样占内存单元。当定义一个函数时，若在函数返回值的类型前加上说明符 extern 时，称此函数为外部函数，外部函数在整个源程序中都有效。答案选择 D 选项。

65. 以下叙述错误的是（　　）。

A. 未经赋值的全局变量值不确定

B. 未经赋值的 auto 变量值不确定

C. 未经赋值的 register 变量值不确定

D. 未经赋值的静态局部变量值为 0

【答案】A

【解析】C 语言中，未经赋值的全局变量默认初始化为 0，答案选择 A 选项。

66. 以下选项中叙述错误的是（　　）。

A. C 程序函数中定义的赋有初值的静态变量，每调用一次函数，赋一次初值

B. 在 C 程序的同一函数中，各复合语句内可以定义变量，其作用域仅限本复合语句内

C. C 程序函数中定义的自动变量，系统不自动赋确定的初值

D. C 程序函数的形参不可以说明为 static 型变量

【答案】A

【解析】函数内定义的静态变量，在整个程序运行期间，占据静态存储区的永久性存储单元。即使退出函数以后，下次再进入该函数时，静态局部变量仍使用原来的存储单元。对未赋初值的静态局部变量，C 编译程序自动给它赋初值 0。因此，在函数中定义的静态变量，只在第 1 次调用赋值，以后调用保留上次的值。答案选择 A 选项。

67．设函数中有整型变量 n，为保证其在未赋值的情况下初值为 0，应选择的存储类别是（　　）。

A．auto

B．register

C．static

D．auto 或 register

【答案】C

【解析】在 C 语言中，对于静态存储类型定义变量，如果未初始化，在编译时，系统会自动对其初始化默认值，其中 int 型的默认初始化值是 0，其他选项的存储类型不能保证变量在未赋值情况的初值。答案选择 C 选项。

68．有以下程序：

```
#include<stdio.h>
int sum(int *array,int len)
{
   if(len == 0)
      return array[0];
   else
      return array[0]+sum(array+1,len-1);
}
main()
{
   int i=1,j=3;
   printf("%d,",i++);
   {
      int i = 0;
      i+=j*2;
      printf("%d,%d,",i,j);
   }
   printf("%d,%d\n",i,j);
}
```

程序运行后的输出结果是（　　）。

A．1,6,3,1,3

B．1,6,3,2,3

C．1,6,3,6,3

D．1,7,3,2,3

【答案】B

【解析】程序执行过程为：输出 i=1，之后 i 自增，得 i=2；然后执行复合语句，赋值 i=0，执行语句 i+=j*2；得 i=6；输出 i=6，j=3；花括号内的 i 的作用域仅限于该花括号内，在括号外的 i 为程序第四行定义的 i，这两个 i 属于不同的变量。之后再输出 i=2，j=3。答案选择 B 选项。

69．有以下程序

```
#include<stdio.h>
int f(int n)
{
   int t=0,a=5;
   if(n/2)
   {
      int a=6;
      t+=a++;
   }
   else
   {
      int a=7;
      t+=a++;
   }
   return t+a++;
}
main()
{
   int s=0,i=0;
   for(;i<2;i++)s+=f(i);
   printf("%d\n",s);
}
```

程序运行后的输出结果是（　　）。

A．36

B．28

C．32

D．24

【答案】D

【解析】函数 f 中，不管是 if 语句块还是 else 语句块，a 都是局部变量。f(0)和 f(1)调用 f 函数都是执行 else 语句块。t+=a++等效于 t=t+a;a++;，else 语句块执行完后，t=7；函数 f 中，a 也是局部变量，执行 return 时，a=5，t=7，返回 12。s= f(0)+ f(1) = 12 + 12 =24。答案选择 D 选项。

70．有以下程序：

```
#include<stdio.h>
void fun(int   n)
{
   static int num=1;
   num=num+n;
   printf("%d",num);
}
main()
{
   fun(3);
   fun(4);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．48

B．34
C．35
D．45
【答案】A
【解析】静态变量在第 2 次调用 fun 函数时不会被重新初始化，在整个程序运行结束时才会被释放。第一次调用后 num 的值为 1+3=4，第二次调用时，num=num+4=4+4=8。答案选择 A 选项。

71．有以下函数：

```
#include<stdio.h>
void fun(int *s)
{
   static int j=0;
   do
   {
      s[j]=s[j]+s[j+1];
   }while(++j<2);
}
main()
{
   int k,a[10]={1,2,3,4,5};
   for(k=1;k<3;k++) fun(a);
   for(k=0;k<5;k++) printf("%d",a[k]);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。
A．12345
B．23445
C．34756
D．35745
【答案】D
【解析】从 main 函数开始，第一个 for 循环：k=1，满足条件 k<3，调用 fun(a)。fun()函数中，s 为指向数组 a 的指针，则有 s[0]=s[0]+s[1]=3，满足条件++j（此时 j=1）小于 2，继续执行 do-while 循环；s[1]=s[1]+s[2]=5，不满足条件++j（此时 j=2）小于 2，退出 do-while 循环。此时：k=2，满足条件 k<3，再次调用 fun(a)。由于 j 定义为静态变量，所以上次函数调用结束时 j=2 仍被保留，则有 s[2]=s[2]+s[3]=7，不满足条件++j（此时 j=3）小于 2，退出 do-while 循环。此时，k=3，不再满足循环条件，退出第一个 for 循环，执行第二个 for 循环，输出五个数，即 35745。答案选择 D 选项。

72．有以下程序：

```
#include<stdio.h>
int f(int m)
{
   static int n=0;
   n+=m;
   return n;
}
main()
{
   int n=0;
   printf("%d",f(++n));
   printf("%d\n",f(n++));
}
```

程序运行后的输出结果是（　　）。

A. 12

B. 11

C. 23

D. 33

【答案】A

【解析】在整个程序运行期间，静态局部变量在内存的静态存储区中占据着永久性的存储单元，可以继续使用存储单元中原来的值。程序先执行++n，即 n 先自增 1，再调用 f 函数，函数中进行 n+=m 运算，结果仍为 1，即第一个输出值即为 1；第二次调用 f 函数时，执行 n+=m 运算，结果为 2，返回输出，即输出值为 2，然后 n 再++。答案选择 A 选项。

73. 有以下程序

```
#include<stdio.h>
int fun()
{
   static int x=1;
   x*=2;
   return x;
}
main()
{
   int i,s=1;
   for(i=1;i<=2;i++)s=fun();
   printf("%d\n",s);
}
```

程序运行后的输出结果是（　　）。

A. 0

B. 1

C. 4

D. 8

【答案】C

【解析】对局部静态变量在编译时赋初值，以后每次调用函数时一直保持不变，只是保留上次函数调用结束时的值。在主函数中调用了两次 fun()函数，第一次调用时 x 初值为 1，执行 x*=2 后，x=2；第二次调用时初值为 2，执行 x*=2 后，x=4。答案选择 C 选项。

74．有以下程序：

```
#include<stdio.h>
int fun()
{
   static int x=1;
   x*=2;
   return x;
}
main()
{
   int i,s=1;
   for(i=1;i<=3;i++)s*=fun();
   printf("%d\n",s);
}
```

程序运行后的输出结果是（　　）。

A．0

B．10

C．30

D．64

【答案】D

【解析】在整个程序运行期间，静态局部变量在内存的静态存储区中占据着永久性的存储单元，可以继续使用存储单元中原来的值。此题中第一次循环时，调用 fun 函数后 x 的值为 2，s 的值为 2；第二次循环时，调用 fun 函数后 x 的值为 4，s 的值为 2*4=8；第三次循环时，调用 fun 函数后 x 的值为 8，s 的值为 8*8=64。答案选择 D 选项。

75．有以下程序

```
#include<stdio.h>
int fun()
{
   static int x=1;
   x+=1;
   return x;
}
main()
{
   int i,s=1;
   for(i=1;i<=5;i++)s+=fun();
   printf("%d\n",s);
}
```

程序运行后的输出结果是（　　）。

A．11

B．21

C．6

D．120

【答案】B

【解析】本题目 fun 函数中定义的变量 x 为静态局部变量，第一循环后 x 的值为 2，s 的值为 3；第二次循环后 x 的值为 3，s 的值为 6；第三次循环后 x 的值为 4，s 的值为 10；第四次循环后 x 的值为 5，s 的值为 15；第五次循环后 x 的值为 6，s 的值为 21。答案选择 B 选项。

76．有以下程序：

```
#include <stdio.h>
int f(int n);
main()
{
   int a=3,s;
   s=f(a);
   s=s+f(a);
   printf("%d\n",s);
}
int f(int n)
{
   static int a=1;
   n+=a++;
   return n;
}
```

程序运行以后的输出结果是（　　）。

A．7

B．8

C．9

D．10

【答案】C

【解析】在函数 f 中，整型变量 a 为静态变量，所以每次调用函数 f 时不再为 a 重新赋值，而且 a 的值只有在程序结束时才被释放。第一次调用 f 后 n=4，a=2，s=4；第二次调用时，a 初值为 2，调用后，a=3，n=5，s=4+5=9，所以输出结果为 9。答案选择 C 选项。

77．有以下程序

```
#include<stdio.h>
int 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（　　）。

A．5,11

B．5,5

C．11,11

D．11,5

【答案】A

【解析】fun 函数中 m 和 i 是静态变量，退出函数以后，下次再进入该函数时，m 和 i 仍然保持上一次的数值。第一次调用 fun(j,m)时，i=2+0+1=3，m=3+1+1=5，返回 5；第二次调用 fun(j,m)时，i=3+5+1=9，m=9+1+1=11，返回 11。答案选择 A 选项。

78．有以下程序：

```
#include<stdio.h>
int fun(int a,int b)
{
   static int m=0,i=2;
   i+=m+1;
   m=i+a+b;
   return m;
}
main()
{
   int k=4,m=1,p;
   p=fun(k,m);
   printf("%d,",p);
   p=fun(k,m);
   printf("%d\n",p);
}
```

程序运行后的输出结果是（　　）。

A．8,17

B．8,16

C．8,8

D．8,20

【答案】A

【解析】static 变量编译时，将其分配在内存的静态存储区中，在整个程序运行期间都不释放这些存储单元，即使退出函数，下次再进入该函数时，静态局部变量仍使用原来的存储单元，值是上一次函数调用结束时的值。程序执行过程为：调用函数，m=0，i=2，i=i+m+1=3，m=8，返回 p=8，再次调用函数，m=8，i=3，i=i+m+1=12，m=12+4+1=17，返回 p=17。答案选择 A 选项。

79．有以下程序

```
#include<stdio.h>
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（　　）。

A．50

B．60

C．45

D．55

【答案】B

【解析】函数 fun 的功能是求数组的 n 个元素之和，fun(a,5)=15，由于 sum 是静态局部变量，所以第二次调用时保持 15 不变，fun(b,4)=45，所以 s=fun(a,5)+fun(b,4)=60。答案选择 B 选项。

80．有以下程序：

```
#include<stdio.h>
int fun(int n)
{
   static int t=1;
   int i=1;
   for(;i<=n;i++)t*=i;
   return t;
}
main()
{
   int t=1,i;
   for(i=2;i<4;i++)
      t+=fun(i);
   printf("%d\n",t);
}
```

程序的运行结果是（　　）。

A．8

B．11

C．15

D．4

【答案】C

【解析】static 变量又称为静态变量，编译时，将其分配在内存的静态存储区中，在整个程序运行期间都不释放这些存储单元，即使退出函数，下次再进入该函数时，静态局部变量仍使用原来的存储单元，值是上一次函数调用结束时的值。程序执行过程为：执行 for 循环，i=2，t=1，调用函数 fun(2)，定义静态局部变量 t=1，此变量的作用域为函数 fun 中，定义局部变量 i=1，for 循环实现 t=1×1×2=2，返回 2，主函数中 t=1+2=3；当 i=3 时，t=3，调用函数 fun(3)，静态局部变量上次调用后结果 t=2，局部变量 i=1，for 循环实现 t=2×1×2×3=12，返回 12，t=3+12=15。最后输出 t 为 15，答案选择 C 选项。

81．有如下程序：

```
#include<stdio.h>
int sum(int data)
{
   static int init=1;
   return init+=data;
}
main()
{
   int i;
   for(i=1;i<=1;i++)printf("%d,",sum(i));
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A. 2,

B. 2,3,

C. 3,

D. 1,

【答案】A

【解析】在主函数中，调用了 sum 函数。sum 函数作用是定义一个静态变量 init=1，对变量 init 进行叠加形参 data 的值，并将结果作为函数的返回值。在 main 函数中，for 循环只执行依次，实参 i=1 传递给 sum 函数的形参 data，因此输出为 2。答案选择 A 选项。

82．有如下程序：

```
#include<stdio.h>
int sum(int data)
{
   static int init=0;
   return init+=data;
}
main()
{
   int i;
   for(i=1;i<=5;i++)printf("%d,",sum(i));
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A. 1,3,6,10,15,

B. 1,2,3,4,5,

C. 0,0,0,0,0,

D. 1,1,1,1,1,

【答案】A

【解析】在整个程序运行期间，静态局部变量在内存的静态存储区中占据着永久性的存储单元，可以继续使用存储单元中原来的值。程序执行过程为：执行 for 循环 i=1，调用函数 sum(1)，init=0，data=1，返回 init=0+1=1，输出 1；i=2，调用函数 sum(2)，data=2，init=1，返回 init=1+2=3，输出 3；i=3，调用函数 sum(3)，data=3，init=3，返回 init=3+3=6，输出 6；i=4，调用函数 sum(4)，data=4，init=6，返回 init=6+410，输出 10；i=5，调用函数 sum(5)，data=5，init=10，返回 init=10+5=15，输出 15；i=6，退出 for 循环。答案选择 A 选项。

83．有如下程序：

```
#include<stdio.h>
int *sum(int data)
{
   static int init=0;
   init+=data;
   return &init;
}
main()
{
   int i,*p;
   for(i=1;i<=4;i++) sum(i);
   p=sum(0);
   printf("%d\n",*p);
}
```

程序运行后的输出结果是（　　）。

A．15

B．0

C．1

D．10

【答案】D

【解析】static 变量编译时，将其分配在内存的静态存储区中，在整个程序运行期间都不释放这些存储单元，即使退出函数，下次再进入该函数时，静态局部变量仍使用原来的存储单元，值是上一次函数调用结束时的值。程序执行过程为：执行 for 循环，i=1，调用函数 sum(1)，data=1，init=0，init=init+data=1；i=2，调用函数 sum(2)，data=2，init=1，init=init+data=3；i=3，调用函数 sum(3)，data=3，init=3，init=init+data=6；i=4，调用函数 sum(4)，data=4，init=6，init=init+data=10；i=5 退出 for 循环。调用函数 sum(0)，data=0，init=10，init=init+data=10，返回指向 init 的指针，输出 init=10。答案选择 D 选项。

84．以下针对全局变量的叙述错误的是（　　）。

A．全局变量的作用域是从定义位置开始至源文件结束

B．全局变量是在函数外部任意位置上定义的变量

C．用 extern 说明符可以限制全局变量的作用域

D．全局变量的生存期贯穿于整个程序的运行期间

【答案】C

【解析】在不同编译单位内用 extern 说明符来扩展全局变量的作用域，extern 可以将全局变量作用域扩展到其他文件，而不是限制全局变量的作用域。选项 C 正确。

85．有以下程序

```
#include <stdio.h>
void fun2(char a,char b)
{
   printf("%c %c",a,b);
}
char a='A',b='B';
void fun1()
{
   a='C';
   b='D';
}
main()
{
   fun1();
   printf("%c %c ",a,b);
   fun2('E','F');
}
```

程序的运行结果是（　　）

A. C D E F

B. A B E F

C. A B C D

D. C D A B

【答案】A

【解析】调用 fun1 函数将全局变量 a 和 b 分别赋值为'C'和'D'，然后输出全局变量 a 和 b，最后调用 fun2 输出局部变量 a 和 b 的值，C 语言规定，在同一源文件中，允许全局变量和局部变量同名，在局部变量作用域内，同名的全局变量不起作用，因此输出'E'和'F'。答案选择 A 选项。

86．有以下程序：

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

程序运行后的输出结果是（　　）。

A. 1012

B. 810

C．1028

D．1016

【答案】C

【解析】因为 int 是全局变量，所以它的值在整个程序结束时才会消失。for 循环第一次 i=2，a[i]=3，所以 fun(&a[i])=3+2=5，这时 b 为 5，所以 b=fun(&a[i])+b=5+5=10。第二次循环时，b=10，然后 fun(&a[i])，代入 a[i]=4，这时 fun(&a[i])=10+4=14，所以 b=fun(&a[i])+b=14+14=28。答案选择 C 选项。

87．有以下程序：

```
#include <stdio.h>
int a=1,b=2;
void fun1(int a,int b)
{
   printf("%d%d",a,b);
}
void fun2()
{
   a=3;
   b=4;
}
main()
{
   fun1(5,6);
   fun2();
   printf("%d%d\n",a,b);
}
```

程序运行后的输出结果是（　　）。

A．1256

B．5634

C．5612

D．3456

【答案】B

【解析】在同一源文件中，允许全局变量和局部变量同名，在局部变量作用域内，同名的全局变量不起作用。第一次 fun1 传递参数 5，6，输出语句中的 a 和 b 是 fun1 内的局部变量，输出 5，6；第二次调用 fun2，fun2 的作用是将全局变量 a 变为 3，b 变为 4，主函数输出语句中的 a 和 b 是全局变量，故输出 3，4。答案选择 B 选项。

88．有以下程序：

```
#include <stdio.h>
int a=2;
int f()
{
   static int n;
   int m;
   m=n=0;
   n++;
   a++;
   m++;
   return m+n+a;
}
main()
{
   int k;
   for(k=0;k<3;k++)
      printf("%d,",f());
   printf("\n");
}
```

程序的运行结果是（　　）。

A. 5,6,7,

B. 5,7,9,

C. 5,8,11,

D. 5,5,5,

【答案】A

【解析】static 变量编译时，将其分配在内存的静态存储区中，在整个程序运行期间都不释放这些存储单元，即使退出函数，下次再进入该函数时，静态局部变量仍使用原来的存储单元，值是上一次函数调用结束时的值。程序执行过程为：k=0，a=2，n=0，m=0；n=1，a=3，m=1，返回 5；k=1，a=3，n=0，m=0；n=1，a=4，m=1，返回 6；k=2，a=4，n=0，m=0；n=1，a=5，m=1，返回 7；k=3 退出循环。答案选择 A 选项。

89. 有以下程序

```
#include<stdio.h>
int a=4;
int f(int n)
{
   int t=0;
   static int a=5;
   if(n%2)
   {
      int a=6;
      t+=a++;
   }
   else
   {
      int a=7;
      t+=a++;
   }
   return t+a++;
}
main()
{
   int s=a,i=0;
   for(;i<2;i++)s+=f(i);
   printf("%d\n",s);
}
```

程序运行后的输出结果是（　　）。

A．36

B．24

C．32

D．28

【答案】D

【解析】函数 f 中，不管是 if 语句块还是 else 语句块，a 都是局部变量。f(0)和 f(1)调用 f 函数都是执行 else 语句块。t+=a++等效于 t=t+a;a++;，else 语句块执行完后，t=7；函数 f 中，a 也是局部变量，执行 return 时，a=5，t=7，返回 12。s=4+ f(0)+ f(1) = 12 + 12 =28。答案选择 D 选项。

二、填空题

1．给定程序中，函数 fun 的功能是计算下式

$$s=\frac{1}{2^2}+\frac{3}{4^2}+\frac{5}{6^2}+\cdots+\frac{(2\times n-1)}{(2\times n)^2}$$

直到

$$\frac{(2\times n-1)}{(2\times n)^2}\le 10^{-3}$$

并把计算结果作为函数值返回。

例如，若形参 e 的值为 1e-3，函数的返回值为 2.985678。

请在程序的下划线处填入正确的内容并把下划线删除，使程序得出正确的结果。

注意：源程序存放在考生文件夹下的 BLANK1.C 中。不得增行或删行，也不得更改程序的结构！

试题程序如下：

```
#include <stdio.h>
double fun(double e)
{
    int i;
    double s,x;
    /**********found**********/
    s=1.0/4;i=①_____;
    x=1.0;
    while(x > e){
        /**********found**********/
        ②_____;
        /**********found**********/
        x=(2.0*i-1)/((③_____)*(2.0*i));
        s=s+x;
    }
    return s;
}
main()
{
    double e=1e-3;
    printf("\nThe result is: %f\n",fun(e));
}
```

【答案】

①1

②i++或++i 或 i+=1 或 i=i+1

③2.0*i

【解析】

填空 1：给 i 赋初值 1。

填空 2：当 x>e 时，i 进行自增运算。

填空 3：根据题中表达式形式可知分母为(2.0*i)(2.0*i)。

2．请根据以下各小题的要求设计 C 应用程序（包括界面和代码）。

请补充 fun 函数，该函数的功能是：计算 N×N 维矩阵元素的方差，结果由函数返回。维数在主函数中输入。

例如：

$$a=\begin{vmatrix}46 & 30 & 32\\ 40 & 6 & 17\\ 45 & 15 & 48\end{vmatrix}$$

的计算结果是 14.414。

求方差的公式为：

$$S=\sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(X_k-X'\right)^2}$$

其中

$$X' = \frac{1}{n}\sum_{k=1}^{n} X_k$$

注意：

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的横线上填入所编写的若干表达式或语句。

试题程序如下：

```
#include <stdio.h>
#include <stdlib.h>
#define N 20
/**********found**********/
double fun(①_____,int n)
{
   int i,j;
   double s=0.0;
   double f=0.0;
   double aver=0.0;
   double sd=0.0;
   for(i=0;i<n;i++)
      for(j=0;j<n;j++)
         s+=a[i][j];
   /**********found**********/
   aver=②_____;
   for(i=0;i<n;i++)
      for(j=0;j<n;j++)
         f+=(a[i][j]-aver)*(a[i][j]-aver);
   f/=(n*n);
   /**********found**********/
   sd=③_____;
   return sd;
}
main()
{
   int a[N][N];
   int n;
   int i,j;
   double s;
   printf("*****Input the dimension of array n******\n");
   scanf("%d",&n);
   printf("*****The array*****\n");
   for(i=0;i<n;i++)
   {
      for(j=0;j<n;j++)
      {
         a[i][j]=rand()%50;
         while (a[i][j]==0)
            a[i][j]=rand()%60;
         printf("%4d",a[i][j]);
      }
      printf("\n\n");
   }
   s=fun(a,n);
   printf("*****THE RESULT*****\n");
   printf("%4.3f\n",s);
}
```

【答案】

①int a[][N]

②s/(n*n)

③sqrt(f)

【解析】

根据题目中的方差公式，先要计算矩阵元素的平均值，再求矩阵元素与平均值差的平方的平均值，最后调用库函数求方差。

填空 1：根据函数中各变量的使用情况及实参的类型，这里应该是对二维数组 a 的定义，因为二维数组做形参，可以省略第一维的长度，但不能省略第二维的长度，所以此处应填 int a[][N]。

填空 2：这里求矩阵元素的平均值，注意元素的个数应该是 n*n。

填空 3：此处应该调用库函数 sqrt()求平方根。

## 三、改错题

### 1．请根据以下各小题的要求设计 C 应用程序（包括界面和代码）。

给定程序中函数 fun()的功能是：按顺序给 s 所指数组中的元素赋予从 2 开始的偶数，然后再按顺序对每五个元素求一个平均值，并将这些值依次存放在 w 所指的数组中。若 s 所指数组中元素的个数不是 5 的倍数，多余部分忽略不计。

例如，s 所指数组有 14 个元素，则只对前 10 个元素进行处理，不对最后的 4 个元素求平均值。

请改正程序中的错误，使它能得到正确结果。

注意：不要改动 main()函数，不得增行或删行，也不得更改程序的结构。

试题程序如下：

```
#include <stdio.h>
#define SIZE 20
int fun(double *s, double *w)
{
   int k,i;
   double sum;
   for(k=2,i=0;i<SIZE;i++)
   {
      s[i]=k;
      k+=2;
   }
   /*********found*********/
   sun=0.0;
   for(k=0,i=0;i<SIZE;i++)
   {
      sum+=s[i];
      /*********found*********/
      if(i+1%5==0)
      {
         w[k]=sum/5;
         sum=0;
         k++;
      }
   }
   return k;
}
main()
{
   double a[SIZE],b[SIZE/5];
   int i, k;
   k = fun(a,b);
   printf("The original  data:\n");
   for(i=0; i<SIZE; i++)
   {
      if(i%5==0)
         printf("\n");
      printf("%4.0f", a[i]);
   }
   printf("\n\nThe result :\n");
   for(i=0; i<k; i++)
      printf("%6.2f ",b[i]);
   printf("\n\n");
}
```

【答案】

（1）错误：sun=0.0

正确：sum=0.0

（2）错误：if(i+1%5==0)

正确：if((i+1)%5==0)

【解析】

错误 1：由题目程序的其他部分可以知道，此处变量名 sun 书写错误，应为 sum。

错误 2：由 C 语言的运算符优先级的定义顺序可以知道：由于%的优先级比+高，所以必须加上括号，因此改为：if((i+1)%5==0)。

2．给定程序 MODI1.C 中函数 fun 的功能是：将 s 所指字符串的正序和反序进行连接，形成一个新串放在 t 所指的数组中。

例如，当 s 所指字符串为：“ABCD”时，则 t 所指字符串中的内容应为：“ABCDDCBA”。请改正程序中的错误，使它能得出正确的结果。

注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构！

试题程序如下：

```
#include <stdio.h>
#include <string.h>
/*********found*********/
void fun(char s,char t)
{
   int i,d;
   d=strlen(s);
   for(i=0;i<d;i++)t[i]=s[i];
   for(i=0;i<d;i++)t[d+i]=s[d-1-i];
   /*********found*********/
   t[2*d-1]='\0';
}
main()
{
   char s[100],t[100];
   printf("\nPlease enter string S:");
   scanf("%s",s);
   fun(s,t);
   printf("\nThe result is:%s\n",t);
}
```

【答案】

（1）错误：void fun(char s,char t)

正确：void fun(char *s,char *t)

（2）错误：t[2*d-1]='\0';

正确：t[2*d]='\0';或 t[d+i]='\0';或 t[2*d]=0;或 t[d+i]=0;

【解析】

错误 1：根据 main 函数中 fun(s,t)可知，fun 函数的第一个参数为数组名称，也就是指针类型。

错误 2：数组 t 中字符个数是 s 的 2 倍，字符存储在 t[0]～t[2*d-1]，t[2*d]= '\0'；第 2 个 for 循环结束后，i=d，答案也可为 t[d+i]='\0'。

四、设计题

1．请根据以下要求设计 C 应用程序。

请编写函数 fun，它的功能是：统计各年龄段的人数。N 个年龄通过调用随机函数获得，并放在主函数的 age 数组中；要求函数把 0 至 9 岁年龄段的人数放在 d[0]中，把 10 至 19 岁年龄段的人数放在 d[1]中，把 20 至 29 岁年龄段的人数放在 d[2]中，其余依次类推，把 100 岁（含 100 岁）以上年龄的人数放在 d[10]中。结果在主函数中输出。

注意：部分源程序给出如下。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入所编写的若干语句。

试题程序如下：

```
#include <stdio.h>
#define N 50
#define M 11
void fun(int *a, int *b)
{

}
double rnd()
{
   static int t=29,c=217,m=1024,r=0;
   r= (r*t+c)%m;
   return ((double) r/m);
}
main ()
{
   int age[N],i,d[M];
   for(i=0; i< N; i++)
      age [i]= (int)(115*rnd ());
   printf ("The original data : \n");
   for(i=0; i< N; i++)
      printf((i+1)%10 == 0? "%4d\n":"%4d",age[i]);
   printf("\n\n");
   fun(age,d);
   for(i=0; i< 10; i++)
      printf("%4d---%4d :%4d\n", i*10,i*10+9,d[i]);
   printf("Over 100 : %4d\n",d[10]);
}
```

答：

```
void fun(int *a, int *b)
{
   int i,j;
   for (j=0; j<M;j++)
      b[j]=0;
   for (i=0; i<N;i++)
   {
      switch (a[i]/10)
      {
         case 0:b[0]++; break;
         case 1:b[1]++; break;
         case 2:b[2]++; break;
         case 3:b[3]++; break;
         case 4:b[4]++; break;
         case 5:b[5]++; break;
         case 6:b[6]++; break;
         case 7:b[7]++; break;
         case 8:b[8]++; break;
         case 9:b[9]++; break;
         default:b[10]++; break;
      }
   }
}
```

【解析】本题的关键是使用条件语句和选择语句统计各年龄段的人数。设计思路是：①使用循环语句对存储人数的形参数组 b 赋初值 0；②利用循环语句遍历所有人的年龄；③利用选择语句统计各年龄段的人数。

2．请编写一个函数，用来删除字符串中的所有空格。例如，输入 asdafaa z67，则输出为 asdafaaz67。注意：部分源程序给出如下。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

试题程序如下：

```
#include <stdio.h>
#include <ctype.h>
#include <conio.h>
void fun(char *str)
{

}
main()
{
   char str[81];
   char Msg[]="Input a string:";
   int n;
   printf(Msg);
   gets(str);
   puts(str);
   fun(str);
   printf("*** str:%s\n",str);
}
```

答：

```
void fun(char *str)
{
   int i=0;
   char *p=str;
   while(*p)/*遍历字符串*/
   {
      /*如果当前元素不为空格*/
      if(*p!=' ')
      {
         /*将当前元素保存到 str 中*/
         str[i]=*p;
         i++;
      }
      p++;
   }
   str[i]='\0';/*字符串最后加上结束标记符'\0'*/
}
```

【解析】题目要求删除空格，也就是重新保存空格以外的其他字符。通过循环删除字符串中的每一个空格，将非空格字符进行重新保存。

# 第 8 章　善于利用指针

## 一、选择题

1．以下叙述中错误的是（　　）。

A．当在程序的开头包含头文件 stdio.h 时，可以给指针变量赋 NULL

B．函数可以返回地址值

C．改变函数形参的值，不会改变对应实参的值

D．可以给指针变量赋一个整数作为地址值

【答案】D

【解析】A 项正确，NULL 是在头文件 stdio.h 中定义的符号常量；B 项正确，函数的返回值可以是地址，即指针；C 项正确，函数调用中形参值的变化不会传递给实参；D 项错误，不能将一个整数直接赋给指针变量作为地址，只能用取地址运算符“&”把该整型变量的地址赋值给该指针变量。答案选择 D 选项。

2．以下关于指针的叙述正确是（　　）。

A．所有类型的指针变量所占内存的大小是一样的

B．指针变量所占内存的大小与其类型有关，char 型指针变量只占 1 个字节，double 型指针变量占 8 个字节

C．指针变量可直接指向任何类型的变量，而不会出现编译或运行错误

D．指针变量既可以直接指向结构体，也可直接指向结构体中某个成员，而不会出现编译或运行错误

【答案】A

【解析】所有变量地址值所占字节都相同，故指针变量所占内存大小相同，A 项正确，B 项错误。只能把具有相同类型的变量地址，存放到指针变量中，结构体变量与某个成员类型不同，不能用同样的指针指向它们，C、D 两项错误。答案选择 A 选项。

3．关于地址和指针，以下说法正确的是（　　）。

A．通过强制类型转换可以将一种类型的指针变量赋值给另一种类型的指针变量

B．可以取一个常数的地址赋值给同类型的指针变量

C．可以取一个表达式的地址赋值给同类型的指针变量

D．可以取一个指针变量的地址赋值给基类型相同的指针变量

【答案】A

【解析】常数的地址存储在内存的常量区，常量区存储的都是常量，值都是不可修改的，所以直接取常量的地址赋给指针变量没有任何意义，C 语言也不允许这样做，编译会出错，B 项错误；表达式的值存储在临时变量中，内存中存在专门用来存储临时变量的区域，对这块地址进行操作也是没有意义的，C 语言不允许这样做，编译会出错，C 项错误；可以取一个指针变量的地址，但是指针变量的地址属于指针，只能赋值给指针类型的指针变量，D 项错误。答案选择 A 选项。

4．以下关于指针的叙述，错误的是（　　）。

A．两个基类型相同的指针变量不能指向同一个对象

B．可以通过对指针变量自增、自减来移动指针

C．只有两个类型相同的指针才能进行减运算

D．一个指针变量可以通过不同的方式获得一个确定的地址值

【答案】A

【解析】只要两个指针变量基类型相同，可以指向同一个对象，答案选择 A 选项。

5．以下叙述中错误的是（　　）。

A．基类型不同的指针可以直接相互赋值

B．函数可以通过指针形参向所指单元传回数据

C．字符型指针可以指向一个字符串

D．一般情况下，指针的运用可使程序代码效率更高

【答案】A

【解析】把一个指针变量的值赋给另一个指针变量，但一定要确保这两个指针变量的基类型是相同的。答案

选择 A 选项。

6．有以下程序：

```
#include <stdio.h>
main()
{
   char *p1 = 0;
   int *p2 = 0;
   float *p3 = 0;
   printf("%d,%d,%d\n", sizeof(p1), sizeof(p2), sizeof(p3));
}
```

程序运行后的输出结果是（　　）。

A．1,4,8

B．4,4,4

C．1,2,4

D．1,1,4

【答案】B

【解析】sizeof 函数用来获取类型或数据对象的长度，也即是一个这种数据类型的变量在内存中所占字节数。由于一个变量的地址也是一个值，因此就可以把这个地址值存放到另一个变量里保存。这种专门用来存放变量地址的变量，称为“指针变量”。所有类型的指针变量都是地址，所占字节数均为 4，答案选择 B 选项。

7．若有定义语句：

double a,*p=&a;

以下叙述中错误的是（　　）。

A．定义语句中的 p 只能存放 double 类型变量的地址

B．定义语句中的*号是一个说明符

C．定义语句中的*号是一个间址运算符

D．定义语句中*p=&a 把变量 a 的地址作为初值赋给指针变量 p

【答案】C

【解析】C 项错误，只有取指针变量的值时，星号*才是间址运算符，引用指针指向的存储单元。A 项正确，p 是 double 类型的指针，只能存放 double 类型的地址；B 项正确，定义指针变量时，星号*是一个说明符，用来说明该变量是指针变量；D 项正确，“&”是求地址运算符，*p=&a 用来求出 double 变量 a 的地址赋给指针变量 p，而使 p 指向 a。答案选择 C 选项。

8．设有定义：

int a,b[10],*c=NULL,*p;

则以下语句错误的是（　　）。

A．p=a;

B．p=b;

C．p=c;

D．p=&b[0];

【答案】A

【解析】p 为指针变量，存放变量的地址。对指针变量赋值，值必须是地址值。a 为整型变量不是地址值，不能赋值给 p，A 项错误，应改为 p=&a。数组名 b 为数组首地址，可以赋值给 p，B 项正确。c 为指针变量，初始化为 NULL，与 p 均为整型指针，可以将其赋值给 p，C 项正确。&为取地址运算符，&b[0]为数组元素 b[0]的地址，可以赋值给 p，D 项正确。答案选择 A 选项。

9．已定义以下函数

```
int fun(int*p)
{
   return *p;
}
```

fun 函数返回值是（　　）。

A. 一个整数

B. 不确定的值

C. 形参 p 中存放的值

D. 形参 p 的地址值

【答案】A

【解析】题目中 fun()函数的返回表达式是*p，是形参 p 所指内容的值，因为 p 为 int 型指针，所以该值是一个整数。答案选择 A 选项。

10. 设有定义：

int x=0,*p;

紧接着的赋值语句正确的是（　　）。

A. *p=x;

B. *p=NULL;

C. p=x;

D. p=NULL;

【答案】D

【解析】指针赋值的正确写法：①p=&x，表示 p 指向 x 的存储单元；②p=NULL，表示 p 是空指针。③*p=x，表示将 p 指向的内容赋值为 x，但前提是 p 已进行了初始化。答案选择 D 选项。

11. 设已有定义：

float x;

则以下对指针变量 p 进行定义且赋初值的语句中正确的是（　　）。

A. float *p=&x;

B. int *p=(float)x;

C. float p=&x;

D. float *p=1024;

【答案】A

【解析】A 项正确，p 是 float 类型的指针，指向 x 的存储单元；B 项错误，p 是 int 类型的指针，只能指向 int 变量的存储单元；C 项错误，p 的 float 类型的变量，&x 是取变量 x 的地址，不能把地址赋值给 float 类型变量；D 项错误，p 是指向 float 类型的指针，不能使用常整型赋值。答案选择 A 选项。

12. 设变量 p 是指针变量，语句 p=NULL;是给指针变量赋 NULL 值，它等价于（　　）。

A. p="";

B. p='0';

C. p=0;

D. p='';

【答案】C

【解析】NULL 的 ASCII 码值为 0，p=NULL 等价于 p='\0';或 p=0。答案选择 C 选项。

13. 以下程序中关于指针输入格式正确的是（　　）。

A. int *p;scanf("%d",&p);

B. int * p;scanf("%d",p);

C. int k,* p=&k;scanf("%d",p);

D. int k,*p;*p=&k;scanf("%d",&p);

【答案】C

【解析】A 项错误，指针 p 未初始化，且 scanf 中 p 的格式不对；B 项，指针 p 未初始化；C 项正确，首先将 k 的地址赋值给 p，然后通过 p 从键盘读入数据给 k 赋值；D 项错误，scanf 中 p 的格式不对。答案选择 C 选项。

14．若有定义语句：double x,y,*px,*py;执行了 px=&x;py=&y;之后，正确的输入语句是（　　）。

A．scanf("%f%f",x,y);

B．scanf("%f%f"&x,&y);

C．scanf("%1f%1f",px,py);

D．scanf("%1f%1f",x,y);

【答案】C

【解析】输入函数 scanf 的标准格式是：scanf(格式控制,地址列表);，AD 两项中地址列表格式不正确，应为&x，&y。格式控制和地址列表间应该用逗号隔开，B 项也错误。%f 用来输入 float 类型变量，%1f 用来输入 double 类型变量，%1e 表示用科学计数法输入 double。答案选择 C 选项。

15．有以下程序

```
#include <stdio.h>
main()
{
   int n,*p=NULL;
   *p=&n;
   printf("Input n:");
   scanf("%d",&p);
   printf("output n:");
   printf("%d\n",p);
}
```

该程序试图通过指针 p 为变量 n 读入数据并输出，但程序有多处错误，以下语句正确的是（　　）。

A．int n,*p=NULL;

B．*p=&n;

C．scanf("%d",&p);

D．printf("%d\n",p);

【答案】A

【解析】B 项的正确写法应为“p=&n;”，将变量 n 的地址赋给指针 p；C 项的正确写法应为“scanf("%d",p);”；D 项的正确写法应为“printf("%d\n",*p);”。答案选择 A 选项。

16．有以下程序：

```
#include <stdio.h>
main()
{
   int *p,x=100;
   p=&x;
   x=*p+10;
   printf("%d\n",x);
}
```

程序运行后的输出结果是（　　）。

A．110

B．120

C．100

D．90

【答案】A

【解析】程序执行过程为：定义指针 p，指向变量 x，p 的值即为 x 的地址，*p 就表示该地址处存放的值，x=(*p)+10=110，输出 110。答案选择 A 选项。

17．有以下程序

```
#include <stdio.h>
main()
{
   int a=1,b=3,c=5;
   int *p1=&a,*p2=&b,*p=&c;
   *p=*p1*(*p2);
   printf("%d\n",c);
}
```

执行后的输出结果是（　　）。

A．4

B．2

C．1

D．3

【答案】D

【解析】本题中*p=*p1*(*p2)=a*b=3，也就是将指针 p 所指存储空间的值改为 3，即 c 改为 3。答案选择 D 选项。

18．以下叙述中正确的是（　　）。

A．在对指针进行加、减算术运算时，数字 1 表示 1 个存储单元的长度

B．如果 p 是指针变量，则*p 表示变量 p 的地址值

C．如果 p 是指针变量，则&p 是不合法的表达式

D．如果 p 是指针变量，则*p+1 和*(p+1)的效果是一样的

【答案】A

【解析】A 项正确，在对指针进行加、减运算时，数字“1”不再代表十进制整数“1”，而是指 1 个存储单元长度。B 项错误，*p 表示 p 的值；C 项错误，&p 表示变量 p 的地址；D 项错误，*p+1 是先取 p 的值，然后执行+1 操作，*(p+1)是先对指针移动 1 个存储空间，然后取值。答案选择 A 选项。

19．以下叙述中正确的是（　　）。

A．函数的形参类型不能是指针类型

B．函数的类型不能是指针类型

C．设有指针变量为 double *p，则 p+1 将指针 p 移动 8 个字节

D．基类型不同的指针变量可以相互混用

【答案】C

【解析】C 项正确，一般情况下，double 类型的变量长度为 8 个字节，对指针进行加、减运算时，数字“1”指 1 个存储单元长度。A 项错误，函数的形参可以是数值类型，也可以是指针类型；B 项错误，C 语言中有指向函数的指针，称为函数指针；D 项错误，int 类型的指针只能指向 int，不能指向 double，基类型不同的指针变量不能混用。答案选择 C 选项。

20．若有定义语句 int year=2009,*p=&year;，以下不能使用变量 year 中的值增至 2010 的语句是（　　）。

A．*p+=1

B．(*p)++;

C．++(*P);

D．* p++;

【答案】D

【解析】A 项，p 所指内容执行增 1 操作；B 项，先取 P 指针所指的内容，再执行后++运算；C 项，先取 P 指针所指内容，再执行前++运算；D 项，实际上是 p 指针执行++操作，而后取值。答案选择 D 选项。

21．有以下程序：

```
#include <stdio.h>
main()
{
   int a[10]={1,3,5,7,11,13,17},*p=a;
   printf("%d,",*(p++));
   printf("%d\n",*(++p));
}
```

程序运行后的输出结果是（　　）。

A．3,7

B．3,5

C．1,5

D．1,3

【答案】C

【解析】程序执行过程：指针 p 指向数组第一个元素；*(p++)先取 p，输出 p 指向的元素 1，之后 p 移动 1 个存储空间，指向数组第二个元素；*(++p)，指针 p 移动 1 个存储空间指向数组第三个元素，之后输出所指元素 5。答案选择 C 选项。

22．有如下程序：

```
#include <stdio.h>
main()
{
   int a=0,*ptr;
   ptr=&a;
   *ptr=3;
   a=(*ptr)++;
   printf("%d,%d\n",a,*ptr);
}
```

程序运行后的输出结果是（　　）。

A．4,4

B．0,1

C．1,4

D．0,4

【答案】A

【解析】程序执行过程为：定义整型变量 a=0 与指针 ptr，使指针指向变量 a。对指针指向的变量进行赋值 a=3，将指针所指向变量加 1，并赋值给 a=4，此时指针依然指向变量 a，输出 a 与指针指向的变量值：4，4，答案选择 A 选项。

23．有以下程序：

```
#include<stdio.h>
main()
{
   int m=1,n=2,*p=&m,*q=&n,*r;
   r=p;
   p=q;
   q=r;
   printf("%d,%d,%d,%d\n",m,n,*p,*q);
}
```

程序运行后的输出结果是（　　）。

A. 1,2,1,2

B. 1,2,2,1

C. 2,1,2,1

D. 2,1,1,2

【答案】B

【解析】m 和 n 的值不变，分别是 1，2；指针*p 和*q 交换了指向的位置，即*p=&n，*q=&m，分别为 2，1。答案选择 B 选项。

24. 有以下程序

```
#include <stdio.h>
main()
{
   int c[6]={10,20,30,40,50,60},*p,*s;
   p=c;
   s=&c[5];
   printf("%d\n",s-p);
}
```

程序运行后的输出结果是（　　）。

A. 5

B. 50

C. 6

D. 60

【答案】A

【解析】指针 p 指向 c，指针 s 指向 c[5]，s-p 就是 c 的地址与 c[5]的地址之差（以 int 的存储空间为单位），s 和 p 都是 int 型变量，s=p+5，所以 s-p=5。答案选择 A 选项。

25. 有以下程序

```
#include <stdio.h>
main()
{
   int a[5]={2,4,6,8,10},*p,**k;
   p = a;
   k = &p;
   printf("%d",*(p++));
   printf("%d\n",**k);
}
```

程序运行后的输出结果是（　　）。

A．24
B．44
C．22
D．46
【答案】A
【解析】p 是 int 类型的指针，指向数组 a 的首元素；k 的 int *类型的指针，指向 p 的地址，即 k 指针和 p 指针都指向元素 2 的位置。*(p++)，先使用 p 的值，输出该地址对应的存储空间的值，即输出 2，再执行 p++，p 指向 a 中第二个元素；k 指针始终指向 p 的地址，那么*k 的值就是 p 本身的值，即 a[1]的地址，**k 的值就是 a[1]，输出 4。答案选择 A 选项。

26．以下程序的功能是：通过调用 calc 函数，把所求得的两数之和值放入变量 add 中，并在主函数中输出。

```
#include <stdio.h>
void calc(float x,float y,float *sum)
{
    _______ = x+y;
}
main ()
{
    float x,y,add;
    scanf("%f%f",&x,&y);
    calc(x,y,&add);
    printf("x+y=%f\n",add);
}
```

calc 函数中下划线处应填入的是（　　）。
A．*sum
B．sum
C．&sum
D．add
【答案】A
【解析】程序的执行过程为：从键盘读入两个 float 类型数据，分别赋给 x，y，调用函数 calc 将 x 与 y 的值与 add 变量地址传入函数，地址赋给指针 sum，函数体中将两数之和放入指针指向的地址，指针正确的引用形式为：*sum，这表示变量，可以被赋值。所以横线处填写*sum。答案选择 A 选项。

27．有以下程序：

```
#include <stdio.h>
void fun(int x,int y,int *z)
{
    *z=y-x;
}
main()
{
    int a,b,c;
    fun(10,5,&a);
    fun(7,a,&b);
    fun(a,b,&c);
    printf("%d,%d,%d\n",a,b,c);
}
```

程序运行后的输出结果是（　　）。

A．5,2,3

B．-5,-12,-7

C．-5,-12,-17

D．5,-2,-7

【答案】B

【解析】程序执行过程为：调用函数 fun(10,5,&a)，将变量 a 地址传入函数，a=5-10=-5；调用函数 fun(7,a,&b)，将变量 b 地址传入函数，b=-5-7=-12；调用函数 fun(a,b,&c)，将变量 c 地址传入函数，c=-12-(-5)=-7。输出-5，-12，-7。答案选择 B 选项。

28．有如下程序：

```
#include <stdio.h>
int change(int *data)
{
   return (*data)++;
}
main()
{
   int data=123;
   change(&data);
   printf("%d,",data);
   data = change(&data);
   printf("%d,",data);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．124,124,

B．123,124,

C．124,123,

D．123,123,

【答案】A

【解析】本题程序执行过程为：调用 change 函数，将变量 data 地址传入函数，返回当前指针指向的变量值 123，之后此变量加一，data=124。输出 data=124。再次调用函数，返回当前指针指向的变量值 124，然后此地址内存中变量加一，data=125，然后将返回值 124 赋给变量 data=124，最后输出 data=124。答案选择 A 选项。

29．有如下程序：

```
#include <stdio.h>
int convert(int *data)
{
   return (*data)++;
}
main()
{
   int data = 56;
   convert(&data);
   printf("%d,",data);
   data = convert(&data);
   printf("%d,\n",data);
}
```

程序运行后的输出结果是（　　）。

A．56,57,

B．57,58,

C．57,57,

D．55,57,

【答案】C

【解析】convert 函数定义了一个形参：指针变量*data。函数体中将指针变量*data 对应地址的值进行加 1 处理，再返回加 1 前 data 的值。main 函数中，定义了变量 data 并赋初值 56，调用 convert 函数，改变变量 data 的值，此时 data=57；程序执行 data=convert(&data)，函数 convert 返回的是执行前 data 的值，所以 data=57。答案选择 C 选项。

30．有以下程序：

```
#include <stdio.h>
int k=5;
void f(int *s)
{
   s=&k;
   *s=7;
}
main()
{
   int m=3;
   f(&m);
   printf("%d,%d\n",m,k);
}
```

程序运行后的输出结果是（　　）。

A．3,5

B．7,7

C．5,7

D．3,7

【答案】D

【解析】函数 f 的功能是定义一个整型的指针变量 s，指向全局变量 k，然后修改 s 指向地址中的值为 7，因此 f 函数只是修改全局变量 k 的值为 7，与 main 函数中临时变量 m 无关，因此最后输出为 3,7。答案选择 D 选项。

31．有以下程序：

```
#include <stdio.h>
int k=5;
void f(int *s)
{
   s=&k;
}
main()
{
   int m=3,*p=&m;
   f(p);
   printf("%d,%d\n",m,*p);
}
```

程序的运行结果是（　　）。

A．3,3
B．5,5
C．3,5
D．5,3
【答案】A
【解析】考查函数传参。C 语言中，数据只能从实参单向传递给形参，指针作为函数参数时，形参仍然作为实参的副本被赋值，形参指针变量的指向改变不能影响实参指针变量的指向。结果不变，答案选择 A 选项。

32．以下选项中，不能对主函数中变量 i 和 j 的值进行交换的程序是（　　）。
A.

```
#include <stdio.h>
void swap(int *p,int *q)
{
    int *t;
    *t=*p;
    *p=*q;
    *q=*t;
}
main()
{
    int i=10,j=20,*a=&i,*b=&j;
    swap(a,b);
    printf("i=%d j=%d\n",i,j);
}
```

B.

```
#include <stdio.h>
void swap(int *p,int *q)
{
    int t;
    t=*p;
    *p=*q;
    *q=t;
}
main()
{
    int i=10,j=20,*a=&i,*b=&j;
    swap(a,b);
    printf("i=%d j=%d\n",i,j);
}
```

C.

```
#include <stdio.h>
#include <stdlib.h>
void swap(int *p,int *q)
{
   int *t;
   t=(int *) malloc(sizeof(int));
   *t=*p;
   *p=*q;
   *q=*t;
   free(t);
}
main()
{
   int i=10,j=20;
   swap(&i,&j);
   printf("i=%d j=%d\n",i,j);
}
```

D.

```
#include <stdio.h>
void swap(int *p,int *q)
{
   int t;
   t=*p;
   *p=*q;
   *q=t;
}
main()
{
   int i=10,j=20,*x=&i,*y=&j;
   swap(x,y);
   printf("i=%d j=%d\n",i,j);
}
```

【答案】A

【解析】A 项，定义了一个临时指针 t，实现两个指针地址的交换，而传入的参数是两个变量 i 和 j 的地址，但是函数内部交换的是地址值，并没有交换主函数中变量 i 与 j 的值，故选择 A 选项；B 项，调用函数传入的是 i 与 j 地址，函数体内交换的是地址内元素，临时变量 t 为整型变量，能实现 i 与 j 值交换；C 项，调用函数传入的是 i 与 j 地址，函数体内交换的是地址内元素，临时变量 t 为整型指针，且已正确开辟内存，能实现 i 与 j 值交换；D 项与 B 项相同，能实现 i 与 j 值交换。答案选择 A 选项。

33．有以下程序：

```
#include <stdio.h>
void fun(int *a,int *b)
{
   int *c;
   c=a;
   a=b;
   b=c;
}
main()
{
   int x=3,y=5,*p=&x,*q=&y;
   fun(p,q);
   printf("%d,%d,",*p,*q);
   fun(&x,&y);
   printf("%d,%d\n",*p,*q);
}
```

程序运行后的输出结果是（　　）。

A. 3,5,5,3

B. 3,5,3,5

C. 5,3,3,5

D. 5,3,5,3

【答案】B

【解析】从程序中可以看出 fun 函数的作用是将两个指针变量所指向的位置互换。在主函数中，两次调用 fun 函数，第一次调用传递的实参是指向数值的指针，在 C 语言中实参变量与形参变量之间的数据传递方式是单向的“值传递”方式，调用 fun 函数不可能改变实参指针变量的值，只可以改变实参指针变量所指变量的值。那么经 fun 函数处理后，不改变指针变量的值，其输出的结果应该是 3,5,；第二次调用传递的实参是存放变量的地址，与第一次调用一样，地址的改变不能被允许，因此其输出结果也是 3,5。答案选择 B 选项。

34．有以下程序：

```
#include <stdio.h>
int *f(int *s,int *t)
{
   if(*s < *t)*s=*t;
   return s;
}
main()
{
   int i=3,j=5,*p=&i,*q=&j,*r;
   r=f(p,q);
   printf("%d,%d,%d,%d,%d\n",i,j,*p,*q,*r);
}
```

程序的运行结果是（　　）。

A. 5,5,5,5,5

B. 3,5,5,5,5

C. 5,3,3,3,5

D. 3,5,3,5,5

【答案】A

【解析】程序执行过程为：p 指向 i 的地址，q 指向 j 的地址，调用函数 f，比较 p 指向地址存储的元素值 3

与 q 指向地址存储的元素值 5 的大小，if 语句成立时将 q 指向地址存储的元素值 5 赋给 p 指向地址存储的元素，即此时 p 指向地址存储的元素值 i=5，返回值指针 r 指向 i，最后输出 5,5,5,5,5，答案选择 A 选项。

35．有以下程序：

```
#include <stdio.h>
void f(int *p,int *q);
main()
{
   int m=1,n=2,*r=&m;
   f(r,&n);
   printf("%d,%d",m,n);
}
void f(int *p,int *q)
{
   p=p+1;
   *q=*q+1;
}
```

程序运行后的输出结果是（　　）。

A．1,3

B．2,3

C．1,4

D．1,2

【答案】A

【解析】语句 p=p+1;只改变指针 p 的地址，与 p 的内容无关，所以 m 值没有改变，而语句*q=*q+1;是改变该指针所指地址中的内容，所以 n 变为 3，m 不变，仍为 1，答案选择 A 选项。

36．若有以下程序

```
#include <stdio.h>
void sp(int *a)
{
   int b=2;
   a=&b;
   *a=*a*2;
   printf("%d,",*a);
}
main()
{
   int k=3,*p=&k;
   sp(p);
   printf("%d,%d\n",k,*p);
}
```

则程序的输出结果是（　　）。

A．4,3,4

B．4,3,3

C．6,3,6

D．6,6,6

【答案】B

【解析】函数 sp(p)调用前，p 指向 k；调用后，a=&b 语句表示指针 a 指向变量 b 的地址，*a=*a*2 等价于

b=b*2，最后在 sp 函数内部输出 4；sp 函数没有改变指针 p 指向的内存单元，k 和*p 仍然是 3。答案选择 B 选项。

37．有以下程序

```
#include <stdio.h>
void fun(int x,int y,int *c,int *d)
{
   *c=x+y;
   *d=x-y;
}
main()
{
   int a=4,b=3,c=0,d=0;
   fun(a,b,&c,&d);
   printf("%d %d\n",c,d);
}
```

程序的输出结果是（　　）。

A．7 1

B．4 3

C．3 4

D．0 0

【答案】A

【解析】main 函数中变量 c 和 d 以传地址的方式传递参数，形参的改变会导致实参的改变。答案选择 A 选项。

38．有以下程序

```
#include <stdio.h>
void fun(int *p,int *q)
{
   int t;
   t = *p;
   *p = *q;
   *q = t;
   *q = *p;
}
main()
{
   int a = 0,b = 9;
   fun(&a,&b);
   printf("%d %d\n",a,b);
}
```

程序的输出结果是（　　）。

A．0 9

B．0 0

C．9 0

D．9 9

【答案】D

【解析】main 函数中变量 a 和 b 以传地址的方式传递参数，形参的改变会导致实参的改变。fun 函数的作用是先交换两个指针的指向的值，即 p 指针指向的值是 9，q 指针指向的值是 0，然后将 p 指针指向的值赋给 q 指

针，q 指针指向的值也成了 9，所以 a 和 b 的值均为 9，答案选择 D 选项。

39．有以下程序

```
#include <stdio.h>
void fun1(char *p)
{
   char *q;
   q=p;
   while(*q!='\0')
   {
      (*q)++;
      q++;
   }
}
main()
{
   char a[]={"Program"},*p;
   p=&a[3];
   fun1(p);
   printf("%s\n",a);
}
```

程序执行后的输出结果是（　　）。

A．Prohsbn

B．Prphsbn

C．Progsbn

D．Program

【答案】A

【解析】函数 fun1 功能是将字符串中元素加 1，main 函数将数组第 4 个元素的指针传进 fun1 函数中，所以字符'g'、'r'、'a'、'm'都变为 ASCII 表上相邻的下一元素，答案选择 A 选项。

40．有以下程序：

```
#include <stdio.h>
void fun(char *a,char *b)
{
   while(*a=='*')a++;
   while(*b=*a)
   {
      b++;
      a++;
   }
}
main()
{
   char *s="*****a*b****",t[80];
   fun(s,t);
   puts(t);
}
```

程序的运行结果是（　　）。

A. *****a*b

B. a*b

C. a*b****

D. ab

【答案】C

【解析】注意两个 while，第一个是判断相等“==”，第二个是赋值“=”，第一个循环终止的条件是指针指向的字符不为'*'，那么就会在遇到字符'a'时停止，第二个循环把 a 指针所指的内容依次赋给了 b 指针指向的内容，直到字符串结束时终止。答案选择 C 选项。

41．有以下程序

```
#include<stdio.h>
void fun(char *c,int d)
{
   *c=*c+1;
   d=d+1;
   printf("%c,%c,",*c,d);
}
main()
{
   char b='a',a='A';
   fun(&b,a);
   printf("%c,%c\n",b,a);
}
```

程序运行后的输出结果是（　　）。

A. b,B,b,A

B. b,B,B,A

C. a,B,B,a

D. a,B,a,B

【答案】A

【解析】fun()函数中的两个局部变量 c 和 d，c 是一个字符指针变量，在程序中取出指针所指内存单元的值进行修改，对其进行的修改影响调用函数中对应的变量的值；而 d 是一个整型变量，在程序中进行的修改是局部的，不影响其他函数。答案选择 A 选项。

42．有以下程序

```
#include <stdio.h>
#include <string.h>
void fun(char *w,int m)
{
   char s,*p1,*p2;
   p1=w;
   p2=w+m-1;
   while(p1<p2)
   {
      s=*p1;
      *p1=*p2;
      *p2=s;
      p1++;
      p2--;
   }
}
main()
{
   char a[] = "123456";
   fun(a,strlen(a));
   puts(a);
}
```

程序运行后的输出结果是（　　）。

A. 654321

B. 116611

C. 161616

D. 123456

【答案】A

【解析】函数 fun 的功能是对字符串 w 中的元素按从大到小的顺序进行排序。其中，字符变量 s 为中间变量，指针 p1 和 p2 的初始值分别为数组元素的首地址和尾地址。若下标小的元素小于下标大的元素，则互换位置。答案选择 A 选项。

43. 有以下程序

```
#include<stdio.h>
void fun(char *t,char *s)
{
   while(*t!=0)t++;
   while((*t++=*s++)!=0);
}
main()
{
   char ss[10]="acc",aa[10]="bbxxyy";
   fun(ss,aa);
   printf("%s,%s\n",ss,aa);
}
```

程序的运行结果是（　　）。

A. accbbxxyy,bbxxyy

B. acc,bbxxyy

C．accxxyy,bbxxyy

D．accxyy,bbxxyy

【答案】A

【解析】函数 fun 的功能是将第二个字符串链接到第一个字符串的末尾，第二个字符串不变。答案选择 A 选项。

44．有以下程序

```
#include<stdio.h>
void swap(char *x,char *y)
{
   char t;
   t=*x;
   *x=*y;
   *y=t;
}
main()
{
   char *s1,*s2;
   char a[]="abc";
   char b[]="123";
   s1=a;
   s2=b;
   swap(s1,s2);
   printf("%s,%s\n",s1,s2);
}
```

程序执行后的输出结果是（　　）。

A．1bc,a23

B．abc,123

C．123,abc

D．321,cba

【答案】A

【解析】字符串是一个特殊的数组，按照数组的规则，s1 应该指向的是数组的首地址，即"abc"的第一个字符的地址。s2 指向的是"abc"的第一个字符的地址。调用 swap 函数之后交换的是两个字符串的第一个字符'a'和'1'的内容，答案选择 A 选项。

二、填空题

给定程序通过定义并赋初值的方式，利用结构体变量存储了一名学生的学号、姓名和 3 门课的成绩。函数 fun 的功能是将该学生的各科成绩都乘以一个系数 a。

请在程序的下画线处填入正确的内容并把下画线删除，使程序得出正确的结果。

注意：部分源程序给出如下。不得增行或删行，也不得更改程序的结构！

试题程序如下：

```
#include <stdio.h>
#include <string.h>
typedef struct
{
   int num;
   char name[9];
   float score[3];
} STU;
void show(STU tt)
{
   int i;
   printf("%d %s:",tt.num,tt.name);
   for(i=0; i<3; i++) printf("%5.1f",tt.score[i]);
   printf("\n");
}
/*********found*********/
void modify(①_____ *ss,float a)
{
   int i;
   for(i=0; i<3; i++)
      /*********found*********/
      ss->②_____*=a;
}
main()
{
   STU std={1,"Zhang",76.5,78.0,82.0};
   float a;
   printf("\nThe original number and name and scores:\n");
   show(std);
   printf("\nInput a number:");
   scanf("%f",&a);
   /*********found*********/
   modify(③_____,a);
   printf("\nA result of modifying:\n");
   show(std);
}
```

【答案】

①STU

②score[i]

③&std

【解析】

填空 1：形参 ss 是一个结构型指针变量，应填 STU。

填空 2：该学生的各科成绩都乘以一个系数 a，应填 score[i]。

填空 3：函数的调用，由于函数定义时使用的指针结构型变量，应填&std。

## 三、改错题

### 1．请根据以下各小题的要求设计 C 应用程序（包括界面和代码）。

下列给定的程序中，函数 fun 的功能是：把主函数中输入的 3 个数，最大的放在 a 中，最小的放在 c 中。例如，输入的数为 55 12 34，输出结果应当是：a=55.0，b=34.0，c=12.0。

请改正程序中的错误，使它能得到正确结果。
注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构。
试题程序如下：

```
#include <stdio.h>
#include <stdlib.h>
void fun(float *p,float *q,float *s)
{
   float *k;
   k = (float *)malloc(sizeof(float));
   if(*p<*q)
   {
      /*********found*********/
      k=*p;*p=*q;*q=k;
   }
   /*********found*********/
   if(*s<*p)
   {
      /*********found*********/
      k=*s; *s=*p; *p=k;
   }
   if(*q<*s)
   {
      /*********found*********/
      k=*q; *q=*s; *s=k;
   }
   free(k);
}
main()
{
   float a,b,c;
   printf("Input a b c: ");
   scanf("%f%f%f",&a,&b,&c);
   printf("a = %4.1f, b = %4.1f, c = %4.1f\n\n",a,b,c);
   fun(&a,&b,&c);
   printf("a = %4.1f, b = %4.1f, c = %4.1f\n\n",a,b,c);
}
```

【答案】
（1）错误：{k=*p;*p=*q;*q=k;}
正确：{*k=*p;* p=*q;*q=*k;}
（2）错误：if(*s<*p)
正确：if(*s>*p)
（3）错误：{k=*s;*s=*p;*p=k;}
正确：{*k=*8;*s=*p;*p=*k;}
（4）错误：{k=*q;*q=*s;*s=k;}
正确：{*k=*q;*q=*s;*s=*k;}
【解析】
错误 1、3、4：函数定义了一个 int 型指针变量 k 作为交换的辅助变量，此时不能直接用 k，应该用指针指向的变量*k。所以第 1、3、4 处错误都应把 k 改成*k。
错误 2：*p 存储最大值，*s 存储最小值，所以应该把 q 和 s 所指向的值与*p 进行比较，如果比*p 大则与*p

交换，所以 if(*s<*p)应改为 if(*s>*p)。

2．下列给定程序中，函数 fun 的功能是：从 s 所指字符串中，找出 t 所指字符串的个数作为函数值返回。例如，当 s 所指字符串中的内容为“abcdabfab”，t 所指字符串的内容为“ab”，则函数返回整数 3。

请改正程序中的错误，使它能得出正确的结果。

注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构！

试题程序如下：

```
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <string.h>
int fun(char *s, char *t)
{
   int n;
   char *p,*r;
   n=0;
   while(*s)
   {
      p=s;
      r=t;
      while(*r)
         /**********found**********/
         if(*r==*p){r++; p++}
         else break;
      /**********found**********/
      if(r=='\0')
         n++;
      s++;
   }
   return n;
}
void main()
{
   char s[100],t[100]; int m;
   system("CLS");
   printf("\nPlease enter strings:");
   scanf("%s",s);
   printf("\nPlease enter substrings:");
   scanf("%s",t);
   m=fun(s,t);
   printf("\nThe result is:m=%d\n", m);
}
```

【答案】

（1）错误：if(*r==*p){r++;p++}

正确：if(*r==*p){r++;p++;}

（2）错误：if(r=='\0')

正确：if（*r=='\0')

【解析】

错误 1：在经过“if”判断后执行后面括号内的语句时，每条语句应以“;”做结尾，“p++”后面没有分号即

是错误。

错误 2：该题目中定义*r 为指针变量，r 为指针名称，对其所指内容进行判断时应加“*”。

四、设计题

1．请编写一个函数 fun，它的功能是：将一个表示正整数的数字字符串转换为一个整数（不得调用 C 语言提供的将字符串转换为整数的函数）。例如，若输入字符串“1234”，则函数把它转换为整数值 1234。函数 fun 中给出的语句仅供参考。

注意：部分源程序存在文件 PROG1.C 中。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入需要编写的若干语句。

试题程序如下：

```
#include <stdio.h>
#include <string.h>
long fun (char *p)
{
   int i,len; /* len 为串长*/
   long x=0;
   len=strlen(p);
   /*以下完成数字字符串转换为数字，注意字符'0'不是数字 0*/


   return x;
}
void main()
{
   char s[6];
   long n;
   printf("Enter a tring:\n");
   gets(s);
   n = fun(s);
   printf("%ld\n",n);
}
```

答：

```
int flag=1;
i=0;
if(*p=='-')/*p 表示负数，置 flag 值为-1*/
{
   p++;
   i++;
   flag=-1;
}
else if(*p=='+')/*p 表示正数，置 flag 值为 1*/
{
   p++;
   i++;
}
for(;i<len;i++)
{
   x=x*10+*p-'0';/*将字符串转换成相应的整数*/
   p++;
}
x = x*flag;
```

【解析】if()语句的作用是判断该字符串应当为正数还是负数。注意：*p 是一个字符（如'9'、'4'），并不是一个数，要将其转成相应的数字需令其减去'0'（不是'\0'），即*p-'0'就得到*p 这个字符的相应数字，如'0'-'0'=0、'8'-'0'=8 等。

2．规定输入的字符串中只包含字母和*号。请编写函数 fun，其功能是：使字符串中尾部的*号不多于 n 个若多于 n 个，则删除多余的*号；若少于或等于 n 个，则不做任何操作，字符串中间和前面的*号不删除。例如，字符串中的内容为“****A*BC*DEF*G*******”，若 n 的值为 4，删除后，字符串中的内容应；“****A*BC*DEF*G****”；若 n 的值为 7，则字符串中的内容仍为“****A*BC*DEF*G*******”。n 的值在函数中输入。编写函数时，不得使用 C 语言提供的字符串函数。

注意：部分源程序给出如下。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

试题程序如下：

```
#include <stdio.h>
void fun(char *a,int n)
{

}
main()
{
   char s[81];
   int n;
   printf("Enter a string:\n");
   gets(s);
   printf("Enter n:");
   scanf("%d",&n);
   fun(s,n);
   printf("The string after deleted:\n");
   puts(s);
}
```

答：

```
void fun(char *a,int n)
{
   int i=0,k=0;
   char *p,*t;
   p=t=a;
   while(*t)
      t++;
   t--;
   while(*t=='*')
   {
      k++;
      t--;
   }
   if(k>n)
   {
      while(*p&&p<t+n+1)
      {
         a[i]=*p;
         i++;
         p++;
      }
      a[i]='\0';
   }
}
```

【解析】通过第一个 while 循环统计出字符串的有效长度，不包括最后的结束符；第二个 while 循环统计字符串尾部*号的个数，循环结束时，k 值记录了原字符串结尾星号的数目，t 指向字符串中最后一个不为星号的字符的下标。然后通过 if 语句比较尾部*号数是否多于 n 个，若大于则保留 n 个*号和其余字符，总共需要保留的字符数是 t+n+1，1 是指最后的结束符。

# 第 9 章　用户自己建立数据类型

## 一、选择题

1．以下叙述中正确的是（　　）。

A．在 C 语言中，预处理命令行都以“#”开头

B．预处理命令行必须位于 C 源程序的起始位置

C．#include<stdio.h>必须放在 C 程序的开头

D．C 语言的预处理不能实现宏定义和条件编译的功能

【答案】A

【解析】编译预处理就是在 C 编译程序对 C 源程序进行编译前，由编译预处理程序对这些编译预处理命令行进行处理的过程。A 项正确，在 C 语言中，凡是以“#”号开头的行，都称为“编译预处理”命令行。B 项错误，预处理命令行可以出现在程序的任何一行的开始部位，其作用一直持续到源文件的末尾；C 项错误，#include<stdio.h>可以出现在程序的任意一行的开始部位；D 项错误，预处理可以实现宏定义、条件编译和文件包含。答案选择 A 选项。

2．以下关于编译预处理的叙述中错误的是（　　）。

A．预处理命令行必须位于源程序的开始

B．源程序中凡是以#开始的控制行都是预处理命令行

C．一行上只能有一条有效的预处理命令

D．预处理命令是在程序正式编译之前被处理的

【答案】A

【解析】通常，预处理命令位于源文件的开头，也可以写在函数与函数之间。答案选择 A 选项。

3．以下关于宏的叙述中正确的是（　　）。

A．宏名必须用大写字母表示

B．宏定义必须位于源程序中所有语句之前

C．宏替换没有数据类型限制

D．宏调用比函数调用耗费时间

【答案】C

【解析】A 项错误，在 C 语言中，宏名可以是任何合法的 C 语言标识符，只不过通常习惯用大写字母；B 项错误，宏可以根据需要出现在程序的任何一行的开始部位；D 项错误，宏定义是“编译预处理”命令，它们的替换过程在编译时期就已经完成了，因此不会占有程序运行的时间。答案选择 C 选项。

4．以下关于宏的叙述错误的是（　　）。

A．宏替换不具有计算功能

B．宏是一种预处理指令

C．宏名必须用大写字母构成

D．宏替换不占用运行时间

【答案】C

【解析】宏名习惯采用大写字母，以便与一般变量区别，但是并没有规定一定要用大写字母，答案选择 C 选项。

5．以下叙述中错误的是（　　）。

A．在程序中凡是以“#”开始的语句行都是预处理命令行

B．预处理命令行的最后不能以分号表示结束

C．#define MAX 是合法的宏定义命令行

D．C 程序对预处理命令行的处理是在程序执行的过程中进行的

【答案】D

【解析】在 C 语言中，以“#”开头的行都称为“编译预处理命令行”，其中的末尾不得用“;”结束，区别语句的定义和说明语句，选项 A、B 正确。C 项中的宏定义为不带参数的宏命令行，宏名之后可以有替换文本，

也可以没有，选项 C 正确。编译预处理是在编译程序对 C 源程序进行编译前执行的，选项 D 错误。答案选择 D 选项。

6．若程序中有宏定义行：

#define N 100

则以下叙述中正确的是（　　）。

A．宏定义行中定义了标识符 N 的值为整数 100

B．在编译程序对 C 源程序进行预处理时用 100 替换标识符 N

C．上述宏定义行实现将 100 赋给标识符 N

D．在运行时用 100 替换标识符 N

【答案】D

【解析】D 项正确，预处理程序对源程序中所有使用宏名的地方进行直接替换。A 项错误，宏定义没有类型限制；B 项错误，预处理程序进行宏替换，而非编译程序；C 项错误，宏定义不是赋值操作，而是进行在预编译时进行替换。答案选择 D 选项。

7．以下选项中的编译预处理命令行，正确的是（　　）。

A．#define PI 3.14

B．##define eps 0.001

C．#DEFINE TRUE

D．#define int INT

【答案】A

【解析】A 项正确。不带参数的宏定义是用一个指定的标识符来代表一个字符串，其一般形式如下：#define 宏名替换文本。需要注意：①在#define、宏名和替换文本之间用空格隔开；②在 C 程序中，宏定义的定义位置一般写在程序的开头；③宏名一般用大写字母表示，便于与变量名区别；④宏定义是用宏名来表示一个字符串，在宏展开时以该字符串取代宏名，这只是一种简单的代换，预处理程序对它不做任何检查；⑤宏定义不是语句，在行末不加分号，如加上分号则连分号也一起替换。B 项 define 前多了一个#。C 项缺少“替换文本”。D 项不能将一个自定义标识符宏定义为关键字。答案选择 A 选项。

8．下面关于编译预处理的命令行，正确的是（　　）。

A．#define PAI 3.14

B．#Define Eps 0.00001

C．##DEFINE FALSE 0

D．#define int INT

【答案】A

【解析】宏定义用法一般表达式为：

#define 宏名 替换文本

A 项满足宏定义用法；C 语言中是区分大小写的，define 属于 C 语言的关键字，不可以大写，B、C 项错误；D 项中，int 是代表整数类型，不满足 define 用法，D 项错误。答案选择 A 选项。

9．有如下程序：

```
#include <stdio.h>
#define D(x)4*x+1
main()
{
   int i=2,j=4;
   printf("%d\n",D(i+j));
}
```

程序运行后的输出结果是（　　）。

A．25

B．13
C．9
D．12
【答案】B
【解析】带参数的宏定义不是进行简单的字符串替换，而是要进行参数替换。替换过程是：用宏调用提供的实参字符串，直接置换宏定义命令行中相应形参字符串，非形参字符保持不变。调用函数 D(i+j)，进行替换为 4*i+j+1=13，输出 13，答案选择 B 选项。

10．有以下程序：

```
#include <stdio.h>
#define S(x) x *x
main()
{
   int k=5,j=2;
   printf("%d,%d\n",S(k+j+2),S(j+k+2));
}
```

程序的运行结果是（　　）。
A．21,18
B．81,81
C．21,21
D．18,18
【答案】A
【解析】带参数的宏的替换过程是，用宏调用提供的实参字符串直接置换宏定义命令行中相应形参字符串，非形参字符保持不变。S(k+j+2)被置换成 k+j+2*k+j+2，计算时先计算 2*k，结果为 21；S(j+k+2)被置换成 j+k+2*j+k+2，计算时先计算 2*j，结果为 18。程序的运行结果是 21,18。答案选择 A 选项。

11．有以下函数：

```
#include <stdio.h>
#define S(x) (x)*x*2
main()
{
   int k=5,j=2;
   printf("%d,",S(k+j));
   printf("%d\n",S(k-j));
}
```

程序运行后的输出结果是（　　）。
A．98,18
B．39,11
C．39,18
D．98,11
【答案】B
【解析】根据该宏的定义，S(k+j) = (k+j)*k+j*2 = 39，S(k-j) = (k-j)*k-j*2 = 11，答案选择 B 选项。

12．有以下程序

```
#include <stdio.h>
#define S(x) 4*(x)*x+1
main()
{
   int k=5,j=2;
   printf("%d\n",S(k+j));
}
```

程序运行后的输出结果是（　　）。

A. 197

B. 143

C. 33

D. 28

【答案】B

【解析】根据宏定义的规定，题中 S(k+j) = 4*(k+j)*k+j+1，分别代入 k、j 的值后求出的结果为 143。答案选择 B 选项。

13. 有以下程序：

```
#include <stdio.h>
#define PT 3.5
#define S(x) PT*x*x
main()
{
   int a=1,b=2;
   printf("%4.1f\n",S(a+b));
}
```

程序运行后的输出结果是（　　）。

A.  14.0

B.  31.5

C.  7.5

D. 程序有错无输出结果

【答案】C

【解析】宏定义分为两种：①无参数的宏定义，即#define PT 3.5；②带参数的宏定义，即#define S(x) PT*x*x。S(a+b)的运算过程即 3.5*1+2*1+2，最后结果为 7.5，注意，7 前面有一个空格字符。%4.1f 要求输出的浮点数的宽度为 4（包括小数点），且小数点后保留一位小数。若数字宽度不够则添加空格。答案选择 C 选项。

14. 若有以下程序

```
#include <stdio.h>
#define S(x) x*x
#define T(x) S(x)*S(x)
main()
{
   int k=5,j=2;
   printf("%d,%d\n",S(k+j),T(k+j));
}
```

则程序的输出结果是（　　）。

A. 17,37

B. 49,2401

C．17,289

D．49,289

【答案】A

【解析】编译系统处理带参数的宏名时，按程序行中指定的字符串，括号内的内容，从左到右进行处理，若遇到形参则以实参代替，非形参字符原样保留，就形成了替换后的内容，这期间没有任何计算。S(k+j)展开后是5+2*5+2 = 17，T(k+j)展开后是 5+2*5+2*5+2*5+2 = 37。答案选择 A 选项。

15．若有以下程序

```
#include <stdio.h>
#define S(x) (x)*(x)
#define T(x) S(x)/S(x)+1
main()
{
   int k=3,j=2;
   printf("%d,%d\n",S(k+j),T(k+j));
}
```

则程序的输出结果是（　　）。

A．11,2

B．25,2

C．11,12

D．25,26

【答案】D

【解析】S(k+j)展开后是(3+2)*(3+2) = 25，T(k+j)展开后是(3+2)*(3+2)/(3+2)*(3+2)+1 = 26。答案选择 D 选项。

16．以下程序：

```
#include <stdio.h>
#define SUB(a) (a)-(a)
main()
{
   int a=2,b=3,c=5,d;
   d=SUB(a+b)*c;
   printf("%d\n",d);
}
```

程序运行后的结果是（　　）。

A．0

B．-12

C．-20

D．10

【答案】C

【解析】将函数的宏替换代入程序中即可。d = SUB(a+b)*c = SUB(2+3)*5 = (2+3)-(2+3)*5 = 5-25 = -20。答案选择 C 选项。

17．有以下程序

```
#include <stdio.h>
#define SUB(X,Y) (X+1)*Y
main()
{
   int a=3,b=4;
   printf("%d\n",SUB(a++,b++));
}
```

程序运行后的输出结果是（　　）。

A．20

B．16

C．12

D．25

【答案】B

【解析】SUB(a++,b++)展开后是(a+++1)*b++ = (3+1)*4 = 16，答案选择 B 选项。

18．有以下程序：

```
#include<stdio.h>
#define M 5
#define f(x,y) x*y+M
main()
{
   int k;
   k=f(2,3)*f(2,3);
   printf("%d\n",k);
}
```

程序的运行结果是（　　）。

A．22

B．41

C．100

D．121

【答案】B

【解析】宏定义中的函数在调用时只做简单的替换，不能进行任何修改。所以 k=2*3+5*2*3+5=41。答案选择 B 选项。

19．有以下程序

```
#include <stdio.h>
#define N 5
#define M N+1
#define f(x) (x*M)
main()
{
   int i1,i2;
   i1=f(2);
   i2=f(1+1);
   printf("%d %d\n",i1,i2);
}
```

程序的运行结果是（　　）。

A．11 7
B．12 12
C．11 11
D．12 7
【答案】A
【解析】f(2)展开后为，2*5+1，值为 11，f(1+1)展开后为，1+1*5+1 值为 7。答案选择 A 选项。

20．有以下程序：

```
#include <stdio.h>
#define N 2
#define M N+1
#define MUN (M+1)*M/2
main()
{
   printf("%d\n",MUN);
}
```

程序运行后的输出结果是（　　）。
A．8
B．9
C．5
D．6
【答案】A
【解析】带参数的宏的调用格式：宏名(实参表)，替换过程是，用宏调用提供的实参字符串，直接置换宏定义命令行中相应形参字符串，非形参字符保持不变。MUN 被置换成(2+1+1)*2+1/2；程序的运行结果是 8，答案选择 A 选项。

21．有以下程序：

```
#include <stdio.h>
#define f(x) x*x*x
main()
{
   int a=3,s,t;
   s=f(a+1);
   t=f((a+1));
   printf("%d,%d\n",s,t);
}
```

程序运行后的输出结果是（　　）。
A．10,64
B．10,10
C．64,10
D．64,64
【答案】A
【解析】C 语言中带参数的宏可以理解为用参数直接替换定义式中的变量，而不经过任何修改。所以 s = f(a+1) = a+1*a+1*a+1 = 3+3+3+1 = 10，t = f((a+1)) = (a+1)*(a+1)*(a+1) = 4×4×4 = 64，所以有无括号的运算结果是不同的。故答案选择 A 选项。

22．有以下程序：

```
#include <stdio.h>
#define FNA(x) x*x
#define FNB(x) x+x
main()
{
   int a=2,b=4;
   printf("%d,%d\n", FNA(FNB(a)), FNB(FNA(b)));
}
```

程序运行后的输出结果是（　　）。

A. 8,16

B. 16,32

C. 8,32

D. 16,16

【答案】C

【解析】带参数的宏定义不是进行简单的字符串替换，而是要进行参数替换不计算，只是进行简单的替换。替换过程是：用宏调用提供的实参字符串，直接置换宏定义命令行中相应形参字符串，非形参字符保持不变。FNA(FNB(a))的替换过程为：FNB(a)*FNB(a) = a+a*a+a，则 FNA(FNB(2))替换为 2+2×2+2，计算结果为 8。FNB(FNA(b))的替换过程为：FNA(b)+FNA(b) = b*b+b*b，FNB(FNA(b))替换为 4×4+4×4 = 32。程序运行后的输出结果是 8，32，答案选择 C 选项。

23．设有宏定义：#define IsDIV(k,n) ((k%n==1)?1:0)且变量 m 已正确定义并赋值，则宏调用：IsDIV(m,5) && IsDIV(m,7)为真时所要表达的是（　　）。

A. 判断 m 是否能被 5 或者 7 整除

B. 判断 m 是否能被 5 和 7 整除

C. 判断 m 被 5 或者 7 整除是否余 1

D. 判断 m 被 5 和 7 整除是否都余 1

【答案】D

【解析】IsDIV(m,5)&&IsDIV(m,7)为真，即表达式((m%5==1)?1:0)结果为 1，且表达式((m%7==1)?1:0)结果也为 1，也就是 m%5，m%7 都等于 1，所以表达的是，判断 m 被 5 和 7 整除是否都余 1。答案选择 D 选项。

24．有以下程序：

```
#include <stdio.h>
#define F(x) 2.84+x
#define PR(a) printf("%d",(int)(a))
#define PRINT(a) PR(a);putchar('\n')
main()
{
   PRINT(F(5)*2);
}
```

程序运行后的输出结果是（　　）。

A. 12

B. 13

C. 15

D. 11

【答案】A

【解析】直接置换宏定义命令行中相应形参字符串，非形参字符保持不变。将 PRINT(F(5)*2)用 PR(F(5)*2); putchar('\n')替换，将 PR(F(5)*2)用 printf("%d", (int)(F(5)*2))替换，再将 F(5)用 2.84+5 替换，最后替换结果为 printf("%d", (int)(2.84+5*2)); putchar('\n')，运行结果为：12<回车>。答案选择 A 选项。

25．有以下程序

```
#include<stdio.h>
main()
{
   int s,t,A=10;
   double B=6;
   s=sizeof(A);
   t=sizeof(B);
   printf("%d,%d\n",s,t);
}
```

在 VC++2010 平台上编译运行，程序运行后的输出结果是（　　）。

A．2,4

B．4,4

C．4,8

D．10,6

【答案】C

【解析】sizeof 的作用就是返回一个对象或者类型所占的内存字节数。在 VC++2010 中整型占 4 个字节，双精度实型占 8 个字节。答案选择 C 选项。

26．有以下程序段

```
int *p;
p=_____ malloc(sizeof(int));
```

若要求使 p 指向一个 int 型的动态存储单元，在横线处应填入的是（　　）。

A．int

B．(int*)

C．int*

D．(*int)

【答案】B

【解析】C 语言标准规定 malloc 函数返回值的类型为 void*，函数的调用形式为：malloc(size)，size 的类型为 unsigned int。p 是指向 int 型的指针，要把 void *强制转换成 int *。答案选择 B 选项。

27．有以下程序

```
#include <stdio.h>
#include <stdlib.h>
main()
{
   int *a,*b,*c;
   a=b=c=(int*)malloc(sizeof(int));
   *a=1;
   *b=2,*c=3;
   a=b;
   printf("%d,%d,%d\n",*a,*b,*c);
}
```

程序运行后的输出结果是（　　）。

A．3,3,3

B．2,2,3

C．1,2,3

D．1,1,3

【答案】A

【解析】库函数 malloc(size)用来分配 size 个字节的存储区，返回一个指向存储区首地址的基类型为 void 的地址。本程序中，系统只分配了一个整型数据的存储空间，并把这个空间的地址分别赋给了指针型变量 a，b 和 c。程序利用指针 a 把数据 1 写入了该空间，然后利用指针 b，把数据 2 写入该空间，因为指针型变量 a、b 和 c 指向了同一个存储空间，所以数据 2 将原来的 1 覆盖掉了，同理数据 3 又将 2 覆盖掉了。因此，此空间中最后留有的数据是 3。又因为 3 个指针都指向该空间，所以输出数据均为 3。答案选择 A 选项。

28．有以下程序：

```
#include<stdio.h>
#include<stdlib.h>
int fun(int n)
{
   int *p;
   p=(int*)malloc(sizeof(int));
   *p=n;
   return *p;
}
main()
{
   int a;
   a=fun(10);
   printf("%d\n",a+fun(10));
}
```

程序运行的结果是（　　）。

A．0

B．10

C．20

D．出错

【答案】C

【解析】fun 函数的功能是申请一个 int 型指针 p，把 p 指向的存储空间赋值为 n，并返回 p 指向的空间的值，即为 n。fun(10)的返回值为 10，所以 a=fun(10)后 a 的值为 10，a+fun(10)=20。答案选择 C 选项。

29．有以下程序：

```
#include <stdio.h>
#include <stdlib.h>
void fun(int*p1,int*p2,int*s)
{
   s=(int*)malloc(sizeof(int));
   *s=*p1+*p2;
   free(s);
}
main()
{
   int a=1,b=40,*q=&a;
   fun(&a,&b,q);
   printf("%d\n",*q);
}
```

程序运行后的输出结果是（　　）。

A．42

B．0

C．1

D．41

【答案】C

【解析】main 函数中定义了 3 个变量，a，b 和指针变量 q，并且 q 存放的是 a 的地址，fun 函数中 s 是重新分配的空间，将重分配的空间中存放*p1 和*p2 即为 1 和 40，然后释放 s，而 q 没有变化，仍然指向变量 a，故*q 仍然为 1，答案选择 C 选项。

30．有以下程序

```
#include <stdio.h>
#include <stdlib.h>
void fun(int*p1,int*p2,int*s)
{
   s=(int*)malloc(sizeof(int));
   *s=*p1+*p2;
}
main()
{
   int a[2]={1,2}, b[2]={10,20},*s=a;
   fun(a,b,s);
   printf("%d\n",*s);
}
```

程序运行后的输出结果是（　　）。

A．1

B．10

C．11

D．2

【答案】A

【解析】初始化后，s 指向数组 a 的首地址，在调用函数 fun 时，形参 s 和实参 s 是两个独立的 int 型指针，在 fun 函数内部，形参 s 指向新的存储空间，但是不会影响实参 s 的指向。所以，在 main 函数中，s 仍然指向数组 a 的首地址，输出结果是 1。答案选择 A 选项。

31．有以下程序：

```
#include <stdio.h>
#include <stdlib.h>
void fun(int*p1,int*s)
{
   int *t;
   t=(int*)malloc(2*sizeof(int));
   *t=*p1+*p1++;
   *(t+1)=*p1+*p1;
   s=t;
}
main()
{
   int a[2]={1,2}, b[2]={0};
   fun(a,b);
   printf("%d,%d\n",b[0],b[1]);
}
```

程序运行后的输出结果是（　　）。

A．2,6

B．0,0

C．2,4

D．1,2

【答案】B

【解析】程序执行过程为：调用 fun(a,b)，形参 p1 对应实参 a，形参 s 对应实参 b，定义指针 t 并使其指向开辟的两个整型数据大小的内存，为第一个内存赋值为数组 a 第一个元素的 2 倍，即 2，然后指针 p1 指向 a 数组第二个元素，为第二个内存赋值为数组 a 第二个元素的 2 倍，即 4，最后使指针 s 指向动态开辟的两个内存的首地址。但对形参的操作没有影响到实参，整个过程中数组 b 没有发生变化，依次输出 b 中元素为 0，0，答案选择 B 选项。

32．有以下程序：

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
main()
{
   char*p1,*p2;
   p1=p2=(char*)malloc(sizeof(char)*10);
   strcpy(p1,"malloc");
   strcpy(p2,p1+1);
   printf("%c%c\n", p1[0], p2[0]);
}
```

程序的运行结果是（　　）。

A．aa

B．ma

C．am

D．mm

【答案】A

【解析】malloc 函数的原型为：malloc(size);，函数的作用是在内存的动态存储区申请分配一个长度为 size 的连续空间。程序执行过程为：定义两个字符类型指针 p1 和 p2，动态开辟 10 个字符类型的内存单元，并且使

指针 p1 与 p2 均指向这 10 个内存单元的第一个单元。调用 strcpy 函数，将字符串“malloc”赋值给这 10 个内存单元的前 7 个单元，存储情况为：malloc\0，此时 p1 指向 10 个内存单元的第一个单元。再次调用 strcpy 函数，这 10 个内存单元的第 2 个单元到第 7 个单元元素重新赋值给 p2 指向的内存单元以及以后的 5 个单元，存储情况为：alloc\0\0，此时 p1 和 p2 均指向 10 个内存单元的第一个单元。输出两个指针指向单元的数据值，结果为：aa，答案选择 A 选项。

33．以下叙述中错误的是（　　）。

A．用 typedef 定义新的类型名后，原有类型名仍有效

B．可以用 typedef 将已存在的类型用一个新的名字来代表

C．可以通过 typedef 增加新的类型

D．用 typedef 可以为各种类型起别名，但不能为变量起别名

【答案】C

【解析】关键字 typedef 的作用只是将 C 语言中已有的数据类型作置换，并不是增加新的类型，答案选择 C 选项。

34．以下关于 typedef 的叙述错误的是（　　）。

A．用 typedef 为类型说明一个新名，通常可以增加程序的可读性

B．typedef 只是将已存在的类型用一个新的名字来代表

C．用 typedef 可以为各种类型说明一个新名，但不能用来为变量说明一个新名

D．用 typedef 可以增加新类型

【答案】D

【解析】用 typedef 只是将已存在的类型用一个新的名字来代表，并没有增加新类型。答案选择 D 选项。

35．以下叙述中错误的是（　　）

A．可以用 typedef 说明的新类型名来定义变量

B．typedef 说明的新类型名必须使用大写字母，否则会出编译错误

C．用 typedef 可以为基本数据类型说明一个新名称

D．用 typedef 说明新类型的作用是用一个新的标识符来代表已存在的类型名

【答案】B

【解析】typedef 定义的新类型名习惯上用大写字母，但没有强制要求用大写字母。答案选择 B 选项。

36．若有说明：typedef struct{int a;char c;}w;，则以下叙述正确的是（　　）。

A．编译后系统为 w 分配 5 个字节

B．编译后系统为 w 分配 6 个字节

C．编译后系统为 w 分配 8 个字节

D．编译后系统不为 w 分配存储空间

【答案】D

【解析】w 是一个自定义类型，不是变量，故编译后系统不为 w 分配存储空间。当 w 定义为结构体变量时才会为其分配存储空间。答案选择 D 选项。

37．以下结构体类型说明和变量定义中正确的是（　　）。

A．typedef struct{int n;char c;}REC;REC t1,t2;

B．struct REC;{int n;char c;};REC t1,t2;

C．typedef struct REC:{int n=0;char c='A';}t1,t2;REC t1,t2;

D．struct{int n;char c;}REC;

【答案】A

【解析】A 项，用 typedef 定义结构体类型名 REC 后，可以用 REC 定义变量。B 项，“struct REC;”后面不应加分号；C 项，没有这样的书写形式；D 项，REC 定义的是一个变量，不是结构体类型，因此不能用来定义结构体变量。答案选择 A 选项。

38．设有以下语句

typedef struct TT

{char c;int a[4]} CIN;

则下面叙述中正确的是（　　）。

A．CIN 是 structTT 类型的变量

B．TT 是 struct 类型的变量

C．可以用 TT 定义结构体变量

D．可以用 CIN 定义结构体变量

【答案】D

【解析】CIN 使用 typedef 定义的类型名，可以用来定义结构体 TT 类型的变量。答案选择 D 选项。

39．若有以下语句

typedef struct S

{int g;char h;}T;

以下叙述中正确的是（　　）。

A．可用 S 定义结构体变量

B．可用 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

【答案】B

【解析】AC 两项，S 是定义的结构体的名字，并不是 struct 类型的变量也不可用来定义结构体变量；D 项，题目中 T 定义为 struct S 类型，即 T 被定义为一个类型名，而不是变量。答案选择 B 选项。

40．设有如下语句：

```
typedef struct Date
{
   int year;
   int month;
   int day;
} DATE;
```

则以下叙述中错误的是（　　）。

A．DATE 是用户定义的结构体变量

B．struct Date 是用户定义的结构体类型

C．DATE 是用户说明的新结构体类型名

D．struct 是结构体类型的关键字

【答案】A

【解析】C 语言允许用 typedef 说明一种新类型名，其一般形式如下：typedef 类型名 新类型名;，题目中定义了新类型 Date，这种类型变量包含 3 个成员，DATE 是这种新类型的名字，不是结构体变量，C 项正确，A 项错误。struct Date 是用户定义的结构体类型，B 项正确。struct 是结构体类型的关键字，D 项正确。答案选择 A 选项。

41．若有定义：

typedef int* T;

T a[20];

则以下与上述定义中 a 类型完全相同的是（　　）。

A．int *a[20];

B．int (*a)[20];

C．int a[20];

D．int **a[20];

【答案】A

【解析】“typedef 类型名 新类型名”表示为一个已定义的类型标识符重新定义一个新类型名，题中选项 A 代表的意思是定义了一个指针数组，该数组的每一个元素都是整型指针。B 选项是数组指针也称指向一维数组的指针（行指针）。首先为整型指针类型命名为 T，再通过 T 定义了一个整型指针数组*a[20]，等价于 int*a[20]，答案选择 A 选项。

42．若有定义：

typedef int T[10];

T *a[20];

则与上述定义完全等价的说明语句是（　　）。

A．int *a[20][10];

B．int *a[20];

C．int *a[10];

D．int *a[10][20];

【答案】A

【解析】新定义的类型 T 为大小为 10 的整型数组，定义 T 型指针数组*a[20]，等价于定义了一个指向整型长度为 10 的数组的指针数组，这个指针数组大小为 20，即为整型指针二维数组，行 20 列 10，答案选择 A 选项。

43．有以下定义：

struct data

{int i;char c;double d;}x;

以下叙述中错误的是（　　）。

A．x 的内存地址与 x.i 的内存地址相同

B．struct data 是一个类型名

C．初始化时，可以对 x 的所有成员同时赋初值

D．成员 i、c 和 d 占用的是同一个存储空间

【答案】D

【解析】变量 i、c、d 是结构体变量 x 中三个不同的成员，占用不同的存储空间。答案选择 D 选项。补充：区分 struct 和 union，union 的各个数据成员共享一块存储空间，struct 不同的成员，占用不同的存储空间。

44．下面结构体的定义语句中，错误的是（　　）。

A．struct ord{int x;int y;int z;};struct ord a;

B．struct ord{int x;int y;int z;}struct ord a;

C．struct ord{int x;int y;int z;}a;

D．struct {int x;int y;int z;}a;

【答案】B

【解析】C 语言中结构体变量的定义有三种方法：①定义结构体类型的同时定义结构体变量，如 C 项；②使用无名结构体类型定义结构体变量，如 D 项；③先定义结构体类型，后定义结构体变量，如 A 项，B 项错在分别定义结构体类型与结构体变量时需要用“;”隔开。故答案选择 B 选项。

45．设有定义：

struct complex

{int real,unreal;}data1={1,8},data2;

则以下赋值语句中错误的是（　　）。

A．data2=data1;

B．data2=(2,6);

C．data2.real=data1.real;

D．data2.real=data1.unreal:

【答案】B

【解析】B 项，对结构体进行初始化时，应该用花括号括起来的一组值，而不是用小括号，而且需要强制转

换数据类型，应该为 data2=(struct complex){2,6};。答案选择 B 选项。

46．设有定义：struct{int n;float x;}s[2],m[2]={{10,2.8},{0,0.0}};，则以下赋值语句中正确的是（　　）。

A．s[0]=m[1];

B．s=m;

C．s.n=m.n;

D．s[2].x=m[2].x;

【答案】A

【解析】定义了结构体类型数组 s，长度为 2，结构体类型数组 m，长度为 2，并对数组 m 进行了初始化。同类型的结构体可以直接用变量名实现赋值，A 项正确；数组名为数组首地址，地址常量之间不可以相互赋值，B 项错误；数组名为地址常量不是结构体变量，不能引用成员，C 项错误；s[2]与 m[2]数组越界，D 项错误。答案选择 A 选项。

47．有以下程序段

```
struct st
{
   int x;
   int *y;
}*pt;
int a[]={1,2},b[]={3,4};
struct st c[2]={10,a,20,b};
pt=c;
```

以下选项中表达式的值为 11 的是（　　）。

A．++pt->x

B．pt->x

C．*pt->y

D．(pt++)->x

【答案】A

【解析】pt->x 值为 10，->优先级高于++，前置++表达式的值为加 1 之后的值，答案选择 A 选项。

48．有以下定义和语句：

```
struct workers
{
   int num;
   char name[20];
   char c;
   struct
   {
      int day;
      int month;
      int year;
   }s;
};
struct workers w,*pw;
pw=&w;
```

能给 w 中 year 成员赋 1980 的语句是（　　）。

A．*pw.year=1980;

B．w.year=1980;

C．pw->year=1980;

D．w.s.year=1980;

【答案】D

【解析】w 是一个结构体变量，pw 是一个结构体指针变量，指向 w 所在的内存单元。A 项错误，pw 是指针，所以在引用其内部变量时应该用操作符“->”而不是“.”。BC 两项错误，year 是结构体 workers 中的结构体成员 s 中的成员。答案选择 D 选项。

49．设有如下定义：

struct{int n;char c;}a[2], *p=a;

则以下错误引用结构体成员 n 的是（　　）。

A．(*a)->11

B．a[0].n

C．p->n

D．(*p).n

【答案】A

【解析】可用以下 3 种形式来引用结构体变量中的成员，其中结构体变量名也可以是已定义的结构体数组的数组元素：①结构体变量名.成员名；②指针变量名->成员名；③(*指针变量名).成员名。题目中 a 为数组名是地址常量，不是指针变量，A 项引用错误。a[0]为变量名，B 项引用正确。p 为指针，且正确的指向结构体变量，C 项引用正确。D 项符合第三种引用方式，引用正确。答案选择 A 选项。

50．设有以下程序段

```
struct MP3
{
    char name[20];
    char color;
    float price;
}std,*ptr;
ptr=&std;
```

若要引用结构体变量 std 中的 color 成员，写法错误的是（　　）。

A．std.color

B．ptr->color

C．std->color

D．(*ptr).color

【答案】C

【解析】结构体变量的引用两种方式：①用箭头操作符：“->”，其中操作数必须是指向结构的指针，右操作数是该结构的成员；②结构体变量名.成员名。C 项，“std”是结构体变量名，应该使用“.”操作符来引用结构体成员。答案选择 C 选项。

51．有如下定义：

```
struct
{
   int num;
   char name[10];
   struct
   {
      int y;
      int m;
      int d;
   }birth;
}s,*ps=&s;
```

以下对内嵌结构体成员的引用形式错误的是（　　）。

A. ps.birth.y

B. ps.birth.y

C. ps->birth.y

D. (*ps).birth.y

【答案】A

【解析】使用结构体指针对结构体成员进行访问时，形式为：结构指针名->结构成员名，或者(*结构指针名).结构成员名，题目中 ps 为结构体指针，答案选择 A 选项。

52. 有以下程序：

```
#include<stdio.h>
struct S{int a;int *b;};
main()
{
   int x1[] = {3,4},x2[] = {6,7};
   struct S x[] = {1,x1,2,x2};
   printf("%d,%d\n",*x[0].b,*x[1].b);
}
```

程序的运行结果是（　　）。

A. 1,2

B. 3,6

C. 4,7

D. 变量的地址值

【答案】B

【解析】程序的执行过程为：定义整型数组 x1，x2 并进行初始化，两个数组长度均为 2。定义结构体数组 x，并为其初始化，则 x[0].a=1，x[0].b=x1，x[1].a=2，x[1].b=x2。输出 x[0]的成员指针 b 指向的内存单元值，即数组 x1 的第一个元素 3，输出 x[1]的成员指针 b 指向的内存单元值，即数组 x2 的第一个元 6。程序的运行结果是 3，6。答案选择 B 选项。

53. 若有以下定义：

```
struct tt{char name[10];char sex;}aa={"aaaa",'F'},*p=&aa;
```

则错误的语句是（　　）。

A. scanf("%c",aa.sex);

B. aa.sex=getchar();

C. printf("%c\n",(*p).sex);

D. printf("%c\n",p->sex);

【答案】A

【解析】sex 是一个 char 类型变量，不是地址，A 项应为 scanf("%c",&aa.sex);。答案选择 A 选项。

54．有以下结构体说明、变量定义和赋值语句

```
struct STD
{
    char name[10];
    int age;
    char sex;
}s[5],*ps;
ps = &s[0];
```

则以下 scanf 函数调用语句有错误的是（　　）。

A．scanf("%d",ps->age);

B．scanf("%d",&s[0].age);

C．scanf("%c",&(ps->sex));

D．scanf("%s",s[0].name);

【答案】A

【解析】A 项错误，ps->age 是取 s[0]中的 age 成员，scanf 函数中需要传入变量的地址；B 项正确，[]和.操作符优先级高于&，等价于&(s[0].age)；C 项正确，ps->sex 是取 s[0]的 sex 成员；D 项正确，s[0].name 是取 s[0]中的 name 成员，name 是 char 类型数组，自身就是首元素地址。答案选择 A 选项。

55．有如下定义：

```
struct st
{
    char name[12];
    int age;
    char sex;
}std[10], *p=std;
```

以下语句错误的是（　　）。

A．scanf("%d", p->age);

B．scanf("%s", std[0].name);

C．scanf("%d", &std[1].age);

D．scanf("%c", &(p->sex));

【答案】A

【解析】A 项中，p->age 是结构指针访问结构成员的方式，p->age 为整型，使用 scanf 输入时，在 p->age 前面应该加入取地址符&。答案选择 A 选项。

56．有如下程序：

```
struct person
{
    char name[10];
    char sex;
    float weight;
}zhangsan, *ptr;
ptr=&zhangsan;
```

若要从键盘读入姓名给结构体变量 zhangsan 的 name 成员，输入项错误的是（　　）。

A．scanf("%s",zhangsan->name);

B．scanf("%s",zhangsan.name);

C．scanf("%s",ptr->name);

D．scanf("%s",(*ptr).name);

【答案】A

【解析】可用以下 3 种形式来引用结构体变量中的成员：①结构体变量名.成员名；②指针变量名->成员名；③(*指针变量名).成员名。程序定义了一个结构体变量 zhangsan，一个结构体指针 ptr，并且使指针指向变量 zhangsan。A 项 zhangsan 为结构体变量名，不能用->引用其成员，A 项错误。zhangsan.name、ptr->name、(*ptr).name 均是正确的引用结构体变量成员的方式。答案选择 A 选项。

57．设有定义

```
struct
{
   char mark[12];
   int num1;
   double num2;
}t1,t2;
```

若变量均已正确赋初值，则以下语句中错误的是（　　）。

A．t2.mark=t1.mark;

B．t2.num1=t1.num1;

C．t1=t2;

D．t2.num2=t1.num2;

【答案】A

【解析】A 项错误，struct 中的 mark 成员是数组，数组名是一个地址常量，不能对数组名直接赋值；B 项正确，num1 是 int 变量，可以直接赋值；C 项正确，C 语言中 struct 变量可以直接赋值，实际操作是内存拷贝；D 项正确，num2 是 int 变量，可以直接赋值；C 语言中，数组不能直接赋值，但是把数组包装在 struct 中，就可以实现直接赋值。答案选择 A 选项。

58．以下叙述中正确的是（　　）。

A．函数的返回值不能是结构体指针类型

B．函数的返回值不能是结构体类型

C．在调用函数时，可以将结构体变量作为实参传给函数

D．结构体数组不能作为参数传给函数

【答案】C

【解析】C 项正确，结构体变量作为实参时，是传值形式调用；AB 两项错误，函数的返回值可以是结构体变量和结构体指针变量；D 项错误，向函数传递结构体数组名时，传递的是实参结构体数组的首地址，是传引用形式调用。答案选择 C 选项。

59．以下叙述中正确的是（　　）。

A．结构体数组名不能作为实参传给函数

B．结构体变量的地址不能作为实参传给函数

C．结构体中可以含有指向本结构体的指针成员

D．即使是同类型的结构体变量，也不能进行整体赋值

【答案】C

【解析】C 项正确，结构体中的成员可以是各种类型的指针变量。当一个结构体中有一个或多个成员的基类型就是本结构体类型时，通常把这种结构体称为可以“引用自身的结构体”，定义二叉树结构时，就使用了这种结构体。AB 两项错误，结构体的数组名和结构体变量的地址作为实参时，都是传引用形式调用函数；D 项错误，同类型的结构体变量之间可以直接赋值，实际操作是内存空间拷贝。答案选择 C 选项。

60．有如下程序：

```
#include<stdio.h>
struct S
{
   int x,y;
};
main()
{
   struct S data[2] = {4,3,1,9};
   int i;
   for(i=0;i<2;i++)
      printf("%d,%d;",data[i].x, data[i].y>>1);
}
```

程序运行后的输出结果是（　　）。

A. 4,1;1,4;

B. 4,1;2,4;

C. 4,3;1,9;

D. 4,3;2,3;

【答案】A

【解析】“>>”右移运算符。程序执行过程为：定义结构体数组 data，并且初始化，则 data[0].x=4，data[0].y=3，data[1].x=1，data[1].y=9。for 循环依次输出结构体变量的成员与经过位运算之后的结果。3=11B，9=1001B，data[0].y>>1=1，data[1].y>>1=4。答案选择 A 选项。

61．有以下程序：

```
#include<stdio.h>
struct S
{
   int x,y;
};
main()
{
   struct S data[3] = {4,3,2,0,8,1};
   int i;
   for(i=0;i<3;i++)
      printf("%d%d;",data[i].x, data[i].y>>1);
   printf("\n");
}
```

程序运行后的输出结果是（　　）。

A. 41;20;80;

B. 41;22;64;

C. 40;21;80;

D. 43;20;81;

【答案】A

【解析】“>>”右移运算符,右移一位等价于除以 2 后取整。程序中定义结构体数组 data 并完成初始化，data[0].x=4，data[0].y=3，data[1].x=2，data[1].y=0，data[2].x=8，data[2].y=1。在 for 循环中，对 3 个结构体元素 y 值转换成 2 进制数后向右移一位，再对此时 x，y 值依次进行输出，答案选择 A 选项。

62．有如下程序：

```
#include<stdio.h>
struct person
{
   char name[10];
   int age;
};
main()
{
   struct person room[2] = {{"Wang",19},{"Li",20}};
   printf("%s:%d\n", (room+1)->name, room->age);
}
```

程序运行后的输出结果是（　　）。

A. Li:19

B. Wang:19

C. Li:20

D. Wang:17

【答案】A

【解析】room 表示数组首地址，首地址+1，指向 room 数组中第二个元素，并将该元素的 name 信息进行输出，同理，对第一个元素中的 age 信息输出。答案选择 A 选项。

63. 有以下程序

```
#include <stdio.h>
struct tt
{
   int x;
   struct tt *y;
} *p;
struct tt a[4]={20,a+1,15,a+2,30,a+3,17,a};
main()
{
   int i;
   p=a;
   for(i=1;i<=2;i++)
   {
      printf("%d,",p->x);
      p=p->y;
   }
}
```

程序的运行结果是（　　）。

A. 20,15,

B. 30,17,

C. 15,30,

D. 20,30,

【答案】A

【解析】本题中结构体 struct tt 的成员包括它本身的指针 struct tt*，定义了结构体数组 a，并完成初始化，a[0].x=20，a[0].7=a+1，……，a[3].x=17,a[3].y=a。在主函数中，第一次 for 循环，i=1 时：指针 p 指向数组 a 的首地址，即&a[0]，因此 p->x=a[0].x=20，p->y=a[0].y=a+1，则 p=p->y 就是将指针 p 后移一位指向 a[1]；i=2 时，输出 p->x 等价于输出 a[1].x，即 15，之后再将 p 后移；当 i=3 时跳出循环。答案选择 A 选项。

64．有以下程序：

```
#include<stdio.h>
struct st
{int x,y;}data[2]={1,10,2,20};
main()
{
   struct st*p=data;
   printf("%d,",p->y);
   printf("%d\n",(++p)->x);
}
```

程序运行的结果是（　　）。

A．10,1

B．20,1

C．10,2

D．20,2

【答案】C

【解析】程序在定义结构体 st 的同时定义了结构体数组 data。可知 data[0]={1,10}，data[1]={2,20}。结构体指针 p 指向数组的首地址 data[0]，++p 则指向 data[1]，所以输出 p->y，(++p)->x 即输出 data[0].y 和 data[1].x。从初始化列表中可以看出，这两个值分别是 10 和 2。答案选择 C 选项。

65．有以下程序：

```
#include <stdio.h>
struct ord
{int x,y;}dt[2]={1,2,3,4};
main()
{
   struct ord *p=dt;
   printf("%d,",++p->x);
   printf("%d",++p->y);
}
```

程序运行后的输出结果是（　　）。

A．1,2

B．2,3

C．3,4

D．4,1

【答案】B

【解析】dt 是一个结构体数组，所以初始化的结果为 dt[0]={1,2}，dt[1]={3,4}。*p=dt;表示 p 指向 dt[0]。++p->x 中，->的优先级大于++，所以表达式等价于++(p->x)，因为 p->x 为 1，所以输出这个值为 2，同理，第二个++p->y 的值为 2+1=3。答案选择 B 选项。

66．有以下程序：

```
#include <stdio.h>
struct S
{
   int a,b;
}data[2]={10,100,20,200};
main()
{
   struct S p=data[1];
   printf("%d\n",++(p.a));
}
```

程序运行后的输出结果是（　　）。

A．10

B．11

C．20

D．21

【答案】D

【解析】题中定义了一个包含两个元素的结构体数组，其中 data[0].a=10，data[0].b=100，data[1].a=20，data[1].b=200。指针 p 指向结构体数组的第 2 个元素，那么 p.a 的值为 20，++(p.a)的值为 21，所以输出结果为 21。答案选择 D 选项。

67．有以下程序：

```
#include <stdio.h>
#include <string.h>
struct S
{
   char name[10];
};
main()
{
   struct S s1,s2;
   strcpy(s1.name,"XXX");
   strcpy(s2.name,"=");
   s1=s2;
   printf("%s\n",s1.name);
}
```

程序运行后的输出结果是（　　）。

A．=

B．XXX

C．=XX

D．X=

【答案】A

【解析】主函数首先定义两个结构体变量 s1，s2，分别使用字符串拷贝函数 strcpy 为 s1 和 s2 的成员 name 赋值，再将 s2 中的成员信息赋值给 s1，因此输出为“=”，答案选择 A 选项。

68．有如下程序：

```
#include <stdio.h>
#include <string.h>
struct S
{
   char name[10];
};
main()
{
   struct S s1,s2;
   strcpy(s1.name,"12345");
   strcpy(s2.name,"ABC");
   s1=s2;
   printf("%s\n",s1.name);
}
```

程序运行后的输出结果是（　　）。

A. ABC12

B. ABC45

C. 12345

D. ABC

【答案】D

【解析】主函数首先定义两个结构体变量 s1，s2，分别使用字符串拷贝函数 strcpy 为 s1 和 s2 的成员 name 赋值，再将 s2 中的成员信息赋值给 s1，因此输出为“ABC”，答案选择 D 选项。

69．有以下程序

```
#include <stdio.h>
typedef struct
{
   int b,p;
}A;
void f(A c)
{
   int j;
   c.b+=1;
   c.p+=2;
}
main()
{
   int i;
   A a={1,2};
   f(a);
   printf("%d,%d\n",a.b,a.p);
}
```

程序运行后的输出结果是（　　）。

A. 1,2

B. 2,4

C. 1,4

D. 2,3

【答案】A

【解析】当把一个结构体变量传送给相应的形参时，传递的是实参结构体变量中的值，系统将为结构体类型形参开辟相应的存储单元，并将实参中各成员的值一一对应赋给形参中的成员，函数中形参结构体变量的改变不会影响到实参结构体变量。所以，函数 f 不会改变 a 中的数据。答案选择 A 选项。

70．有如下程序：

```
#include <stdio.h>
struct person
{
   char name[10];
   int age;
};
main()
{
   struct person room[4] = {{"Zhang",19}, {"Li",20}, {"Wang",17}, {"Zhao",18}};
   printf("%s:%d\n",(room+2)->name, room->age);
}
```

程序运行后的输出结果是（　　）。

A．Wang:19

B．Wang:17

C．Li:20

D．Li:19

【答案】A

【解析】可用以下 3 种形式来引用结构体变量中的成员：①结构体变量名.成员名；②指针变量名->成员名；③(*指针变量名).成员名。数组名 room 是指向数组首地址，也可以当做指向数组的指针来使用。room+2 指向数组第三个元素，(room+2)->name 为字符串 Wang；room 指向数组第一个元素，room->age=19。答案选择 A 选项。

71．有以下程序：

```
#include<stdio.h>
#include<string.h>
typedef struct
{
   char name[9];
   char sex;
   float score[2];
}STU;
void f(STU*a)
{
   strcpy(a->name,"Zhao");
   a->sex='m';
   a->score[1]=90.0;
}
main( )
{
   STU c={"Qian",'f',95.0,92.0}, *d=&c;
   f(d);
   printf("%s,%c,%2.0f,%2.0f\n", d->name, c.sex, c.score[0], c.score[1]);
}
```

程序的运行结果是（　　）。

A．Qian,f,95,92

B．Zhao,f,95,90

C．Zhao,m,95,90

D．Zhao,f,95,92

【答案】C

【解析】f 函数调用时，结构体数组名作为实参传给形参指针，结构体指针 a 指向数组 c 的首地址。因此，f 可以对数组 c 中的元素赋值，故返回主函数之后，数组 c 中的成员值已被更新。main 函数中有赋值语句“*d=&c;”，指针 d 指向结构体数组 c 的首地址，故 d->name=c.name，输出结果为 Zhao,m,95,90。答案选择 C 选项。

72．有以下程序：

```
#include<stdio.h>
main()
{
   struct STU
   {
      char name[9];
      char sex;
      double score[2];
   };
   struct STU a={"Zhao",'m',85.0,90.0}, b={"Qian",'f',95.0,92.0};
   b=a;
   printf("%s,%c,%2.0f,%2.0f\n", b.name, b.sex, b.score[0], b.score[1]);
}
```

程序运行的结果是（　　）。

A．Qian,f,95,92

B．Qian,f,85,90

C．Zhao,f,95,92

D．Zhao,m,85,90

【答案】D

【解析】在 C 语言中，相同类型的结构体变量可以通过等号直接赋值，它会将对应成员一一对应赋值。所以，本题声明并初始化了两个 STU 结构体变量 a 和 b，然后将 a 赋给 b，最后逐个输出 b 的各个成员，其实就是初始化 a 的内容。答案选择 D 选项。

73．有以下程序：

```
#include <stdio.h>
#include <string.h>
typedef struct
{
   char name[9];
   char sex;
   float score[2];
} STU;
void f(STU a)
{
   STU b={"zhao",'m',85.0,90.0};
   int i;
   strcpy(a.name, b.name);
   a.sex = b.sex;
   for(i=0; i<2; i++) a.score[i]=b.score[i];
}
main()
{
   STU c={"Qian",'f',95.0, 92.0};
   f(c);
   printf("%s,%c,%2.0f,%2.0f\n", c.name, c.sex, c.score[0], c.score[1]);
}
```

程序的运行结果是（　　）。

A．Qian,f,95,92

B．Qian,m,85,90

C．Zhao,f,95,92

D．Zhao,m,85,90

【答案】A

【解析】结构体作为函数参数时是传值调用。本题中，函数传递的是实参结构体变量中的值。函数体内对形参结构体变量中任何成员的操作都不会影响对应实参中成员的值。因此，f 函数没有任何实际作用。答案选择 A 选项。

74．有以下程序

```
#include <stdio.h>
#include <string.h>
typedef struct
{
   char name[9];
   char sex;
   float score[2];
} STU;
STU f(STU a)
{
   STU b={"zhao",'m',85.0,90.0};
   int i;
   strcpy(a.name, b.name);
   a.sex = b.sex;
   for(i=0; i<2; i++) a.score[i]=b.score[i];
   return a;
}
main()
{
   STU c={"Qian",'f',95.0,92.0},d;
   d=f(c);
   printf("%s,%c,%2.0f,%2.0f\n", d.name, d.sex, d.score[0], d.score[1]);
}
```

程序的运行结果是（　　）。

A．Qian,m,85,90

B．Zhao,m,85,90

C．Qian,f,95,92

D．Zhao,f,95,92

【答案】B

【解析】struct 变量作为形参和返回值时，传递的是结构体类型的值。f 函数中，把 b 变量赋值给 a，然后将 a 返回并赋值给 d，所以，d 中的数据与 b 的数据相同。答案选择 B 选项。

75．有以下程序

```
#include <stdio.h>
#include <string.h>
typedef struct
{
   char name[9];
   char sex;
   int score[2];
} STU;
STU f(STU a)
{
   STU b={"zhao",'m',85,90};
   int i;
   strcpy(a.name, b.name);
   a.sex = b.sex;
   for(i=0; i<2; i++) a.score[i]=b.score[i];
   return a;
}
main()
{
   STU c={"Qian",'f',95,92},d;
   d=f(c);
   printf("%s,%c,%d,%d,", d.name, d.sex, d.score[0], d.score[1]);
   printf("%s,%c,%d,%d\n", c.name, c.sex, c.score[0], c.score[1]);
}
```

程序运行后的输出结果是（　　）。

A．Zhao,m,85,90,Qian,f,95,92

B．Zhao,m,85,90,Zhao,m,85,90

C．Qian,f,95,92,Qian,f,95,92

D．Qian,f,95,92,Zhao,m,85,90

【答案】A

【解析】本题考查的是函数调用时的参数传递问题。struct 变量作为形参和返回值时，传递的是结构体类型的值。f 函数中，把 b 变量赋值给 a，然后将 a 返回并赋值给 d，所以，d 中的数据与 b 的数据相同，但 c 中的成员并没有受到任何影响。答案选择 A 选项。

76．有以下程序：

```
#include <stdio.h>
struct STU
{
    char name[9];
    char sex;
    int score[2];
};
void f(struct STU a[])
{
    struct STU b = {"zhao",'m',85,90};
    a[1]=b;
}
main()
{
    struct STU c[2] = {{"Qian",'f',95,92}, {"Qian",'f',95,92}};
    f(c);
    printf("%s,%c,%d,%d,", c[0].name, c[0].sex, c[0].score[0], c[0].score[1]);
    printf("%s,%c,%d,%d\n", c[1].name, c[1].sex, c[1].score[0], c[1].score[1]);
}
```

程序运行后的结果是（　　）。

A．Zhao,m,85,90,Sun,m,98,99

B．Zhao,m,85,90,Qian,f,95,92

C．Qian,f,95,92,Sun,m,98,99

D．Qian,f,95,92,Zhao,m,85,90

【答案】D

【解析】函数 f 的功能是将重新定义的结构体 b 整体赋给 a[1]，即数组 a 的第二个元素，而数组 a 的第一个元素不变。因此 main 函数中调用函数 f 时，即数组 c 的第二个元素变成结构体 b，第一个元素不变。答案选择 D 选项。

77．有以下程序：

```
#include <stdio.h>
#include <string.h>
typedef struct stu
{
    char name[10];
    char gender;
    int score;
}STU;
void f(STU *c)
{
    strcpy(c->name,"Qian");
    c->gender='f';
    c->score=350;
}
main()
{
    STU a={"Zhao",'m',290},b;
    b=a;
    f(&b);
    printf("%s,%c,%d,", a.name, a.gender, a.score);
    printf("%s,%c,%d\n", b.name, b.gender, b.score);
}
```

程序运行后的输出结果是（　　）。

A．Zhao,m,290,Qian,f,350

B．Zhao,m,290,Qian,m,290

C．Zhao,m,290,Zhao,m,290

D．Zhao,m,290,Qian,m,350

【答案】A

【解析】main 函数中，首先定义两个结构体 STU 类型的变量 a，b，并对 a 完成初始化，然后将 a 赋值给 b，使得结构体 a，b 的值都是{"Zhao", 'm', 290}，再调用 f 函数。f 函数的功能是将形参结构体指针 c 指向的结构体赋值为{"Qian", 'f', 350}，由于形参是指针，C 指向的结构体就是 b，所以实参 b 的值被修改为{"Qian", 'f', 350}，而 a 的值不变，答案选择 A 选项。

78．若有以下程序

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
typedef struct stu
{
   char *name,gender;
   int score;
}STU;
void f(char *p)
{
   p=(char *)malloc(10);
   strcpy(p,"Qian");
}
main()
{
   STU a={NULL,'m',290},b;
   a.name=(char *)malloc(10);
   strcpy(a.name,"Zhao");
   b=a;
   f(b.name);
   b.gender='f';
   b.score=350;
   printf("%s,%c,%d,", a.name, a.gender, a.score);
   printf("%s,%c,%d\n", b.name, b.gender, b.score);
}
```

则程序的输出结果是（　　）。

A. Qian,m,290,Qian,f,350

B. Zhao,m,290,Qian,f,350

C. Qian,f,350,Qian,t,350

D. Zhao,m,290,Zhao,f,350

【答案】D

【解析】初始化时，struct a 中的数据：a.name:Zhao，a.gender:m，a.score:290。执行 b=a，则 b 中的数据：b.name:Zhao，b.gender:m，b.score:290，struct 在赋值过程中实现的是浅拷贝，a.name 和 b.name 指向同一块内存空间。执行 f(b.name)，函数 f 中形参 p 和 b.name 指向同一块内存空间，在 f 函数内部，p 又重新指向另外一块分配的内存空间，而 b.name 的指向没有变化，指向的内存空间的值也没有发生变化，函数 f 返回后，b.name 中的值依然是 Zhao。答案选择 D 选项。

79. 有以下程序：

```
#include <stdio.h>
#include <string.h>
typedef struct stu
{
   char name[10];
   char gender;
   int score;
}STU;
void f(char *name,char gender,int score)
{
   strcpy(name,"Qian");
   gender='f';
   score=350;
}
main()
{
   STU a={"Zhao",'m',290},b;
   b=a;
   f(b.name,b.gender,b.score);
   printf("%s,%c,%d,", a.name, a.gender, a.score);
   printf("%s,%c,%d\n", b.name, b.gender, b.score);
}
```

程序的运行结果是（　　）。

A．Zhao,m,290,Qian,m,290

B．Zhao,m,290,Zhao,m,290

C．Zhao,m,290,Qian,m,350

D．Zhao,m,290,Qian,f,350

【答案】A

【解析】程序执行过程为：将 a 赋给 b，调用 f 函数，将 b 中 name 地址与 gender 和 score 的值传进函数，执行函数体，strcpy 函数改变 b 中 name 值，但由于后两个成员 gender 和 score 为值传递，不能改变 b 中 gender 和 score 值。调用函数结束后，结构体 a= {"Zhao", 'm', 290}，b= {"Qian', 'm', 290}。答案选择 A 选项。

80．有以下程序：

```
#include <stdio.h>
#include <string.h>
typedef struct stu
{
   char name[10];
   char gender;
   int score;
}STU;
void f(char *name,char *gender,int *score)
{
   strcpy(name,"Qian");
   *gender='f';
   *score=350;
}
main()
{
   STU a={"Zhao",'m',290},b;
   b=a;
   f(b.name,&b.gender,&b.score);
   printf("%s,%c,%d,", a.name, a.gender, a.score);
   printf("%s,%c,%d\n", b.name, b.gender, b.score);
}
```

程序运行后的输出结果是（　　）。

A．Zhao,m,290,Qian,f,350

B．Qian,m,290,Zhao,m,290

C．Zhao,m,290,Zhao,m,290

D．Zhao,m,290,Qian,m,290

【答案】A

【解析】main 函数中，首先定义两个结构体 stu 类型的变量 a，b，把 a 成员信息复制给 b 成员后，再调用函数 f，在 f 函数中，利用指针重新对 b 成员信息进行赋值，分别输出 a，b 成员信息。答案选择 A 选项。

81．若有以下程序

```
#include <stdio.h>
typedef struct stu
{
   char name[10],gender;
   int score;
} STU;
void f(STU a, STU b)
{
   b = a;
   printf("%s,%c,%d,", b.name, b.gender, b.score);
}
main()
{
   STU a={"Zhao", 'm', 290}, b={"Qian", 'f', 350};
   f(a,b);
   printf("%s,%c,%d\n", b.name, b.gender, b.score);
}
```

则程序的输出结果是（　　）。

A．Zhao,m,290,Qian,f,350

B．Zhao,m,290,Zhao,m,290

C．Qian,f,350,Qian,f,350

D．Zhao,m,290,Zhao,f,350

【答案】A

【解析】当把结构体变量中的数据作为一个整体传送给相应的形参时，传递的是实参结构体变量中的值，系统将为结构体类型形参开辟相应的存储单元，并将实参中各成员的值一一对应赋给形参中的成员，函数中形参结构体变量的改变不会影响到实参结构体变量。所以，在函数 f 中对 a，b 的操作不会影响 main 函数中 a，b 的值。答案选择 A 选项。

82．有以下程序：

```
#include <stdio.h>
typedef struct stu
{
   char name[10];
   char gender;
   int score;
} STU;
void f(STU a, STU *b)
{
   *b = a;
   printf("%s,%c,%d,", b->name, b->gender, b->score);
}
main()
{
   STU a={"Zhao", 'm', 290}, b={"Qian", 'f', 350};
   f(a,&b);
   printf("%s,%c,%d\n", b.name, b.gender, b.score);
}
```

程序运行后的输出结果是（　　）。

A．Zhao,m,290,Qian,f,350

B．Qian,m,290,Zhao,m,290

C．Qian,f,350,Qian,f,350

D．Zhao,m,290,Zhao,m,290

【答案】D

【解析】考察结构体和结构体成员的引用。main 函数中，首先为结构变量 a，b 初始化，再调用函数 f，在 f 函数中，使用结构体变量 a 对指针 b 指向的结构体进行赋值，使得指针 b 指向的结构体成员依次赋值为结构体 a 的成员，然后依次输出指针 b 指向的结构体成员的值；由于 f 函数形参 b 为指针，所以指针 b 指向的值被修改为 a 的同时，main 函数中的实参 b 的值也被修改成 a。答案选择 D 选项。

83．有以下程序：

```
#include <stdio.h>
typedef struct stu
{
    char name[10];
    char gender;
    int score;
} STU;
void f(STU a, STU *b)
{
    a = *b;
    printf("%s,%c,%d,", a.name, a.gender, a.score);
}
main()
{
    STU a={"Zhao", 'm', 290}, b={"Qian", 'f', 350};
    f(a,&b);
    printf("%s,%c,%d\n", a.name, a.gender, a.score);
}
```

程序运行后的输出结果是（　　）。

A．Qian,f,350,Qian,f,350

B．Zhao,m,290,Qian,f,350

C．Qian,f,350,Zhao,m,290

D．Zhao,m,290,Zhao,m,290

【答案】C

【解析】f 函数的功能：用形参结构体指针 b 指向的结构体对形参 a 赋值，此时形参结构体 a 的值为指针 b 指向的值，输出 a 即输出指针 b 指向的结构体；注意在函数 f 中形参 a 是值传递，对形参的操作不会影响实参，因此返回到 main 函数后，实参 a 的值没有改变，仍为初始化时的值。答案选择 C 选项。

84．有以下程序：

```
#include <stdio.h>
typedef struct stu
{
   char name[10];
   char gender;
   int score;
} STU;
void f(STU *a, STU *b)
{
   *b = *a;
   printf("%s,%c,%d,", b->name, b->gender, b->score);
}
main()
{
   STU a={"Zhao", 'm', 290}, b={"Qian", 'f', 350};
   f(&a,&b);
   printf("%s,%c,%d\n", b.name, b.gender, b.score);
}
```

程序的运行结果是（　　）。

A. Zhao,m,290,Zhao,m,290

B. Zhao,in,290,Qian,f,350

C. Qian,f,350,Qian,f,350

D. Qian,f,350,Zhao,m,290

【答案】A

【解析】调用函数时，若采用传引用的方式，那么对形参进行操作，实参也会产生相应的变化。程序执行过程为：调用 f 函数，将结构体 a 与 b 的地址作为参数传入函数，函数中指针 a 指向结构体 a，指针 b 指向结构体 b，将指针 a 指向的结构体 a 赋给指针 b 指向的结构体 b，结构体 b 内容被修改。输出 b 指向的结构体 b 的元素 Zhao，m，290。调用函数结束实参 b 的成员值也发生改变，输出结构体 b 的元素 Zhao，m，290。答案选择 A 选项。

85. 有以下程序

```
#include <stdio.h>
struct stu
{
   int num;
   char name[10];
   int age;
};
void fun(struct stu *p)
{
   printf("%s\n", p->name);
}
main()
{
   struct stu x[3] = {{01,"Zhang",20}, {02,"Wang",19}, {03,"Zhao",18}};
   fun(x+2);
}
```

程序运行后的输出结果是（　　）。

A．Zhang
B．Zhao
C．Wang
D．19
【答案】B
【解析】程序定义了结构体 stu 和结构体数组 x，其中 x 就是数组首地址，即&x[0]，x+2 代表了指向第三个元素的指针，即&x[2]，所以输出 p->name 为 Zhao。答案选择 B 选项。

86．有以下程序：

```
#include <stdio.h>
struct STU
{
    int num;
    float TotalScore;
};
void f(struct STU p)
{
    struct STU s[2] = {{20044,550}, {20045,537}};
    p.num = s[1].num;
    p.TotalScore = s[1].TotalScore;
}
main()
{
    struct STU s[2] = {{20041,703}, {20042,580}};
    f(s[0]);
    printf("%d %3.0f\n", s[0].num, s[0].TotalScore);
}
```

程序运行后的输出结果是（　　）。
A．20045 537
B．20044 550
C．20042 580
D．20041 703
【答案】D
【解析】把结构体变量作为一个参数传递给函数时，传递的是实参结构体变量的值，改变形参变量的内容对实参不会有任何影响。因此本题的函数 f 对结构体 s[0]没做改动，结果还是 20041 703。答案选择 D 选项。

87．有以下程序：

```
#include <stdio.h>
#include <string.h>
typedef struct
{
   char name[10];
   char sex;
   int age;
}STU;
void fun(STU *t)
{
   strcpy((*t).name,"Tong");
   (*t).age++;
}
main()
{
   STU s[2] = {"Hua", 'm', 18, "Qin", 'f', 19};
   fun(s+1);
   printf("%s,%d,%s,%d\n", s[0].name, s[0].age, s[1].name, s[1].age);
}
```

程序运行后的输出结果是（　　）。

A. Hua,18,Tong,20

B. Hua,18,Qin,19

C. Tong,19,Qin,19

D. Hua,19,Tong,19

【答案】A

【解析】程序执行过程为：定义 STU 类型数组，长度为 2，并初始化：s[0].name ="Hua"，s[0].sex = 'm'，s[0].age = 18，s[1].name = "Qin"，s[1].sex = 'f'，s[1].age = 19。调用函数 fun(s+1)将变量 s[1]地址传入函数，因此对形参操作就等价于对实参进行操作，函数执行后 s[1].name = "Tong"，s[1].age = 20。答案选择 A 选项。

88. 有以下程序：

```
#include <stdio.h>
#include <string.h>
typedef struct
{
   char name[10];
   char sex;
   int age;
}STU;
void fun(STU t)
{
   strcpy(t.name,"Tong");
   t.age++;
}
main()
{
   STU s[2] = {"Hua", 'm', 18, "Qin", 'f', 19};
   fun(s[1]);
   printf("%s,%d,%s,%d\n", s[0].name, s[0].age, s[1].name, s[1].age);
}
```

程序运行后的输出结果是（　　）。

A．Hua,19,Tong,19

B．Hua,18,Tong,20

C．Tong,19,Qin,19

D．Hua,18,Qin,19

【答案】D

【解析】程序执行过程为：定义 STU 类型数组，长度为 2，并为其初始化。调用函数 fun(s[1])将变量 s[1]值传入函数，则 t.name= "Qin"，t.sex= 'f'，t.age = 19。调用函数 strcpy 使 t.name = "Tong"，t.age = 20，函数调用结束。注意值传递时对形参的操作不会改变实参的值，结构体数组中元素并没有发生改变，答案选择 D 选项。

89．有以下程序

```
#include<stdio.h>
#include<string.h>
struct A
{
   int a;
   char b[10];
   double c;
};
void f(struct A t);
main()
{
   struct A a={1001, "ZhangDa",1098.0};
   f(a);
   printf("%d,%s,%6.1f\n",a.a,a.b,a.c);
}
void f(struct A t)
{
   t.a=1002;
   strcpy(t.b, "ChangRong");
   t.c=1202.0;
}
```

程序运行后的输出结果是（　　）。

A．1001,ZhangDa,1098.0

B．1002,ChangRong,1202.0

C．1001,ChangRong,1098.0

D．1002,ZhangDa,1202.0

【答案】A

【解析】当把结构体变量中的数据作为一个整体传送给相应的形参时，传递的是实参结构体变量中的值，系统将为结构体类型形参开辟相应的存储单元，并将实参中各成员的值一一对应赋给形参中的成员。函数体内对形参结构体变量中任何成员的操作，都不会影响对应实参中成员的值，从而保证了调用函数中数据的安全，但这也限制了将运算结果返回给调用函数。函数 f 中的结构体变量 t 是一个形参，在函数中对其值的改变只在函数体内有效，不影响实参。即，主函数中调用函数 f 以后，结构体变量 a 的值没有改变。答案选择 A 选项。

90．有以下程序：

```
#include <stdio.h>
#include <string.h>
struct A
{
   int a;
   char b[10];
   double c;
};
struct A f(struct A t);
main()
{
   struct A a={1001,"ZhangDa",1098.0};
   a=f(a);
   printf("%d,%s,%6.1f\n",a.a,a.b,a.c);
}
struct A f(struct A t)
{
   t.a=1002;
   strcpy(t.b,"ChangRong");
   t.c=1202.0;
   return t;
}
```

程序运行后的输出结果是（　　）。

A．1001,ZhangDa,1098.0

B．1002,ZhangDa,1202.0

C．1001,ChangRong,1098.0

D．1002,ChangRong,1202.0

【答案】D

【解析】函数 f 对结构体成员进行修改，并返回新的结构体；main 函数先定义了一个结构体变量 a 并为它赋初值，然后调用函数 f 修改结构体变量的成员值，最后输出新的结构体变量成员值。答案选择 D 选项。

91．有以下程序

```
#include <stdio.h>
struct S
{
   int n;
   int a[20];
};
void f(struct S *p)
{
   int i,j,t;
   for(i=0;i<p->n-1;i++)
      for(j=i+1;j<p->n;j++)
         if(p->a[i]>p->a[j])
         {
            t=p->a[i];
            p->a[i]=p->a[j];
            p->a[j]=t;
         }
}
main()
{
   int i;
   struct S s = {10, {2,3,1,6,8,7,5,4,10,9}};
   f(&s);
   for(i=0;i<s.n;i++) printf("%d,", s.a[i]);
}
```

程序运行后的输出结果是（　　）。

A. 1,2,3,4,5,6,7,8,9,10,

B. 10,9,8,7,6,5,4,3,2,1,

C. 2,3,1,6,8,7,5,4,10,9,

D. 10,9,8,7,6,1,2,3,4,5,

【答案】A

【解析】将结构体变量的地址作为实参传递，函数调用可以修改实参结构体中成员的值。函数 f 的作用是，把 p 指向的 struct 中的数组 a 的元素按照从小到大的方式排序。答案选择 A 选项。

92．有以下程序

```
#include <stdio.h>
struct S
{
   int n;
   int a[20];
};
void f(int *a, int n)
{
   int i;
   for(i=0;i<n-1;i++)
   a[i]+=i;
}
main()
{
   int i;
   struct S s = {10,{2,3,1,6,8,7,5,4,10,9}};
   f(s.a,s.n);
   for(i=0;i<s.n; i++) printf("%d,", s.a[i]);
}
```

程序运行后的输出结果是（　　）。

A. 1,2,3,6,8,7,5,4,10,9,

B. 3,4,2,7,9,8,6,5,11,10,

C. 2,3,1,6,8,7,5,4,10,9,

D. 2,4,3,9,12,12,11,11,18,9

【答案】D

【解析】将数组首地址作为实参传递，函数调用可以修改实参数组中元素的值。f 函数将数组中前 9 个元素的值加上对应的下标值。答案选择 D 选项。

93. 有如下程序：

```
#include <stdio.h>
struct pair
{
   int first,second;
};
struct pair get_min_max(int*array, int len)
{
   int i;
   struct pair res;
   res.first=array[0];
   res.second=array[0];
   for(i=1;i<len;i++)
   {
      if(array[i]<res.first)
         res.first=array[i];
      if(array[i]>res.second)
         res.second=array[i];
   }
   return res;
}
main()
{
   int array[5]={9,1,3,4};
   struct pair min_max = get_min_max(array,5);
   printf("min=%d,max=%d\n", min_max.first, min_max.second);
}
```

程序运行后的输出结果是（　　）。

A. min=1,max=9

B. min=0,max=9

C. min=1,max=4

D. min=0,max=4

【答案】B

【解析】在对数组进行初始化时，如果在定义数组时给出了长度，但没有给所有的元素赋予初始值，那么 c 语言将自动对余下的元素赋初值 0，则 array[5] = {9,1,3,4,0}。程序的执行过程为：调用函数 get_min_max(array,5)，将数组 array 的首地址传入函数，定义结构体变量 res，并为其成员赋值。for 循环查找数组 array 的最小值 0，将其赋值给 res 的成员 first，查找数组最大值 9，并将其赋值给 res 的成员 second。最后返回结构体变量 res，则 min_max=res。输出 min_max.first=0，min_max.second=9。答案选择 B 选项。

94. 有如下程序：

```
#include<stdio.h>
#include<string.h>
struct S
{
   char name[10];
};
void change(struct S *data,int value)
{
   strcpy(data->name, "****");
   value=13;
}
main()
{
   struct S input;
   int num = 4;
   strcpy(input.name, "THIS");
   change(&input,num);
   printf("%s,%d\n",input.name,num);
}
```

程序运行后的输出结果是（　　）。

A. ****,4

B. ****,13

C. THIS,4

D. THIS,13

【答案】A

【解析】程序执行过程为：定义结构体变量 input，调用 strcpy(input.name, "THIS")，使 input.name = "THIS"，调用函数 change(&input,4)，将结构体地址传入函数，执行函数体，调用 strcpy(data->name, "****")使 input.name = "****"，而 4 是值传递，并没有改变 num 的值。答案选择 A 选项。

95．有以下程序：

```
#include <stdio.h>
#include <string.h>
struct S
{
  char name[10];
};
void change(struct S *data, int value)
{
    strcpy(data->name,"#");
    value = 6;
}
main()
{
    struct S input;
    int num = 3;
    strcpy(input.name,"OK");
    change(&input,num);
    printf("%s,%d\n", input.name, num);
}
```

程序运行后的输出结果是（　　）。

A．OK,6

B．#,6

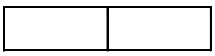
C．OK,3

D．#,3

【答案】D

【解析】考察结构体成员的引用规则。声明结构类型 S 中有一个字符数组 name，在 main 函数中，定义了一个结构变量 input，为 input 中成员 name 初始化“OK”后，调用 change 函数。chang 函数执行功能是把“#”赋值给 name，由于 change 函数的形参为结构体指针，因此 change 函数改变形参指针 data 指向的 name 值的同时，也改变了实参 input 的成员 name 值；而形参 value 为整型变量，是 num 的一个副本，修改了 nnm 的副本，对 num 本身不产生影响，所以输出为#,3。答案选择 D 选项。

96．为了建立如图所示的存储结构（即每个结点含两个域，data 是数据域，next 是指向结点的指针域）：




则在下面结构体定义中划线处应填入的选项是（　　）。

```
struct link
{
    char data;
    _____;
}node;
```

A．link*next

B．link  next

C．struct link*next

D．struct link next

【答案】C

【解析】结构体中的成员可以是各种类型的指针变量。当一个结构体中有一个或多个成员的基类型是本结构体类型时，称为“引用自身的结构体”。题目中 next 指针指向 struct 自身结点。答案选择 C 选项。

97．若有以下定义和语句：

```
struct st
{
   int n;
   struct st*next;
};
struct st a[3] = {5,&a[0], 6,&a[1],7,&a[2]}, *p;
p = &a[0];
```

则值为 6 的表达式是（提示：运算符->的优先级高于++）（　　）。

A．(*p).n++

B．p++->n

C．p->n++

D．(++p)->n

【答案】D

【解析】定义指向结构体变量的指针 p，并将结构体数组首地址赋给 p。执行(++p)->n，p 指针加一指向数组第二个元素 a[1]，a[1].n=6，D 项正确。A 项，(*p).n++，*p 为结构体数组第一个元素 a[0]，a[0].n=5，先取值，表达式为 5。B 项，p++->n，p 指向结构体数组第一个元素 a[0]，a[0].n=5。C 项，p->n++，p 指向结构体数组第一个元素 a[0]，a[0].n=5，由于++是后缀，先取值，所以表达式为 5，之后再加一。答案选择 D 选项。

98．若有以下程序段

```
struct st
{
   int n;
   struct st*next;
};
struct st a[3] = {5,&a[1],7,&a[2],9,'\0'}, *p;
p = &a[0];
```

则以下选项中值为 6 的表达式是（　　）。

A．p->n++

B．(*p).n

C．++(p->n)

D．p->n

【答案】C

【解析】定义长度为 3 的 struct 数组 a，a 中每个元素的 next 指向下一个元素，实际上数组 a 的元素构成了一条单链表，指针 p 指向单链表的头部，p->n 等于 5，p->next 指向 a[1]。ABD 三项返回的都是 p->n 的值 5。答案选择 C 选项。

99．有以下程序：

```
#include <stdio.h>
struct link
{
   int data;
   struct link *next;
};
main()
{
   struct link *h,a,b;
   h=&a;
   a.data=10;
   a.next = &b;
   b.data = 20;
}
```

程序运行时不能输出 10，20 的语句是（　　）。

A. printf("%d,%d\n", h->data, a.next.data);

B. printf("%d,%d\n", a.data, (*a.next).data);

C. printf("%d,%d\n", h->data, (*a.next).data);

D. printf("%d,%d\n", a.data, a.next->data);

【答案】A

【解析】可用以下 3 种形式来引用结构体变量中的成员：①结构体变量名.成员名；②指针变量名->成员名；③(*指针变量名).成员名。主函数定义了两个结构体变量 a，b，其成员 data 分别为 10 和 20，且 a 成员指针指向 b。A 选项中 a.next 为指针，其引用格式为 a.next->data 与(*a.next).data，A 项错误。B 项 a.data=10，(*a.next).data=20，能正确输出 10，20；C 项 h->data=10，(*a.next).data=20，能正确输出 10，20；D 项 a.data=10，a.next->data=20，能正确输出 10，20。答案选择 A 选项。

100．有以下程序：

```
#include <stdio.h>
main()
{
   struct node
   {
      int n;
      struct node *next;
   } *p;
   struct node x[3] = {{2,x+1}, {4,x+2}, {6,NULL}};
   p=x;
   printf("%d,",p->n);
   printf("%d\n",p->next->n);
}
```

程序运行后的输出结果是（　　）。

A. 2,3

B. 2,4

C. 3,4

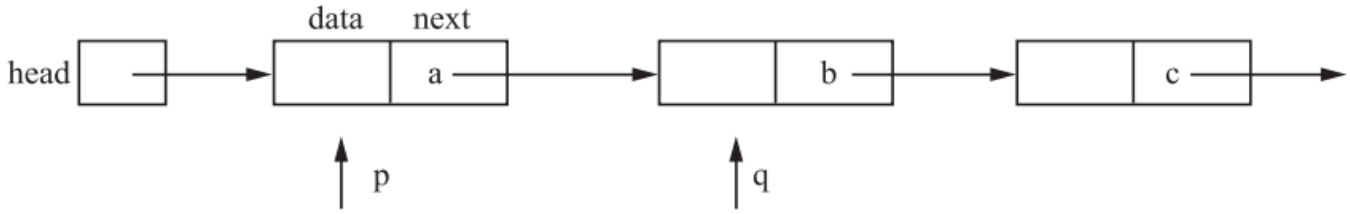
D. 4,6

【答案】B

【解析】程序中定义了一个含有三个结构体 node 元素的数组，数组中元素的 next 指针分别指向后一个元素，最后一个元素的 next 指针置为 NULL。p 指针最初指向 x[0]，因此 p->n=x[0].n=2；即第一次输出 2，p->next 指

向 x[1]，则 p->next->n=x[1].n=4。答案选择 B 选项。

101．假定已建立以下数据链表结构，且指针 p 和 q 已指向如下图所示的结点：




则以下选项中可将 q 所指结点从链表中删除并释放该结点的语句是（　　）。

A．(*)p.next=(*q).next;free(p);

B．b=q->next;free(q);

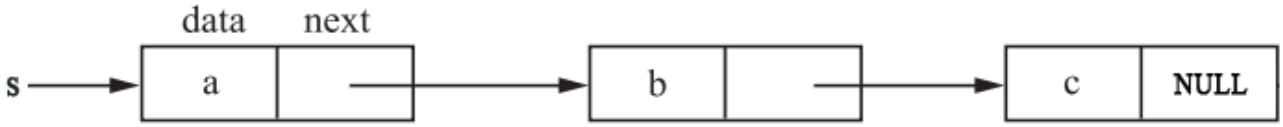
C．p=q;free(q);

D．p->next=q->next;free(q);

【答案】D

【解析】要删除结点 q，首先要将 q 的上一个结点 P 的指针域指向 q 的指针域所指向的结点，防止删除 q 结点后丢失 q 结点后的链表，即 p->next=q->next，然后释放结点 q，free(q);。答案选择 D 选项。

102．程序中已构成如下图所示的不带头结点的单向链表结构，指针变量 s、p、q 均已正确定义，并用于指向链表结点，指针变量 s 总是作为头指针指向链表的第一个结点。




若有以下程序段：

```
q=s;
s=s->next;
p=s;
while (p->next) p=p->next;
p->next=q;
q->next=NULL;
```

该程序段实现的功能是（　　）。

A．删除首结点

B．尾结点成为首结点

C．首结点成为尾结点

D．删除尾结点

【答案】C

【解析】s 指向 a 结点（链表头），所以语句 q=s;使 q 也指向 a 结点。s=s->next;语句使 s 指向 a 的下一个结点 b。p=s;语句使 p 也指向 b 结点。接下来是一个 while 循环，循环条件为 p->next，即当 p 没有指向链表尾时执行循环体 p=p->next;让 p 指向下一个结点，所以当循环结束时，p 指向链表尾（c 结点）。执行到此时，s、p、q 分别指向的结点是 b、c、a。然后执行 p->next=q;，使 c 结点的后续指针指向了 a 结点。最后执行 q->next=NULL;，使 a 结点的后续指针为 NULL。由此可见，该程序段实现了将 a 结点移动到 c 结点之后，即首结点成为尾结点。答案选择 C 选项。

## 二、填空题

人员的记录由编号和出生年、月、日组成，N 名人员的数据已在主函数中存入结构体数组 std 中，且编号唯一。函数 fun 的功能是：找出指定编号人员的数据，作为函数值返回，由主函数输出，若指定编号不存在，返回数据中的编号为空串。

请在程序的下画线处填入正确的内容并把下画线删除，使程序得出正确的结果。

注意：部分源程序给出如下。不得增行或删行，也不得更改程序的结构！

试题程序如下：

```
#include <stdio.h>
#include <string.h>
#define N 8
typedef struct
{
   char num[10];
   int year,month,day ;
}STU;

/**********found**********/
①_____ fun(STU *std,char *num)
{
   int i;
   STU a={"",9999,99,99};
   for (i=0; i<N; i++)
      /**********found**********/
      if(strcmp(②_____,num)==0)
         /**********found**********/
         return (③_____);
   return a;
}
void main()
{
   STU std[N]={{"111111",1984,2,15}, {"222222",1983,9,21}, {"333333",1984,9,1}, {"444444",1983,7,15},
{"555555",1984,9,28}, {"666666",1983,11,15}, {"777777",1983,6,22}, {"888888",1984,8,19}};
   STU p;
   char n[10]="666666";
   p=fun(std,n);
   if(p.num[0]==0)
      printf("\nNot found !\n");
   else
   {
      printf("\nSucceed !\n    ");
      printf("%s    %d-%d-%d\n",p.num,p.year,p.month,p.day);
   }
}
```

【答案】

①STU

②std[i].num

③std[i]

【解析】

填空 1：根据函数 fun 的返回值类型可知，函数类型标识符应该是结构体类型的，所以填入 STU。

填空 2：根据题目说明，找出指定编号人员，并将其数据返回。通过 strcmp 函数比较编号，若相同则函数值为 0，所以填入 std[i].num。

填空 3：由题目可知，假如编号对应，则返回其编号对应数据，所以填入数据 std[i]。

三、改错题

下列给定程序是建立一个带头结点的单向链表，并用随机函数为各结点赋值。函数 fun 的功能是将单向链表结点（不包括头结点）数据域为偶数的值累加起来，并且作为函数值返回。

请改正函数 fun 中的错误，使它能得出正确的结果。
注意：部分源程序在文件 MODI1.C 中，不要改动 main 函数，不得增行或删行，也不得更改程序的结构！
试题程序如下：

```
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
typedef struct aa
{
   int data;
   struct aa *next;
}NODE;
int fun(NODE *h)
{
   int sum=0;
   NODE *p;
   p=h->next;
   /*********found*********/
   while(p->next)
   {
      if(p->data%2==0)
         sum+=p->data;
      /*********found*********/
      p=h->next;
   }
   return sum;
}
NODE *creatlink(int n)
{
   NODE *h,*p,*s;
   int i;
   h=p=(NODE *)malloc(sizeof(NODE));
   for(i=1;i<n;i++)
   {
      s=(NODE *)malloc(sizeof(NODE));
      s->data=rand()%16;
      s->next=p->next;
      p->next=s;
      p=p->next;
   }
   p->next=NULL;
   return h;
}
```

```
outlink(NODE *h)
{
   NODE *p;
   p=h->next;
   printf("\n\nThe LIST:\n\n HEAD");
   while(p)
   {
      printf("->%d",p->data);
      p=p->next;
   }
   printf("\n");
}
main()
{
   NODE *head;
   int sum;
   system("CLS");
   head=creatlink(10);
   outlink(head);
   sum=fun(head);
   printf("\nSUM=%d\n",sum);
}
```

【答案】

（1）错误：while(p->next)

正确：while(p)或 while(p!=NULL)

（2）错误：p=h->next;

正确：p= p ->next;

【解析】

错误 1：执行 p=p->next 后，p 指针已经指向链表第一个包含数据域的结点。fun 函数的 while 循环判断当前指针 p 指向的结点是否存在，若存在则对该结点数据域进行判断操作，而不是判断 p 指针的指针域是否为空。

错误 2：fun 函数的 while 循环中判断结束后指针指向下一个结点，操作为 p=p->next。

四、设计题

学生的记录由学号和成绩组成，N 名学生的数据已在主函数中放入结构体数组 s 中，请编写函数 fun，它的功能是：按分数的高低排列学生的记录，高分在前。

注意：部分源程序存在文件 PROG1.C 中。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

试题程序如下：

```
#include <stdio.h>
#define N 16
typedef struct
{
   char num[10];
   int s;
}STREC;
int fun(STREC a[])
{
   STREC tmp;
   int i,j;
   for(i=0;i<N;i++)
      for(j=i+1;j<N;j++)
         {
            /*请按题目要求，完成一下代码*/

         }
}
main()
{
   STREC s[N]={
{"GA005",85},{"GA003",76},{"GA002",69},{"GA004",85},{"GA001",91},{"GA007",72},{"GA008",64},{"GA006",87},{"GA015",85},{"GA013",91},{"GA012",64},{"GA014",91},{"GA011",66},{"GA017",64},{"GA018",64},{"GA016",72}};
   int i;
   FILE *out;
   fun(s);
   printf("The data after sorted:\n");
   for(i=0;i<N;i++)
   {
      if((i)%4==0)
         printf("\n");
      printf("%s %4d ",s[i].num,s[i].s);
   }
   out=fopen("c:\\test\\out.dat","w");
   for(i=0;i<N;i++)
   {
      if((i)%4 == 0 && i)
         fprintf(out,"\n");
      fprintf(out,"%4d\n",s[i].s);
   }
   fprintf(out,"\n");
   fclose(out);
}
```

答：

```
if(a[i].s<a[j].s)
{
   tmp=a[i];
   a[i]=a[j];
   a[j]=tmp;
}
```

【解析】对 N 个数进行排序的算法很多，其中最简单的排序算法是冒泡算法。利用双层 for 循环嵌套和一个 if 判断语句来实现，外层循环用来控制需比较的轮数，内层循环用来控制两两比较。

# 第 10 章　对文件的输入输出

1．下列关于 C 语言文件的叙述中，正确的是（　　）。

A．文件由一系列数据依次排列组成，只能构成二进制文件

B．文件由结构序列组成，可以构成二进制文件或文本文件

C．文件由数据序列组成，可以构成二进制文件或文本文件

D．文件由字符序列组成，其类型只能是文本文件

【答案】C

【解析】C 语言将文件看作是一个字符（字节）的序列，即由一个一个字符（字节）数据顺序组成。根据数据的组成形式，可将文件分为两种：①ASCII 文件，又称文本（text）文件，它的每一个字节可放一个 ASCII 码，代表一个字符；②二进制文件，是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。所以 C 文件就是一个字节流或一个二进制流。答案选择 C 选项。

2．以下叙述中正确的是（　　）。

A．C 语言中的文件是流式文件，因此只能顺序存取数据

B．打开一个已存在的文件并进行了写操作后，原有文件中的全部数据必定被覆盖

C．在一个程序中当对文件进行了写操作后，必须先关闭该文件然后再打开，才能读到第 1 个数据

D．当对文件的读（写）操作完成之后，必须将它关闭，否则可能导致数据丢失

【答案】D

【解析】D 项正确，C 语言中读写文件是通过文件缓冲区完成的，在完成了对文件的操作之后，应当关闭文件，否则文件缓冲区中的剩余数据可能丢失。A 项错误，C 程序中的输入、输出文件都以数据流的形式存储在介质上，用顺序存取和直接存取两种方式；B 项错误，可以以追加的方式写文件；C 项错误，“r+”为读和写而打开文本文件，在读和写操作之间不必关闭文件，用 fseek()函数进行重新定位就能够读到第 1 个数据也不必关闭文件。答案选择 D 选项。

3．下面选项中关于“文件指针”概念的叙述正确的是（　　）。

A．文件指针就是文件位置指针，表示当前读写数据的位置

B．文件指针是程序中用 FILE 定义的指针变量

C．文件指针指向文件在计算机中的存储位置

D．把文件指针传给 fscanf 函数，就可以向文本文件中写入任意的字符

【答案】B

【解析】文件指针实际上是指向一个结构体类型的指针。B 项正确，结构体类型名为 FILE，用来定义文件指针，文件指针的定义形式为：FILE*　文件指针名。A 项错误，文件指针是指在程序中定义的 FILE 类型的变量，通过 fopen 函数调用给文件指针赋值，使文件指针和某个文件建立联系，C 程序中通过文件指针实现对文件的各种操作；文件位置指针只是一个形象化的概念，表示当前读或写的数据在文件中的位置；C 项错误，文件在计算机中的存储位置由操作系统负责，文件指针并没有指向文件的存储位置；D 项错误，fscanf 可以用于读文件，而非写文件。答案选择 B 选项。

4．以下叙述正确的是（　　）。

A．文件指针是指针类型的变量

B．在使用文件指针时，不需要在内存中为其分配空间

C．文件指针变量的值是文件的当前读取位置

D．调用 fscanf 函数能向所有类型的文件中写入任意字符

【答案】A

【解析】A 项正确，文件指针是指向文件类型的指针变量，文件指针的定义形式为：FILE*　文件指针名；B、C 项错误，文件指针指向的是文件缓冲区，而不是文件本身位置；D 项错误，fscanf 函数作用是从指定的文件中格式化读数据，读取数据类型由格式控制符决定。答案选择 A 选项。

5．以下选项中叙述正确的是（　　）。

A．文件指针是指针类型的变量

B. 文件指针可同时指向不同文件

C. 文件指针的值是文件在计算机磁盘中的路径信息

D. 调用 fscanf 函数可以向文本文件中写入任意字符

【答案】A

【解析】文件指针是指向文件类型的指针变量，文件指针的定义形式为：FILE* 文件指针名，A 项正确。一个指针在同一时间只能指向一个文件，B 项错误。文件指针是指针类型变量，存储的是文件缓存区首地址，而不是文件在计算机磁盘中的路径信息，C 项错误。fscanf 函数从从指定的文件中格式化读数据，而不是向文本文件中写入数据，D 项错误。答案选择 A 选项。

6. 以下叙述正确的是（　　）。

A. 在 C 语言中调用 fopen 函数就可把程序中要读、写的文件与磁盘上实际的数据文件联系起来

B. fopen 函数的调用形式为：fopen（文件名）

C. fopen 函数的返回值为 NULL 时，则成功打开指定的文件

D. fopen 函数的返回值必须赋给一个任意类型的指针变量

【答案】A

【解析】A 项正确，C 语言中打开一个文件通过系统函数 fopen 实现，通过这个函数把程序中要读、写的文件与磁盘上实际的数据文件联系起来；B 项错误，函数调用中缺少参数“操作方法”，其调用的一般形式为：文件指针名=fopen（文件名,使用文件方式）；C 项错误，fopen 函数返回一个指向指定文件的文件指针，如果不能实现打开指定文件的操作，则返回一个空指针 NULL；D 项错误，fopen 函数的返回值必须赋给一个文件类型的指针变量。

7. 设文件指针 fp 已定义，执行语句 fp=fopen("file","w")后，以下针对文本文件 file 操作的叙述中正确的是（　　）。

A. 写操作结束后可以从头开始读

B. 只能写不能读

C. 可以在原有内容后追加写

D. 可以随意读和写

【答案】B

【解析】用“w”方式打开的文件只能用于向该文件写数据，而不能用于向计算机输入。如果指定的文件不存在，系统将用在 fopen 调用中指定的文件名建立一个新文件；如果指定的文件已存在，则将从文件的起始位置开始写，文件中原有的内容将全部消失。答案选择 B 选项。

8. 有以下程序：

```
#include <stdio.h>
main()
{
   FILE *f;
   f=fopen("filea.txt","w");
   fprintf(f,"abc");
   fclose(f);
}
```

若文本文件 filea.txt 中原有内容为：hello，则运行以上程序后，文件 filea.txt 中的内容为（　　）。

A. helloabc

B. abelo

C. abc

D. abchello

【答案】C

【解析】fopen("filea.txt","w");表示以写的形式打开 filea.txt，fprintf(f,"abc");是先将文件清空再写入。本题将 filea.txt 内容清空后再写入 abc。答案选择 C 选项。

9．有以下程序段：

```
FILE *fp;
if((fp=fopen("test.txt","w")) == NULL)
{
   printf("不能打开文件!");
   exit(0);
}
else
   printf("成功打开文件!");
```

若指定文件 test.txt 不存在，且无其他异常，则以下叙述错误的是（　　）。

A．输出“不能打开文件!”

B．输出“成功打开文件!”

C．系统将按指定文件名新建文件

D．系统将为写操作建立文本文件

【答案】A

【解析】fopen 函数以一定方式打开指定文件，返回一个指向文件的文件指针，如果不能实现打开指定文件的操作，则返回一个空指针 NULL。如果指定文件不存在则创建一个文件名为指定文件名的新文件，然后打开它。程序中，文件 test.txt 不存在，但无其他异常，表示可以建立新文件，命名为 test.txt，C 项正确，并以只写方式打开它，D 项正确，返回指向文件的指针，if 条件不成立，输出“成功打开文件!”，B 项正确。答案选择 A 选项。

10．有以下程序段：

```
FILE *fp;
if((fp=fopen("test.txt","w"))==NULL)
{
   printf("不能打开文件!");
   exit(0);
}
else
   printf("成功打开文件!");
```

若文件 test.txt 已存在，则以下叙述正确的是（　　）。

A．程序运行后，文件 test.txt 中的原有内容将全部消失

B．程序运行时，会因文件存在而出错

C．对文件 test.txt 进行写操作后，可以随机进行读取

D．对文件 test.txt 写入的内容总是被添加到文件尾部

【答案】A

【解析】fopen 函数以一定方式打开指定文件，返回一个指向文件的文件指针，如果不能实现打开指定文件的操作，则返回一个空指针 NULL。如果指定文件不存在，则创建一个文件名为指定文件名的新文件，然后打开它。在指定文件有错误或者指定文件不存在却不能创建新文件的情况下，打开文件操作错误，返回空指针。本题程序中，文件 test.txt 已存在，以“w”方式打开文件时，文件 test.txt 中的原有内容将全部消失，A 选项正确。文件原本就存在，不会导致程序出错，会按照指定的方式打开文件，B 选项错误。对文件进行写操作，只能对指针指向的位置内容进行写操作，不能随机读写，C 选项错误。对文件 test.txt 写入的内容写到指针所指向的位置，而不是添加在文件尾部，D 选项错误。答案选择 A 选项。

11．以下程序：

```
#include <stdio.h>
main()
{
   FILE *fp;
   char str[10];
   fp=fopen("myfile.dat","w");
   fputs("abc",fp);
   fclose(fp);
   fp=fopen("myfile.dat","a+");
   fprintf(fp,"%d",28);
   rewind(fp);
   fscanf(fp,"%s",str);
   puts(str);
   fclose(fp);
}
```

程序运行后的输出结果是（　　）。

A．abc

B．28c

C．abc28

D．因类型不一致而出错

【答案】C

【解析】程序一开始以只写方式(“w”)打开文件 myfile.dat，此时，若文件不存在，系统将会建立名为 myfile.dat 的新文件；若文件已存在，将会清空文件内容后从文件头写入字符串"abc"，文件关闭后再以附加方式（“a+”）打开可读写文件，此时写入的数据"28"会被加到文件尾，然后把文件指针移到开头位置，再读入整个字符串到 str 中，最后输出 str 的值。答案选择 C 选项。

12．有以下程序：

```
#include <stdio.h>
main()
{
   FILE *fp;
   int a[10]={1,2,3},i,n;
   fp=fopen("d1.dat","w");
   for(i=0;i<3;i++) fprintf(fp,"%d",a[i]);
   fprintf(fp,"\n");
   fclose(fp);
   fp=fopen("d1.dat","r");
   fscanf(fp,"%d",&n);
   fclose(fp);
   printf("%d\n",n);
}
```

程序的运行结果是（　　）。

A．12300

B．123

C．1

D．321

【答案】B

【解析】程序开始定义了一个指针文件 FILE *fp。语句 fopen("d1.dat", "w");打开文件 d1.dat，然后 for 循环

语句向文件中依次写入数据。此时 d1.dat 应该为"123"，fopen("d1.dat", "r");以只读方式打开 d1.dat 文件，fscanf(fp, "%d", &n);读取一个整数，但这时它把 123 作为一个整数读进来，最后输出 n 的值为 123。答案选择 B 选项。

13．以下关于 fclose(fp)函数的叙述正确的是（　　）。

A．当程序中对文件的所有写操作完成之后，必须调用 fclose(fp)函数关闭文件

B．当程序中对文件的所有写操作完成之后，不一定要调用 fclose(fp)函数关闭文件

C．只有对文件进行输入操作之后，才需要调用 fclose(fp)函数关闭文件

D．只有对文件进行输出操作之后，才能调用 fclose(fp)函数关闭文件

【答案】A

【解析】程序编写者应该在程序终止之前关闭所有文件，如果不关闭，文件将会丢失数据。用 fclose 函数关闭文件，它先把缓冲区中的数据输出到磁盘文件，然后才释放文件指针变量，A 项正确，B 项错误。只要对文件进行操作后，都要调用 fclose 文件关闭文件，C、D 项错误。答案选择 A 选项。

14．以下叙述中错误的是（　　）。

A．gets 函数用于从终端读入字符串

B．getchar 函数用于从磁盘文件读入字符

C．fputs 函数用于把字符串输出到文件

D．fwrite 函数用于以二进制形式输出数据到文件

【答案】B

【解析】fgetc 函数从磁盘文件中读入字符，getchar 函数从终端或者键盘接收字符。答案选择 B 选项。

15．有如下程序：

```
#include <stdio.h>
main()
{
   int i;
   FILE* fp;
   for(i=0; i<5;i++)
   {
      fp = fopen("output.txt","w");
      fputc('A' + i, fp);
      fclose(fp);
   }
}
```

程序运行后，在当前目录下会生成一个 output.txt 文件，其内容是（　　）。

A．E

B．EOF

C．ABCDE

D．A

【答案】A

【解析】程序执行过程为：i=0 时，以只写方式打开一个文本文件 output.txt，调用函数 fputc 向文件输入 A，关闭文件；i=1 时，再次以只写方式打开 output.txt，调用函数 fputc 向文件输入 B 覆盖原本的 A，关闭文件；之后文件内的值依次为 C、D、E，当 i=4 时，文件内为 E，然后关闭文件；i=5 退出循环。output.txt 文件中内容为 E。答案选择 A 选项。

16．设 fp 为指向某二进制文件的指针，且已读到此文件末尾，则函数 feof(fp)的返回值为（　　）。

A．非 0 值

B．'\0'

C．0

D．NULL

【答案】A

【解析】本题考查的是文件指针 feof 的运用。当文件读到结尾时，feof(fp)为非零值，否则为 0。答案选择 A 选项。

17．以下叙述正确的是（　　）。

A．EOF 只能作为文本文件的结束标志，其值为-1

B．EOF 可以作为所有文件的结束标志

C．EOF 只能作为二进制文件的结束标志

D．任何文件都不能用 EOF 作为文件的结束标志

【答案】A

【解析】返回符 EOF（End of file）是在头文件 stdio.h 中定义的宏，表示文件的结束标志，值为-1，这种以 EOF 作为文件结束标志的文件，必须是文本文件。在文本文件中，数据都是以字符的 ASCII 代码值的形式存放，由于不可能出现-1，因此可以用 EOF 作为文件结束标志。答案选择 A 选项。

18．有以下程序

```
#include <stdio.h>
main()
{
   FILE *fp;
   int k,n,a[6]={1,2,3,4,5,6};
   fp=fopen("d2.dat","w");
   fprintf(fp,"%d%d%d\n",a[0],a[1],a[2]);
   fprintf(fp,"%d%d%d\n",a[3],a[4],a[5]);
   fclose(fp);
   fp=fopen("d2.dat","r");
   fscanf(fp,"%d%d",&k,&n);
   printf("%d%d\n",k,n);
   fclose(fp);
}
```

程序运行后的输出结果是（　　）。

A．123456

B．14

C．1234

D．12

【答案】A

【解析】将有 6 个元素的整型数组分两行输出到一个文件中，因为输出的都是数字并且每行都没有分隔符，所以当再对其进行读取操作时，每一行都会被认为是一个完整的数，而换行符则作为它们的分隔符。答案选择 A 选项。

19．有以下程序：

```
#include <stdio.h>
main()
{
   FILE *fp;
   int k,n,i,a[6]={1,2,3,4,5,6};
   fp=fopen("d2.dat","w");
   for(i=0;i<6;i++) fprintf(fp,"%d\n",a[i]);
   fclose(fp);
   fp=fopen("d2.dat","r");
   for(i=0;i<3;i++) fscanf(fp,"%d%d",&k,&n);
   fclose(fp);
   printf("%d,%d\n",k,n);
}
```

程序运行后的输出结果是（　　）。

A．1,2

B．3,4

C．5,6

D．123,456

【答案】C

【解析】程序中首先定义一个一维数组，然后将数组中的值写入到 d2.dat 中，然后再从 d2.dat 中读出数组中的元素，每次读出 2 个元素，分别赋值给 k 和 n，循环执行 3 次，故最终 k 和 n 的值为 5 和 6。答案选择 C 选项。

20．标准库函数 fgets(s,n,f)的功能是（　　）。

A．从文件 f 中读取长度不超过 n-1 的字符串存入指针 s 所指的内存

B．从文件 f 中读取长度为 n 的字符串存入指针 s 所指的内存

C．从文件 f 中读取 n 个字符串存入指针 s 所指的内存

D．从文件 f 中读取 n-1 个字符串存入指针 s 所指的内存

【答案】A

【解析】fgets 函数功能是从 f 所指文件中读入 n-1 个字符放入 s 为起始地址的空间内，并在尾端自动加一个结束标志“\0”。同时将读/写位置指针向前移动字符串长度个字节。在读出 n-1 个字符之前，如遇到了换行符或 EOF，则读出结束，A 项正确。B 项中“读取长度为 n”错误。C 项与 D 项中“读取 n/n-1 个字符串”错误，读取的是 n-1 个字符。答案选择 A 选项。

21．以下不能对文件进行输出的库函数是（　　）。

A．fwrite

B．fputs

C．fpout

D．fprintf

【答案】C

【解析】fwrite 函数的功能是用来向文件写数据块。fputs 函数的功能是用来向指定文件输出一个字符串。fprintf 函数按照格式向文本文件中输出数据。这三者都是库函数，而 fpout 不是库函数。答案选择 C 选项。

22．读取二进制文件的函数调用形式为“fread(buffer,size,count,fp);”，其中 buffer 代表的是（　　）。

A．一个内存块的首地址，代表读入数据存放的地址

B．一个整型变量，代表待读取的数据的字节数

C．一个文件指针，指向待读取的文件

D．一个内存块的字节数

【答案】A

【解析】“fread(void*buffer,size t_size,size t_count,FILE*stream);”功能是从一个文件流中读数据，读取 count

个元素，每个元素占 size 个字节，如果调用成功返回 count，出错或读到文件末尾时返回的记录数小于 count，也可能返回 0。buffer：用于接收数据的内存地址，大小至少是 size*count 个字节；size：单个元素的大小，单位是字节；count：元素的个数，每个元素占 size 个字节；stream：输入流。答案选择 A 选项。

23．有如下定义：

```
struct st
{
   int a;
   float b;
}x[10];
FILE *fp;
```

若文件已正确打开，且数组 x 的 10 个元素均已赋值，以下将数组元素写到文件中的语句错误的是（　　）。

A．for(i=0; i<10; i++) fwrite(x,sizeof(struct st), 1,fp);

B．fwrite(x,10*sizeof(struct st), 1,fp);

C．fwrite(x,sizeof(struct st), 10,fp);

D．for(i=0; i<10; i++) fwrite(&x[i],sizeof(struct st), 1,fp);

【答案】A

【解析】A 项中，因为函数 fwrite 中第三个参数为 1，即每次写入 1 个结构体数据，x 是数组的首地址，因此，每次写入的数据都是数组的首个结构体元素，没有将整个数组写入文件中去。答案选择 A 选项。

24．有以下程序

```
#include <stdio.h>
main()
{
   FILE *fp;
   int a[10]={1,2,3,0,0},i;
   fp=fopen("d2.dat","wb");
   fwrite(a,sizeof(int),5,fp);
   fwrite(a,sizeof(int),5,fp);
   fclose(fp);
   fp=fopen("d2.dat","rb");
   fread(a,sizeof(int),10,fp);
   fclose(fp);
   for(i=0;i<10;i++) printf("%d,",a[i]);
}
```

程序的运行结果是（　　）。

A．1,2,3,0,0,0,0,0,0,0,

B．1,2,3,1,2,3,0,0,0,0,

C．123,0,0,0,0,123,0,0,0,0,

D．1,2,3,0,0,1,2,3,0,0,

【答案】D

【解析】首先用函数 fopen 以“wb”的方式（wb 只写打开或新建一个二进制文件，只允许写数据）打开文件 d2.dat，然后调用两次 fwrite 函数将数组 a 的 5 个元素，依次输出到文件 fp 中，共 10 个字节，关闭文件。再次打开文件，使用文件指针指向文件的开头，调用 fread 函数从文件中读取这 l0 个字节的数据到数组 a 中。答案选择 D 选项。

25．有以下程序：

```
#include <stdio.h>
main()
{
   FILE *fp;
   int i,a[6]={1,2,3,4,5,6},k;
   fp=fopen("data.dat","w+b");
   fwrite(&a[0],sizeof(int),1,fp);
   for(i = 1;i < 6;i++)
   {
      fseek(fp,0L,0);
      fread(&k,sizeof(int),1,fp);
      fseek(fp,0L,0);
      a[i]+=k;
      fwrite(&a[i],sizeof(int),1,fp);
   }
   rewind(fp);
   fread(&k,sizeof(int),1,fp);
   fclose(fp);
   printf("%d\n",k);
}
```

程序的运行结果是（　　）。

A．21

B．6

C．123456

D．11

【答案】A

【解析】“w+”表示打开可读写文件，若文件存在则文件长度清为零，即该文件内容会消失；若文件不存在则建立该文件；加入 b 字符后“w+b”用来告诉函数库打开的文件为二进制文件。程序执行过程为：以读/写方式打开一个新的二进制文件 data.dat，从地址为 a 的数据块开始，一次输出一个整型字节的数据，只输出一次，将 1 写入文件中；执行 for 循环，将文件指针移到文件开头，将文件第一个数值读出赋给 k，再将指针移动到开头，向文件内输入 a[i]+k=3。for 循环实现将 a 数组中元素累加，结果 21 存放在文件中。调用 rewind 将文件指针移动到开头，调用 fread 函数从文件中读出一个整型数据赋给 k=21，调用 fclose 函数关闭文件，打印 k 值，答案选择 A 选项。

26．函数 rewind(fp)的作用是（　　）。

A．函数 rewind(fp)的作用是使文件读写指针指向文件开始位置

B．使文件位置指针指向文件的末尾

C．使文件位置指针移至前一个字符的位置

D．使文件位置指针移至下一个字符的位置

【答案】A

【解析】rewind 函数作用是使文件读写指针指向文件开始位置。答案选择 A 选项。

27．有以下程序：

```
#include<stdio.h>
main()
{
   FILE*fp;
   int i,a[6]={1,2,3,4,5,6};
   fp=fopen("d2.dat","w+");
   for(i=0;i<6;i++) fprintf(fp,"%d\n", a[i]);
   rewind(fp);
   for(i=0;i<6;i++) fscanf(fp,"%d", &a[5-i]);
   fclose(fp);
   for(i=0;i<6;i++) printf("%d,", a[i]);
}
```

程序运行后的结果是（　　）。

A. 4,5,6,1,2,3,

B. 1,2,3,3,2,1,

C. 1,2,3,4,5,6,

D. 6,5,4,3,2,1,

【答案】D

【解析】函数的功能是将数组 a 中的元素顺序写入文件，再将文件中的元素倒序读出存储到数组 a 中，最后输出数组 a 的元素，答案选择 D 选项。

28. 以下函数不能用于向文件写入数据的是（　　）。

A. ftell

B. fwrite

C. fputc

D. fprintf

【答案】A

【解析】函数 ftell 用于得到文件位置指针当前位置相对于文件首的偏移字节数。在随机方式存取文件时，由于文件位置频繁地前后移动，程序不容易确定文件的当前位置。调用函数 ftell 就能非常容易地确定文件的当前位置。A 项不能写入数据，BCD 三项都可以向文件中写入数据。答案选择 A 选项。

29. 有以下程序：

```
#include<stdio.h>
main()
{
   FILE*pf;
   char*s1="China",s2="Beijing";
   pf=fopen("abc.dat","wb+");
   fwrite(s2,7,1,pf);
   rewind(pf); /*文件位置指针回到文件开头*/
   fwrite(s1,5,1,pf);
   fclose(pf);
}
```

以上程序执行后 abc.dat 文件的内容是（　　）。

A. China

B. Chinang

C. ChinaBeijings

D. BeijingChina

【答案】B

【解析】pf 是一个文件指针，fopen("abc.dat","wb+");执行后，pf 指向可读写的二进制文件 abc.dat。语句 fwrite(s2,7,1,pf);是将 s2 的前 7*1 个字符的内容写入 pf 中，即 Beijing。rewind(pf);是将文件位置指针移回到文件开头，语句 fwrite(s1,5,1,pf);是从文件的开头位置，将 s1 的前 5*1 个字符的内容写入，替换掉原来位置上的内容，所以结果为 Chinang。答案选择 B 选项。

30．有以下程序：

```
#include <stdio.h>
main()
{
   FILE *fp;
   int i,a[6]={1,2,3,4,5,6},k;
   fp=fopen("data.dat","w+b");
   fprintf(fp,"%d\n",a[0]);
   for(i = 1;i < 6;i++)
   {
      fseek(fp,0L,0);
      fscanf(fp,"%d",&k);
      fseek(fp,0L,0);
      fprintf(fp,"%d\n",a[i]+=k);
   }
   rewind(fp);
   fscanf(fp,"%d",&k);
   fclose(fp);
   printf("%d\n",k);
}
```

程序的运行结果是（　　）。

A．21

B．6

C．123456

D．11

【答案】A

【解析】程序执行过程为：以读/写方式建立一个新的文本文件 data.dat，将 l 写入文件；执行 for 循环，将文件指针移到文件开头，将文件第一个数值 1 赋给 k，再将指针移动到开头，向文件内输入 a[i]+k=3。for 循环实现将 a 数组中元素累加，结果 21 存放在文件中。调用 rewind 将文件指针移动到开头，调用 fscanf 函数从文件中读出数值赋予 k=21，调用 fclose 函数关闭文件，打印 k 值。答案选择 A 选项。

31．有以下程序：

```
#include <stdio.h>
main()
{
   FILE *fp;
   int i,a[6]={1,2,3,4,5,6},k;
   fp=fopen("data.dat","w+b");
   fprintf(fp,"%d\n",a[0]);
   for(i = 1;i < 6;i++)
   {
      rewind(fp);
      fprintf(fp,"%d\n",a[i]);
   }
   rewind(fp);
   fscanf(fp,"%d",&k);
   fclose(fp);
   printf("%d\n",k);
}
```

程序运行后的输出结果是（　　）。

A．6

B．21

C．123456

D．654321

【答案】A

【解析】本题首先定义文件指针变量 fp 和数组 a[ ]，再打开文件 data.dat，随后先给文件写入数据 a[0]，rewind 函数将文件指针从当前位置重新指向文件开始位置，所以 for 循环依次将数组 a 中的数据写入文件开始位置，退出循环后，文件中的数据顺序为：654321，重新使指针指向文件开始位置，将此时 fp 指向的数据（即文件中第一个数据）写入变量 k 中，关闭文件，输出 k 值，答案选择 A 选项。

## 二、填空题

请根据以下各小题的要求设计 C 应用程序（包括界面和代码）。

请补充 main 函数，该函数的功能是：先以只写方式打开文件“out52.dat”，再把字符串 str 中的字符保存到这个磁盘文件中。

注意：部分源程序给出如下。

请勿改动主函数 main 和其他函数中的任何内容，仅在 main 函数的横线上填入所编写的若干表达式或语句。

试题程序如下：

```
#include <stdio.h>
#define N 80
main()
{
   FILE*fp;
   int i=0;
   char ch;
   char str[N]="I'm a student!";
   if((fp=fopen(①_____))==NULL)
   {
      printf("cannot open out52.dat\n");
      exit(0);
   }
   while(str[i])
   {
      ch=str[i];
      ②_____;
      putchar(ch);
      i++;
   }
   ③_____;
}
```

【答案】

①"out52.dat","w"

②fputc(ch,fp)

③fclose(fp)

【解析】

填空 1：根据题目的要求需要将文件以“只写”的方式打开，注意文件名是一个字符串，必须加双引号，打开方式也要加双引号。

填空 2：while 循环的功能是把字符串中的字符保存在磁盘文件中，把单个字符写入文件型指针变量 fp 所指的文件的表达式是 fputc(ch,fp)。

填空 3：一定要注意对文件操作完后要关闭文件，所以此处应该填 fclose(fp)。

三、设计题

请编写一个函数 fun，它的功能是：将 ss 所指字符串中所有下标为奇数位置的字母转换为大写（若该位置上不是字母，则不转换）。

例如，若输入“abc4Efg”，则应输出“aBc4EFg”。

注意：部分源程序在文件 PROG1.C 中。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

试题程序如下：

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
void fun(char *ss)
{

}
main()
{
   FILE *wf;
   char tt[81],ss[81]="abcEfg";
   system("CLS");
   printf("\nPlease enter a string within 80 charasters:\n");
   gets(tt);
   printf("\n\nAfter changeing, the string\n %s",tt);
   fun(tt);
   printf("\nbecomes\n %s\n",tt);
   /*******************/
   wf=fopen("out.dat","w");
   fun(ss);
   fprintf(wf,"%s",ss);
   fclose(wf);
   /*******************/
}
```

答：

```
void fun(char *ss)
{
   int i;
   if(ss==NULL)return;
   for(i=0;ss[i]!='\0';i++)
   {
      if(i%2==1&&ss[i]>='a'&&ss[i]<='z')
         ss[i]=ss[i]-32;
   }
}
```

【解析】题目要求将给定字符串中奇数位置的字母转换为大写，需要先判断奇数位置以及此位置的字符是否是小写字母，如果是再通过其转换方法进行转换。C 语言中，只要将小写字母减去 32 即转成大写字母，程序用 if 语句实现转换功能。

# 第 11 章　常见错误分析

说明：本章内容重点将初学者在学习和使用 C 语言时容易犯的错误列举出来，这些内容在之前各章中大多已谈到，故此处不再整理习题。