

C 语言精选 200 题

1、C 语言源程序名的后缀是()。

A、.exe

B、.c

C、.obj

D、.cp

【解析】：本题的考查点是 C 语言源程序名的后缀。C 语言源程序名的后缀是 .c，对源程序进行编译后得到一个后缀为 .obj 的目标程序，然后把目标程序进行连接操作，则可得到一个后缀为 .exe 的可执行文件。故本题答案为 B。

2、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

【解析】：本题的考查点是 C 程序的基本概念。我们把由高级语言编写的程序称为“源程序”，由二进制代码表示的程序称为“目标程序”，由汇编语言编写的程序称为“汇编程序”。对于计算机本身来说，它并不能直接识别由高级语言编写的程序，只能识别和执行由 0 和 1 组成的二进制指令。为了把源程序转换成机器能接受的目标程序，软件工作者编制了一系列软件，通过这些软件可以把用户按规定语法写出的语句一一翻译成二进制的机器指令，这种具有翻译功能的软件称为“编译程序”。然后将编译后的目标程序与系统的函数库和其他目标程序连接起来，形成可执行程序由计算机直接执行。故本题答案为 D。

3、C 语言主要是借助以下()功能来实现程序模块化。

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

【解析】：本题的考查点是模块化结构。C 语言是一种结构化程序设计语言。它提供了定义“函数”的功能，在 C 语言中没有子程序的概念，它提供的函数可以完成子程序的所有功能；C 语言允许对函数单独进行编译，从而可以实现模块化。故本题答案为 A。

4、关于程序设计基本概念，以下叙述错误的是()。

- A、结构化算法可以解决任何复杂的问题
- 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、计算机可以直接执行由任意高级语言编写的程序

【解析】：本题考查的是程序设计。对于计算机本身来说，它并不能直接识别由高级语言编写的程序，只能接受和处理由 0 和 1 的代码构成的二进制指令或数据。故本题答案为 D。

5、以下()是不合法的 C 语言常量。”

- A、0xFF
- B、1.2e0.5
- C、2L
- D、'72'

【解析】：C 语言中的指数形式，字母 E 之前必须有数字，且 E 后面指数必须为整数。

本题答案为 B。

6、假定 X 和 Y 为 double 型，则表达式 X=2,Y=X+3/2 的值是()。

- A、3.500000
- B、3
- C、2.000000
- D、3.000000

【解析】：本题考查 C 语言数据类型。本题中 X、Y 为双精度浮点型数，3/2 为整型相除结果为 1，所以本题结果为 3.000000。本题答案为 D。

7、阅读以下程序：

```
main()
{
    int x;
    scanf("%d",&x);
    if(x--<5)printf("%d\n",x);
```

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

程序运行后，如果从键盘上输入 5，则输出结果是()。

- A、3
- B、4
- C、5
- D、6

【解析】：本题考查 C 语言中的自增(++)与自减(--)运算符。c 语言中“++”与“-”运算符有前置与后置两种形式，前置时变量先自增一或自减一后再参与表达式的运算，后置时变量先参与表达式的运算后再自增一或自减一。本题答案为 B。

8、以下选项中合法的用户标识符是()。

- A、long
- B、_2Test
- C、3Dmax
- D、A.dat

【解析】：本题的考查点是用户标识符。选项 A 是 C 语言中的关键字，不能作为标识符；选项 C 中数字不能打头；选项 D 中“.”不能作为标识符的一部分，选项 D 错；本题答案为 B。

9、已有定义：int x=3, y=4, z=5;，则表达式!(x+y)+z-1&& y+z/2 的值是()。

- A、6
- B、0
- C、2
- D、1

【解析】：本题的考查点是运算符的优先级。在本题中，运算符的优先级分别为：“!” > “/” > “+/-” > “&&”，即：!(3+4)+5-1&&4+5/2=0+5-1&&4+2=4&&6=1。故本题答案为 D。

10、以下选项中，与 k=n++完全等价的表达式是()。

- A、k=n, n=n+1
- B、n=n+1, k=n
- C、k=++n
- D、k+=n+1

【解析】：本题的考查点是自增及赋值运算。n++是自增运算，是先使用，然后再使 n 加 1，分开来写即为：k=n, n=n+1。本题答案为 A。

11、在一个 C 语言程序中()。

- A、main 函数必须出现在所有函数之前
- B、main 函数可以在任何地方出现
- C、main 函数必须出现在所有函数之后
- D、main 函数必须出现在固定位置

【解析】：本题的考查点是 main 函数。一个 C 程序中必须有且只能有一个用“main”命名的主函数，其它函数由用户自行命名。main 函数可以在任何地方出现。本题答案为 B。

12、有以下程序

```
main()
{
    int a, b, d=25;
    a=d/10%9;
    b=a&&(-1);
    printf("%d, %d\n", a, b);
}
```

程序运行后的输出结果是()。

- A、6, 1
- B、2, 1
- C、6, 0
- D、2, 0

【解析】：本题考置的是算术运算符。如果算术运算符“/”中参与运算的变量都是整型变量，则“/”表示整除运算，“%”表示求余。本题中“a=d/10%9;”的值为 $25/10\%9=2$ ；“b=a&&(-1);”为 $2\&\&(-1)=1$ (注意：-1 表示真，只有 0 才表示假)，所以 a, b 的值分别为 2, 1。本题答案为 B。

13、以下程序的功能是：给 r 输入数据后计算半径为 r 的圆面积 s，程序在编译时出错。

```
main()
/*Beginning*/
```

```
{int r;float s;
scanf("%d",&r);
s=Π*r*r;printf("s=%f\n",s) ;
}
```

出错的原因是()。

- A、注释语句书写位置错误
- B、存放圆半径的变量 r 不应该定义为整型
- C、输出语句中格式描述符非法
- D、计算圆面积的赋值语句中使用了非法变量

【解析】： 本题考查的是 C 程序的编写。本题并未对变量“ Π ”作任何定义，所以编译时会出现非法字符的错误信息。本题答案为 D。

14、以下叙述中错误的是()。

- A、C 程序中的#include 和#define 行均不是 C 语句
- B、除逗号运算符外，赋值运算符的优先级最低
- C、C 程序中，j++；不是赋值语句
- D、C 程序中，+、-、*、/、%号是算术运算符，可用于整型和实型数的运算

【解析】： 本题考查的是 C 语言运算符。“%”是求余运算符，其两侧的数值只能为整型，所以它不能用于实型的运算。本题答案为 D。

15、下列叙述错误的是()。

- A、一个 C 语言程序只能实现一种算法
- B、C 程序可以由多个程序文件组成
- C、C 程序可以由一个或多个函数组成
- D、一个 C 函数可以单独作为一个 C 程序文件存在

【解析】： 本题考查的是 C 程序的构成。一个 C 语言程序可以实现多种算法。一个 C 程序可以由一个或多个源程序文件组成，一个源程序文件由一个或多个函数组成，构成 C 语言程序的基本单位是函数，函数也是 C 程序中模块化程序设计的基础，C 函数可分为标准库函数和用户定义函数两类。本题答案为 A。

16、以下选项中不属于 C 语言程序运算符的是()。

- A、<>
- B、sizeof
- C、()
- D、&&

【解析】：本题考查的是运算符。c 语言中 sizeof:求字节运算符，():函数调用，&&:逻辑与。<>不是运算符，<是小于号，>是大于号。本题答案为 A。

17、C 语言中，最基本的数据类型包括()。

- A、整型、实型、字符型
- B、整型、字符型、数组
- C、整型、实型、逻辑型
- D、整型、实型、结构体

【解析】：本题考查的是数据类型。数据类型分为基本类型、构造类型、指针类型、空类型。基本类型包括整型、字符型、实型（单精度型、双精度型）、枚举度型；构造类型包括数组类型、结构体类型、共用体类型。本题答案为 A。

18、请选出合法的 C 语言赋值语句()。

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

【解析】：本题的考查点是赋值语句。选项 A 没有用分号表示语句结束，不合法;选项 B 为 i=i+1 ;这并不是赋值语句；选项 D，函数名和 c 语言的关键字重名，不合法。本题答案为 C。

19、若想给已定义为 int 型的变量 a, b, c, d 赋值为 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;

【解析】：本题考查的是赋值语句。逗号隔开的赋值语句从左往右执行，A 选项中只有 d 的值为 1，a、b、c 并未赋值。本题答案为 A。

20、以下选项中错误的是()。

- A、`printf("%s\n",'s');`
- B、`printf("%d %c\n",'s','s');`
- C、`printf("%c\n",'s'-32);`
- D、`printf("%c\n",65);`

【解析】：本题的考查知识点是输出语句。A 选项中格式字符为 s，应当输出一个字符串，而 's' 是一个字符，选项 A 错误。所以本题答案选 A。

21、以下有关 scanf 函数的叙述中错误的是()。

- A、在 scanf 函数中的格式控制字符串不会输出到屏幕上
- B、scanf 函数有返回值，其值就是本次调用 scanf 函数时正确读入的数据项个数
- C、scanf 函数从输入字符流中按照格式控制指定的格式解析出相应数据，送到指定地址中
- D、在 scanf 函数的格式字符前既可以加入正整数指定输入数据所占的宽度，也可以对实数指定小数位的宽度

【解析】：在 scanf 函数的格式字符前可以加入一个正整数指定输入数据所占的宽度，但不可以对实数指定小数位的宽度。本题答案为 D。

22、若执行下面程序时从键盘上输入 5，

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

```
}
```

则输出是()。

A、7

B、6

C、5

D、4

【解析】：本题的考查点是 if 条件的判断。根据 C 语言的语法，x+是在使用 x 之后，再将 x 的值加 1，在 if 语句中，x 的值为 5，条件不成立，执行 else 后面的语句，因为 x 的值已经加 1，为 6，所以打印结果为 6。本题答案为 B。

23、设 ch 是 char 型变量，其值为 A，且有下列的表达式

```
ch=(ch>='A' && ch<='Z')?(ch+32):ch
```

上面表达式的值是()。

A、A

B、a

C、Z

D、z

【解析】：本题的考查点是条件运算符。条件表达式的一般形式为：表达式 1?表达式 2:表达式 3，条件运算符的执行顺序：先求解表达式 1，若非 0 则求解表达式 2，此时表达式 2 的值就作为整个表达式的值，若表达式 1 的值为 0，则求解表达式 3，此时表达式 3 的值就作为整个表达式的值。本题中，表达式 1 的值为 1，所以整个表达式的值为表达式 (ch+32) 的值，即为 a。本题答案为 B。

24、有如下程序段

```
int a=14, b=15, x ;char c='A' ;
```

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

执行该程序段后，x 的值为()。

A. true

B. false

C、0

D、1

【解析】：c 语言中没有逻辑型数据，以 1 代表“真”，以 0 代表“假”。a&&b 的值为真，c<'B' 的值也为真，所以 (a&&b)&&(c<'B') 的值为真。所以 x 的值应为 1。本题答案为 D。

25、若执行以下程序时从键盘上输入 9，则输出结果是（ ）。

```
main()
{
    int n;
    scanf("%d:", &n);
    if(n++<10) printf("%d\n", n) ;
    else printf("%d\n", n--);
}
```

A、 11

B、 10

C、 9

D、 8

【解析】:n++是在语句结束后再加一,因此,if 语句中相当于判断的是 $9 < 10$,先执行“printf(“%d \n”, n);”,而这时,因为 if 语句被执行,n 的值也被加 1,所以在打印时,n 值为 10,而 else 语句此时并未执行。本题答案为 B。

26、以下程序的输出结果是（ ）。

```
main ()
{
    int a=4, b=5, c=0, d;
    d=!a&&!b||!c;
    printf("%d\n", d);
}
```

A、 1

B、 0

C、 非 0 的数

D、 -1

【解析】: !a 值为 0, !a&& ! b 的值为 0, !c 的值为 1, 而 $0 || 1$ 的值应该是 1。所以答案是 A。本题答案为 A。

27、下列叙述中正确的是（ ）。

A、 C 语言中既有逻辑类型也有集合类型

B、 C 语言中没有逻辑类型但有集合类型

C、C 语言中有逻辑类型但没有集合类型

D、C 语言中既没有逻辑类型也没有集合类型

【解析】：本题的考查点是 c 语言中的基本数据类型。数据类型是所允许的数据及其操作的集合，是高级语言的重要特征。程序设计中可以利用数据类型发现程序中的某些错误。

C 语言提供了三大类数据类型，即基本类型、复合类型和地址类型。本题答案为 D。

28、以下叙述中正确的是（ ）。

A、改变 if-else 语句的缩进格式，会改变程序的执行流程

B、if 语句只能嵌套一层

C、不能在 else 子句中再嵌套 if 语句

D、if 子句和 else 子句中可以是任意的合法的 C 语句

【解析】：本题考查的是 if 语句。if 语句中可以嵌套多层，格式为 if-else(if)，else 中也可以嵌套 if 语句，在 if 嵌套中书写可以采用缩进形式，便于程序的阅读，但若不缩进，也不会改变执行流程。本题答案为 D。

29、以下叙述中正确的是（ ）。

A、只能在循环体内和 switch 语句体内使用 break 语句

B、当 break 出现在循环体中的 switch 语句体内时，其作用是跳出该 switch 语句体，并中止循环体的执行

C、continue 语句的作用是：在执行完本次循环体中剩余语句后，中止循环

D、在 while 语句和 do-while 语句中无法使用 continue 语句

【解析】：本题考查的是 break、continue 语句。break 语句可以使流程跳出 switch 语句体，也可以用 break 语句在循环体内中终止本层循环体，从而提前结束本层循环；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断，可以理解为仅结束本次循环。break、continue 语句适用于各种循环语句中。本题答案为 A。

30、C 语言中，下列叙述正确的是()。

- A、不能使用 do-while 语句构成的循环
- B、do-while 语句构成的循环，必须用 break 语句才能退出
- C、do-while 语句构成的循环，当 while 语句中的表达式值为非零时结束循环
- D、do-while 语句构成的循环，当 while 语句中的表达式值为零时结束循环

【解析】：本题的考查点是 do-while 语句。选项 A 是错误的，c 语言支持 do-while 语句；do-while 构成的循环，当 while 语句中的表达式值为零时结束循环，而不是非零，选项 B、C 都是错误的。本题答案为 D。

31、以下程序的输出结果是()。

```
main()
{
    int n=4;
    while(n--)printf("%d",--n) ;
}
```

- A、 20
- B、 31
- C、 321
- D、 210

【解析】：本题的考查点是 while 语句的判断。n--是先使用后运算，--n 是先运算后使用。n=4 时，执行 n--后，n 变为 3，执行--n 后，n 变为 2，所以，第一次输出时，n 为 2；同上过程，第二次输出时，n 为 0。本题答案为 A。

32、下面函数的功能是()。

```
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
- D、将字符串 x 连接到字符串 y 后面

【解析】：本题的考查点是 while() 循环语句。这个函数的功能是求字符串的长度，需要注意的是，在返回长度时一定要减一，否则计算出的串长度将会多一。本题答案是 A。

33、有以下程序：

```
main ()
{
    int i, s=0;
    for(i=1;i<10;i+=2) s+=i+1;
    printf("%d\n",s) ;
}
```

程序执行后的输出结果是()。

- A、自然数 1 ~ 9 的累加和
- B、自然数 1 ~ 10 的累加和
- C、自然数 1 ~ 9 中奇数之和
- D、自然数 1 ~ 10 中偶数之和

【解析】：通过 for(i=1;i<10;i+=2) 可知，i 为从 1 开始的小于 10 的奇数，即 1, 3, 5, 7, 9, i+1 即为 2, 4, 6, 8, 10。也就是自然数 1 到 10 中的偶数，s+=i+1 就是累加这几个偶数。本题答案为 D。

34、有以下程序：

```
main()
{
    int i, n=0;
    for(i=2;i<5;i++)
```

```
{
    do
    {   if(i%3) continue;
        n++ ;
    }while(!i) ;
    n++;
}
printf("n=%d\n",n);
}
```

程序执行后输出结果是()。

A、n=5

B、n=2

C、n=3

D、n=4

【解析】：

当 i=2 时，i%3 为真，来到 while (!i) ;此时 n=0，!i 为假，结束 while 循环，执行 n++，此时 n 为 1。

当 i=3 时，i%3=0，此时 n++，n=2，!i 为假，结束 while 循环，执行 n++，此时 n 为 3。

当 i=4 时，i%3=1 为真，来到 while (!i) ，!i 为假，结束 while 循环，继续往下执行 n++，此时 n 为 4。

本题答案为 D。

35、有以下程序段

```
int n, t=1, s=0;
scanf("%d", &n) ;
do{s=s+t; t=t-2; }while (t!=n);
```

为使此程序段不陷入死循环，从键盘输入的数据应该是()。

A、任意正奇数

B、任意负偶数

C、任意正偶数

D、任意负奇数

【解析】：本题考查的是 while 循环语句。要使得程序段不陷入死循环，则必须存在终止循环的条件“ $t=n$ ”，由于 t 的初始值为 1，且每执行一次循环都会使得 t 的值减 2，执行本程序后 t 有可能的最大值为 -1，然后，依次减 2，所以键盘输入的 n 的值必须是 t 的值之一，即为任意负奇数。本题答案为 D。

36、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)

【解析】：本题考查的是 while 条件表达式的判断。while (表达式) 语句用来实现“当型”循环结构，当表达式为非 0 值时，执行 while 语句中的内嵌语句。选项 A 非零即为 1，选项 B 与选项 D 的结果也为 1，只有选项 C 为 0。本题答案为 C。

37、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, j=0; t[i]=s[j]; i++, j++);

C、do{*t++=*s++;}while(*s);

D、for(i=0, j=0; t[i++] = s[j++];);

【解析】：本题的考查点是字符串复制。复制字符串是将 s 字符串中(包括 '\0') 的字符逐个放到 t 所指的对应单元中。选项 C 中 do-while 循环，是先执行循环体后判断循环条件，当将 s 字符串中 '\0' 前的最后一个字符放到 t 所指的对应单元中后，再执行++运算，s 则指向 '\0'，然后判断循环条件不成立，退出循环，而 s 字符串中的 '\0' 没有复制到 t 所指的对应单元中，即复制字符串没有完成。选项 A、B、D 都能将 s 所指字符串正确复制到 t 所指存储空间。本题答案为 C。

38、关于“while(条件表达式)循环体”，以下叙述正确的是()。

- A、条件表达式的执行次数总是比循环体的执行次数多一次
- B、循环体的执行次数总是比条件表达式的执行次数多一次
- 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

【解析】：本题的考查点是 while(条件表达式)循环体。对于 while (条件表达式)循环体，我们是先判断，若满足条件表达式则执行循环体，若不满足则退出循环。当遇到不满足条件时，我们执行了条件表达式，并没有执行循环体。故条件表达式的执行次数总是比循环体的执行次数多一次。本题答案为 A。

39、设有说明语句：char a='\72';则变量 a ()。

- A、包含 1 个字符
- B、包含 2 个字符
- C、包含 3 个字符
- D、说明不合法

【解析】：转义字符常量 '\xx' 可以把 ' \ ' 后面的数字转换为对应的 ASCII 字符，'\72' 就是 ' : ' 本题答案为 A。

40、以下程序的输出结果是()。

```
main()
{   char c='z';
    printf("%c",c-25) ;
```

- A、a
- B、Z
- C、z-25
- D、y

【解析】：本题考查 C 语言的字符变量与整数的运算。C 语言中，可以将字符变量看成是一个整型变量，可以与整数运算，运算使用其 ASCII 值参与运算。本题答案为 A。

41、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、空串一定比空格打头的字符串小

【解析】： 本题考查的是字符串。

C 语言中没有字符串类型的常量和变量，只有字符型的常量和变量，我们可以通过定义字符型的数组或指针来得到字符串；对于字符的比较，不可以用关系运算符，而只能用字符串比较函数 `strcmp()` 进行比较。字符串比较函数的比较规则是对两个字符串从左至右逐个字符相比（按 ASCII 码值大小比较），直到出现不同的字符或遇到“\0”为止，对这两个字符串并没有长度的限制。无任何字符的字符串叫“空串”，用“”表示，空串的长度是零，它小于一切非空字符串。本题答案为 D。

42、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

【解析】： 本题的考查点是 `getchar()` 函数的作用。`getchar()` 函数的作用是从终端（或系统隐含指定的输入设备）输入一个字符，而不是从磁盘读入字符。本题答案为 B。

43、以下叙述中正确的是()。

- A、字符常需要用单号括起来
- B、字符常量在内存中占 2 个字节
- C、转义字符要用双引号括起来，以便与普通的字符常量区分开
- D、字符常量是不能进行字符关系运算的

【解析】： 本题考查的是字符常量。

在 C 语言中字符常量需要用单引号括起来；字符常量在内存中占一个字节；C 语言中定义了一些字母前加“\”来表示常见的那些不能显示的 ASCII 字符，如 \0, \t, \n 等，就称为转义字符；字符常量可以进行字符关系运算。本题答案为 A。

44、有如下程序

```
int func(int a, int b)
{ return(a+b); }

main()
{
    int x=2, y=5, z=8, r;
    r=func(func(x, y), z);
    printf("%d\n", r);
}
```

该程序的输出结果是()。

A、 12

B、 13

C、 14

D、 15

【解析】：调用函数 func(x, y)，返回 x+y，即 7；再调用函数 func(7, z)，返回 7+z=7+8，即 15。本题答案为 D。

45、下列叙述中正确的是()。

A、 C 语言编译时不检查语法

B、 C 语言的子程序有过程和函数两种

C、 C 语言的函数可以嵌套定义

D、 C 语言的函数可以嵌套调用

【解析】：选项 A 的错误在于编译过程中是检查语法的，若发现源程序有语法错误，则系统会提示出错信息；选项 B 的错误在于 C 语言中，子程序的作用是由函数来完成的，无过程的概念；选项 C 的错误在于函数不可以嵌套定义，但可以嵌套调用。本题答案为 D。

46、在调用函数时，如果实参是简单变量，它与对应形参之间的数据传递方式是（ ）。

- A、地址传递
- B、单向值传递，
- C、实参传给形，再由形参传回实参
- D、传递方式由用户指定

【解析】： 本题考查 C 语言函数的参数的传值方式。

C 语言函数中的参数传递方式是按值传递，指将实参的值复制一份传递给形参，形参的改变而不影响实参，即为单向传递。本题答案为 B。

47、以下函数值的类型是（ ）。

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

- A、int
- B、不确定
- C、void
- D、float

【解析】： 本题考查 C 语言函数的缺省的函数类型。C 语言中，如果函数前不加任何数据类型时，缺省函数的类型为整型，函数的类型就是函数返回值的类型。本题答案为 A。

48、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

【解析】：本题考查的是库函数。`h` 是头文件，头文件是包含函数声明和定义的文件，调用输入输出函数时，要求在源文件中包含头文件“`stdio.h`”。`#include` 是编译预处理指令，就是在编译前将 `stdio.h` 这个文件里的函数都添加到你写的 `cpp` 文件中，然后参与编译，生成 `.obj` 文件。本题答案为 A。

49、库函数 `rand()` 的功能是产生一个在 $0 \sim 32767$ 之间的随机数。若要用此函数随机产生一个 $0 \sim 99.99$ (2 位小数) 之间的数，以下能实现此要求的表达式是()。

- A、`(rand()%10000)/100.0`
- B、`(rand())% 10000)/100`
- C、`(rand()%9000+ 1000.0)/100.0`
- D、`(rand()%100)/100.0`

【解析】：`rand()%10000` 的取值范围为 $0 \sim 9999$ ，所以 `(rand())%10000)/100.0` 的取值范围为 $0 \sim 99.99$ 。本题答案为 A。

50、若 `x` 是整型变量，`pb` 是基类型为整型的指针变量，则正确的赋值表达式是()。

- A、`pb=&x;`
- B、`pb=x;`
- C、`*pb=&x;`
- D、`*pb=*x`

【解析】：本题的考查点是赋值表达式。

选项 A 是将变量 `x` 的地址赋给指针变量 `pb`，使 `pb` 指向 `x`，为正确的赋值表达式。

选项 B 是将 `x` 的值当作地址赋给指针 `pb`，不正确；

选项 C 是将 `x` 的地址赋给指针 `pb` 指向的内存单元，不正确；

选项 D 是不正确的语句；

本题答案为 A。

51、以下叙述中错误的是()。

- A、函数形参的值，不会改变对应实参的值
- B、函数可以返回地址值
- C、可以给指针变量赋一个整数作为地址值
- D、当在程序的开头包含头文件 `stdio.h` 时，可以给指针变量赋 `NULL`

【解析】：本题的考查点是函数间值的传递。

函数调用中发生的数据传送是单向的。即只能把实参的值传送给形参，而不能把形参的值反向地传送给实参。因此在函数调用过程中，形参的值发生改变，而实参中的值不会变化。指针变量表示的是变量的地址，而变量地址是内存区每一个字节的编号，而这个编号是在程序编译时自动给变量分配的，所以指针变量不能赋一个整数作为地址值。本题答案为 C。

52、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

【解析】：本题主要考查了指针变量的使用。定义了指针变量 `p`、`q`、`r`，其中 `p`、`q` 分别存放了变量 `m`、`n` 的地址，即 `p` 指向了 1 所在存储单元，`q` 指向了 2 所在的存储单元。当执行语句“`r=p;p=q;q=r;`”后，`p`、`q` 中存储的地址进行了交换，即 `p` 中存储的是 `n` 的地址，`q` 中存储的是 `m` 的地址，但 `m`、`n` 的值并未改变，

所以最终输出 1, 2, 2, 1。

本题答案为 B。

53、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'

【解析】： 本题主要考查了指针变量的运用。

“*b=*a”表示将指针变量 a 所指向的值赋给*b，即将 a 所指字符串复制到 b 所指内存空间。注意：“=”是赋值操作，而不是比较是否相等的符号“==”。本题答案为 A。

54、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*号是一个间址运算符
- B、定义语句中的*号只是一个说明符
- C、定义语句中的 p 只能存放 double 类型变量的地址
- 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

【解析】： 本题的考查点是指针变量的引用。

*号是地址运算符，也是一个间址运算符。&:取地址运算符。&a 为变量 a 的地址，*p 为指针变量 p 所指向的存储单元。因此*p=&a 是把变量 a 的地址作为初值赋给指针变量 p。指针变量 p 的基类型是 double，p 只能存放 double 类型变量的地址。本题答案为 B。

55、以下关于地址和指针的叙述中正确的是()。

- A、可以取变量的地址赋值给同类型的指针变量
- B、可以取常量的地址赋值给同类型的指针变量

C、可以取一个指针变量的地址赋给本指针变量,这样就使得指针变量指向自身

D、有指针变如果未赋初值,则自动赋空值 NULL

【解析】: 本题考查的是指针。

指针变量可以指向同类型的变量,但不能指向同类型的常量, B 错误; C 选项中可以取一个指针变量的地址赋给本指针变量,这样就使得本指针变量指向指针变量,故 C 错误;所有指针变量如果未赋初值,则指针没有指向,故 D 错误。本题答案为 A。

56、若有以下说明:

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

```
char c='a', d, g;
```

则值为 4 的表达式是()。

A、 $a['g' - c]$

B、 $a[4]$

C、 $a['d' - 'c']$

D、 $a['d' - c]$

【解析】: 本题的考查点是数组元素的引用。

数值为 4 实际上就是 $a[3]$, 而 'd' 的 ASCII 码和 'a' 的 ASCII 码刚好相差 3, 变量 c 的值为 'a', 所以 $'d' - 'c' = 3$ 。本题答案为 D。

57、当调用函数时,实参是一个数组名,则向函数传送的是()。

A、数组的长度

B、数组的首地址

C、数组每一个元素的地址

D、数组每个元素中的值

【解析】: 当调用函数时,实参是一个数组名,则向函数传送的是数组的首地址,函数中的形参可定义成以下三种形式:

(1)形参定义成数组;

(2)形参定义成可变长数组;

(3)形参定义为指针变量。

本题答案为 B。

58、以下定义语句中，错误的是()。

- A、`int a[]={1,2};`
- B、`char *a[3];`
- C、`char s[10]="test";`
- D、`int n=5,a[n];`

【解析】： 本题考查 C 语言数组的定义与初始化。

一维数组的定义方式为：类型说明符数组名[常量表达式]；其中，常量表达式中可以包括常量和符号常量，不能包括变量。也就是说，C 不允许对数组的大小作动态定义，即数组的大小不依赖于程序运行过程中变量的值。本题答案为 D。

59、假定 int 类型变量占用两个字节，若有定义：`int x[10]={0,2,4};`，则数组 x 在内存中所占字节数是()。

- A、3
- B、6
- C、10
- D、20

【解析】： 本题考查 C 语言的数组的定义。

C 语言中可以为数组赋初值，或部分赋初值。当数组定义后，系统就为其分配内存空间，不论其中有没有内容。本题虽然只给数组 x 赋了 3 个初值，但系统仍为数组 x 分配了 10 个内存空间。所以数组 x 在内存中所占

字节数为 $2 \times 10 = 20$ 。

本题答案为 D。

60、已有定义：`char a[]="xyz"`，`b[]={'x','y','z'}`；，以下叙述中正确的是()。

- A、数组 a 和 b 的长度相同
- B、a 数组长度小于 b 数组长度
- C、a 数组长度大于 b 数组长度
- D、上述说法都不对

【解析】：本题考查的是数组定义。

C 语言规定 ' \0 ' 为字符串结束标志，系统对字符串常量自动加一个 ' \0 ' 为结束符。所以 " char a[]=" xyz"" 的数组长度为 4，而 "b[]={ ' x', 'y', 'z' };" 的数组长度为 3（此处的数组长度与 strlen 函数所求的长度不同，此处的长度是指数组占内存空间的大小），故 a 数组长度大于 b 数组长度。故本题答案为 C。

61、有以下程序

```
main()
{
    int a[3][3], *p, i;
    p=&a[0][0];
    for(i=0; i<9; i++) p[i]=i;
    for(i=0; i<3; i++) printf("%d", a[1][i]);
}
```

程序运行后的输出结果是（ ）。

A、012

B、123

C、234

D、345

【解析】：本题考查的是指针变量对数组的引用。

本题通过 " for (i=0; i<9; i++) p[i]=i;" 对二维数组 a 的元素依次赋值。赋值后, a 的数组元素的值分别为 a[0][0]=0, a[0][1]=1, a[0][2]=2, a[1][0]=3, a[1][1]=4, a[1][2]=5, a[2][0]=6, a[2][1]=7, a[2][2]=8, 故本题输出的值为 345。本题答案为 D。

62、设有以下定义和语句

```
char str[20]="Program" , *p;
p=str;
```

则以下叙述中正确的是（ ）。

A、*p 与 str[0] 中的值相等

B、str 与 p 的类型完全相同

C、str 数组长度和 p 所指向的字符串长度相等

D、数组 str 中存放的内容和指针变量 p 中存放的内容相同

【解析】：本题考查的是指针变量对数组的引用。

语句 $p = \text{str}$ 表示指针变量 p 指向数组 $\text{str}[20]$ 的首地址也就是 $\text{str}[0]$ 。所以 *p 与 $\text{str}[0]$ 中的值相等。故本题答案为 A。

63、若有定义： $\text{int } w[3][5];$ ，则以下不能正确表示该数组元素的表达方式是()。

A、 $*(w+3)$

B、 $*(w+1)[4]$

C、 $*((w+1))$

D、 $*(&w[0][0]+1)$

【解析】：本题考查的是数组的表达方式。

选项 A： $*(w+3) = *((w+0)+3) = *(w[0]+3) = w[0][3]$ ，符合数组定义的范围可以正确表示数据元素。

选项 B： $*(w+1)[4] = *((w+1)+4) = *w[5] = w[5][0]$ ，超出了数据定义的范围。

选项 C： $*((w+1)) = *w[1] = w[1][0]$ ，符合数组定义的范围可以正确表示数据元素。

选项 D： $*(&w[0][0]+1); = *(w[0]+1) = w[0][1]$ ，符合数组定义的范围可以正确表示数据元素。

故本题答案为 B。

64、若有定义语句： $\text{int } a[3][6];$ ，按在内存中的存放顺序，a 数组的第 10 个元素是()。

A、 $a[0][4]$

B、 $a[1][3]$

C、 $a[0][3]$

D、 $a[1][4]$

【解析】：本题的考查点是二维数组的排列方式。

在一维存储器中存放二维数组，有两种方式：一种是按行排列，即放完一行

之后顺次放入第二行。另一种是按列排列，即放完一列之后再顺次放入第二列。在C语言中，二维数组是按行排列的。本题数组“a[3][6]”表示a数组存放了三行六列，第一行从a[0][0]到a[0][5]，第二行从a[1][0]到a[1][5]，第三行从a[2][0]到a[2][5]，所以a数组的第10个元素是a[1][3]。

故本题答案为B。

65、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};

【解析】：本题的考查点是数组的定义和引用。

C语言中多维数组赋初值时可以部分赋值，如果对全部元素赋初值，则定义数组时对第一维的长度可以不指定，但第二维的长度不能省，所以选项C错误。

66、以下叙述正确的是()。

- A、char c1, *c2, **c3, c4[2];是C语言的合法语句
- B、不能在数组说明符的一对方括号中使用表达式
- C、数组下标的最小值可以是负值
- D、若有数组定义int array[5];则语句printf("%d", array[1.23]);是合法的

【解析】：本题考查的是数组。

数组说明符的一对方括号内是整型常量表达式；C语言规定每个数组第一个元素的下标总是0，然后依次递增，不可以为赋值；数组说明符的一对方括号内的数值只可以是0和正整数。故本题答案为A。

67、指针s所指字符串的长度为()。

char*s="\t1Name\\Address\n";

A、19

B、15

C、18

D、说明不合法

【解析】：本题的考查点是字符串长度的计算。

观察题目，\t 表示的一个 TAB，长度为 1，1 表示的一个字符”1”，长度为 1，“Name” 的长度为 4，“\\” 代表的是一个”\” 符，长度为 1，“Address” 的长度为 7，“\n” 是一个换行符，长度为 1， 所以整个字符串的长度应当为 1+1+4+1+7+1=15。故本题答案为 B。

68、已定义以下函数，该函数的功能是()。

```
fun(char *p2, char *p1)
{ while((*p2=*p1)!=' \0' ) {p1++;p2++;}}
```

- A、将 p1 所指字符串复制到 p2 所指内存空间
- B、将 p1 所指字符串的地址赋给指针 p2
- C、对 p1 和 p2 两个指针所指字符串进行比较
- D、检查 p1 和 p2 两个指针所指字符串中是否有 '^ \0'

【解析】：本题的考查点是函数。

while((*p2=*p1)!=' \0' 表示将 p1 所指向的值赋给*p2，即将 p1 所指字符串复制到 p2 所指内存空间。在这里”*P2=*p1”是赋值操作，而不是比较是否相等，比较应当为双等号”==”。所以 B，C，D 都不对。故本题答案为 A。

69、以下程序段中，不能正确赋字符串(编译时系统会提示错误)的是()。

- A、char s[10]="abcdefg";
- B、char t[]="abcdef",*s=t;
- C、char s[10];s="abcdefg";
- D、char s[10];strcpy(s,"abcdefg");

【解析】：C 语言中不能给数组整体赋初值，所以选项 C 是不正确的。

70、有以下程序

```
void swap(char *x, char *y)
{   char t;
```

```
t=*x;*x=*y;*y=t;
main ()
{   char *s1="abc",*s2="123";
    swap(s1,s2);printf("%s, %s\n",s1,s2);
```

程序执行后的输出结果是()。

- A、123, abc
- B、abc, 123
- C、1bc, a23
- D、321 , cba

【解析】： 本题考查的是指针变量值的交换。

C 语言中，字符串常量在内存中是以字符数组的形式进行存放的，因此字符指针 x 和 y 指向的是各字符串的首地址，也就是字符串第一个字符的地址，则*x 与*y 交换的是字符串的第一个字符，即字符"a"与"1"的交换，而字符串中其它字符保持不变。故本题答案为 C。

71、以下程序运行后输出的结果是()。

```
main()
{   char s[]={"aeiou"}, *ps;
    ps=s;
    printf("%c\n", *ps+4);
}
```

- A、 a
- B、 e
- C、 u
- D、 元素 s[4]的地址

【解析】： 本题考查的是指针变量的运算。

此题中先定义一个指针变量 ps，并将 s 的首地址赋给此指针变量，*ps +4 相当于先从指针变量 ps 指向的地址中取出相应数据"a"后，对此数据加 4，a 的 ASCII 码为 32，32+4=36， 对应于字符 e。故本题答案选 B。

72、若有定义：char *x="abcdefghi";，以下选项中正确运用了 strcpy 函

数的是()。

- A、char y[10]; strcpy(y, x[4]);
- B、char y[10]; strcpy(++y, &x[1]);
- C、char y[10], *s; strcpy(s=y+5, x);
- D、char y[10], *s; strcpy(s=y+1, x+1);

【解析】： 本题考查的是用于字符串处理的函数。

选项 A 中 x [4] 是取字符 e，也就是将字符复制到 y 中，strcpy 实现的是地址的复制所以选项 A 错误；选项 B 中 ++y 是错误的，不允许对常量进行自加运算(y 是一个确定的地址值)，所以选项 B 错误；选项 C 指针变量 s 指向了 y 向后的第 5 位，则存放 x 时会出现越界问题，所以选项 C 错误；选项 D 中，指针变量 s 指向了 y 向后的第一位，此时 s 可存放数据的长度为 9，而地址：“x+1”起的字符串的地址长度也刚好为 9 (包括 “\0”)，所以开始复制不会出现地址越界问题。故本题答案为 D。

73、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

【解析】： 本题的考查点是字符串长度函数。

strlen (字符数组)是测试字符串的长度的函数。函数的值为字符串的实际长度，不包括 ‘\0’ 在内。C 语言中字符串是以 ‘\0’ 字符结束的，且 strlen() 函数计算的是 ‘\0’ 字符前的所有字符的个数，所以本题 strlen(s)的值是 7。需要与其区分的是 sizeof () 函数，它的功能是计算变量或数组的所分配到的内存空间的大小。如果本题要求计算 sizeof (s)的话，则其值应为 10。

故本题答案为 A。

74、下面是有关 C 语字符数组的描述，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

【解析】：本题主要考查了字符数组的概念。

用来存放字符数据的数组是字符数组。字符数组中的一个元素存放一个字符。所以选项 D 只能存放字符串的说法是错误的，应该是字符数组只能存放字符，而字符串是用字符数组来表示的。故本题答案为 D。

75、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);

【解析】：本题的考查点是字符串比较函数 strcmp。字符串比较函数"strcmp(char*s1, char*s2)" 的比较规则是对两个字符串自左至右逐个字符相比（按 ASCII 码值大小比较），直到出现不同的字符或遇到" \0" 为止。如果全部相同，则认为相等；若出现不相同的字符，则以第一个不相同的字符的比较结果为准。当 s1<s2， 返回值<0；当 s1=s2 时，返回值=0；当 s1>s2 时， 返回值>0。故本题答案为 D。

76、在函数调用过程中，如果函数 funA 调用了函数 funB，函数 funB 又调用了函数 funA，则（ ）。

- A、称为函数的直接递归调用
- B、称为函数的间接递归调用
- C、称为函数的循环调用
- D、C 语言中不允许这样的递归调用

【解析】：本题考的是函数的递归调用。

在调用一个函数的过程中又出现直接或间接地调用该函数本身，称为函数的递归调用；在调用 f1 函数过程中要调用 f2 函数，而在调用 f2 函数过程中又要调用 f1 函数是间接调用，所以本题是函数的间接递归调用。故本题答案为 B。

77、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

【解析】： 本题的考查点是函数的递归调用。

若执行调用语句” n=fun(3);”，那么执行 fun (3)时返回 fun(3-1)+1, 即 fun(2)+1;执行 fun(2)时返回 fun(2-1)+1, 即 fun(1)+1; 执行 fun(1)时返回值为 1。由以上分析可知，fun 函数被调用 3 次。故本题答案为 B。

78、以下程序执行后的输出结果是()。

```
#include <stdio.h>
void fun(int a)
{
    if( a>1 ) fun(a-1);
    printf("%d",a);
}
main()
{ int q=5; fun(q); printf("\n");}

```

A、12345

B、54321

C、012345

D、543210

【解析】： 本题的考查知识点是函数的递归调用。

fun 函数中 a=5, a>1 成立，再次调用 fun 函数，此时 fun 函数中|a=4, a>1

成立，再次调用 fun 函数，以此类推，直到 a=1，此时输出 1，再返回到上一级函数，输出 2，直到输出 5 为止，故选项 A 正确。所以本题的答案选 A。

79、以下叙述中正确的是（ ）。

- A、全局变量的作用域一定比局部变量的作用域范围大
- B、静态(static) 类别变量的生存期贯穿于整个程序的运行期间
- C、函数的形参都属于全局变量
- D、未在定义语句中赋初值的 auto 变量和 static 变量的初值都是随机值

【解析】： 本题的考查点是全局变量和局部变量。

在函数内定义的变量是局部变量，而在函数之外定义的变量称为外部变量，外部变量是全局变量。它的有效范围为：从定义变量的位置开始到该源文件结束。选项 A 说法不正确，全局变量的作用域还跟定义变量的位置有关；选项 C 也不正确，函数的形参是局部变量，其它函数不能调用；若在定义局部变量不赋初值，则对 static 变量来说，编译时自动赋初值 0（对数值型变量）或空字符（对字符变量）；而对 auto 变量来说，如果不赋初值则它的值是一个不确定的值，所以选项 D 说法不正确。故本题答案为 B。

80、以下与存储类别有关的四组说明符中，全部属于静态类的一组是（ ）。

- A、register 和 extern
- B、auto 和 static
- C、register 和 static
- D、extern 和 static

【解析】： 本题的考查知识点是静态类。

C 语言中，有两种存储类别：一种是自动类，一种是静态类。局部变量既可以说明成自动类，也可以说明成静态类；而全局变量只能是静态类。对于全局变量可使用 extern 和 static 两种说明符，故选项 D 正确。所以本题答案选 D。

81、有以下程序，程序运行后的输出结果是（ ）。

```
#include <stdio.h>
```



```
void fun(int n)
{
    static int x[3]={1,2,3};
    int k;
    for( k=0; k<3; k++ ) x[k]+= x[k]-n;
    for( k=0; k<3; k++ ) printf("%d,",x[k]);
}

main ()
{    fun(0); fun(1);}
```

- A、 2, 4, 6, 3, 7, 11,
- B、 2, 4, 6, 1, 3, 5,
- C、 1, 2, 3, 0, 1, 2,
- D、 2, 4, 6, 4, 8, 12,

【解析】： 本题的考查知识点是静态局部变量。

fun 函数中 n=0 时，第一个 for 循环语句中，k=0 时， $x[0]=2x[0]-0=2$ ，k=1 时， $x[1]=2x[1]-0=4$ ，k=2 时， $x[2]=2x[2]-0=6$ ；第二个 for 循环结束后，输出 2, 4, 6, 。

fun 函数中 n=1 时，第一个 for 循环语句中，k=0 时， $x[0]=2x[0]-1=3$ ，k=1 时， $x[1]=2x[1]-1=7$ ，k=2 时， $x[2]=2x[2]-1=11$ ；第二个 for 循环结束后，输出 3, 7, 11, 。故选项 A 正确。

82、有以下程序，程序运行后的输出结果是（ ）。

```
#define P 3

void F(int x) {return(P*x*x);}

main()
{printf("%d\n",F(3+5));}
```

- A、 192
- B、 29
- C、 25
- D、 编译出错

【解析】：本题考查的是宏定义。return 语句用于从被调函数带回一个函数值。void 关键字表示“无类型”，即不需要从被调函数中带回函数值，所以不需要 return 语句，故编译时出错。本题答案为 D。

83、以下叙述中正确的是()。

- A、预处理命令行必须位于源文件的开头
- B、在源文件的一行上可以有多条预处理命令
- C、宏名必须用大写字母表示
- D、宏替换不占用程序的运行时间

【解析】：本题考查的是预处理概念。

通常，预处理命令位于源文件的开头，但不一定必须位于开头，也可以写在函数与函数之间；由于预处理命令的末尾不加分号，所以，不能在一行上写多条预处理命令，否则，系统处理时就会把它当作一条命令；宏名一般习惯用大写字母表示，以便与变量名相区别，但这并非规定，也可用小写字母。故本题答案为 D。

84、有以下程序，程序运行后的输出结果是()。

```
#define f(x) (x*x)

main()
{
    int i1, i2;

    i1=f(8)/f(4) ;i2=f(4+4)/f(2+2);

    printf("%d, %d\n",i1, i2) ;
}
```

- A、64, 28
- B、4, 4
- C、4, 3
- D、64, 64

【解析】：本题考查的是宏定义。对带参数的宏的展开只是将语句中宏名后面括号内的实参字符串代替#define 命令行中的形参。本题中“i1=f(8)/f(4)”用#define f(x) (x*x)”代替，得 i1=(8*8)/(4*4)，结果 4；“i2=f(4+4)/f(2+2)”用“#define f(x) (x*x)”代替，得 i2= (4+4*4+4)/(2+2*2+2)，求得结果为 3。

故本题输出的值为 4,3。本题答案为 C。

85、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、对 C 源程序进行编译时用 100 替换标识符 N
- D、在运行时用 100 替换标识符 N

【解析】：本题的考查点是宏定义。无参宏的宏名后不带参数。其定义的一般形式为：#define 标识符 字符串，其中的“#”表示这是一条预处理命令。凡是以“#”开头的均为预处理命令。“define”为宏定义命令。“标识符”为所定义的宏名。“字符串”可以是常数、表达式、格式串等。例如：#define N 100 它的作用是规定标识符 N 来代替常数 100。在编写源程序时，所有的 100 都可由 N 代替，而对源程序作编译时，将先由预处理程序进行宏代换，即用 100 去置换所有的宏名 N，然后再进行编译。故本题答案为 B。

86、设有以下说明语句，则下面的叙述中不正确的是()。

```
struct ex
{ int x; float y; char z; } example;
```

- A、struct 是结构体类型的关键字
- B、example 是结构体类型名
- C、x, y, z 都是结构体成员名
- D、struct ex 是结构体类型

【解析】：example 是结构体变量名。本题答案为 B。

87、以下程序的输出结果是()。

```
struct HAR
{ int x, y; struct HAR *p; } h[2];
main ()
{ h[0].x=1; h[0].y=2;
```

```

        h[1].x=3; h[1].y=4;
        h[0].p=&h[1]; h[1].p=h;
        printf("%d%d\n", (h[0].p)->x, (h[1].p)->y);
    }

```

A、12

B、23

C、14

D、32

【解析】： 本题考查的是 C 语言中结构体与链表。

本题中是一个含有两个结点的循环链表。C 语言中结构体的定义为：

struct 结构体类型名

```

{
    成员项表;
};

```

本题答案为 D。

88、以下程序的输出结果是()。

```

union myun
{
    struct
    {
        int x,y,z;}u;
    int k;
}a;
main()
{
    a.u.x=4; a.u.y=5; a.u.z=6;
    a.k=0;
    printf("%d\n",a.u.x) ;
}

```

A、4

B、5

C、6

D、0

【解析】： 本题主要考查 C 语言中构造类型，联合体与结构体。结构体中的

成员使用各自的存储区，而联合体中的成员使用共同的存储区。所以，当给联合体中的成员变量 k 赋值后，联合体中的成员 u. x 也被覆盖，因为它们占用的是同一存储单元。本题答案为 D。

89、有以下程序，程序运行后的输出结果是()。

```
struct s
{int x,y;} data[2]= {10, 100, 20, 200};
main ()
{
    struct s *p=data;
    printf("%d\n",++(p->x));
}
```

A、 10

B、 11

C、 20

D、 21

【解析】：本题的考查点是结构体变量的初始化。该题是一个对外部存储类型的结构体变量进行的初始化。初始化后，x 的值为 10，y 的值为 100。++ (p->x) 中首先 p->x 是把 p 指向结构体变量 s 中的 x 成员，此++(p->x) 就相当于++x。这时 x 先自增，再使用，所以此时 x 的值为 11。故本题答案为 B。

90、有以下说明和定义语句

```
struct student
{
    int age; char num[8];
};struct
student stu[3]={ {20, " 200401" }, {21, " 200402" }, {19, " 200403"} };
struct student *p=stu;
```

以下选项中引用结构体变量成员的表达式错误的是()。

A、 (p++)->num

B、 p->num

C、 (*p). num

D、stu[3].age

【解析】：本题考的是结构体变量的引用。

引用形式有以下三种：①结构体变量.成员名；②(*p).成员名；③p→成员名。所以 A, B, C 答案都是正确的。故本题答案为 D。

91、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->y

B、pt->X

C、++pt->x

D、(pt++)->x

【解析】：本题考查的是结构体类型的指针。

题目中定义了一个 st 结构体类型，然后定义了 st 型的结构体指针变量*pt 及结构体数组 c，并对结构体数组进行了初始化，且将结构体指针变量 pt 指向了数组 c，则 pt->x 的值为 10，则++pt->x 的值为 11（注：“->”运算符的优先级高于“++”），故选项 C 正确。

92、有以下程序

```
main()
{
    union
    {
        char ch[2];
        int d;
    }s;
```

```
s.d=0x4321;
printf("%x, %x\n", s.ch[0], s.ch[1]);
}
```

在 16 位编译系统上，程序执行后的输出结果是()。

- A、 21, 43
- B、 43, 21
- C、 43, 00
- D、 21, 00

【解析】：本题考查的是共用体。int 类型变量和字符数组 ch 共用两个字节的相同的存储单元，通常 ch[0] 位于低字节，ch[1] 位于高字节，所以 s.ch[0]=21，s.ch[1]=43。

本题答案为 A。

93、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

【解析】：本题的考查点是结构体变量成员的引用。

有三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。其中点号“.” 称为成员运算符；箭头“->” 称为结构指向运算符。

故本题答案为 C。

94、设有以下语句

```
char a=3, b=6, c;
```

```
c=a^b<<2;
```

则 c 的二进制值是()。

A、00011011

B、00010100

C、00011100

D、00011000

【解析】：本题的考查点是位运算。

“<<”是 C 语言中规定的左移运算符，例如， $a=a<<2$ ，这个语句即是将 a 的二进制数左移两位，左移一位相当于该数乘以 2，左移两位相当于该数乘以 2 的 2 次方：是异或运算符，所以，c 的二进制值应为 00011011。故本题答案为 A。

95、有以下程序：

```
main()
```

```
{    unsigned char a, b;  
    a=4|3;  
    b=4&3;  
    printf("%d %d\n", a, b);  
}
```

执行后输出结果是()。

A、7 0

B、0 7

C、11

D、43 0

【解析】：“按位与”运算符(&)参加运算的两个运算量，如果两个相应的位都为 1，则该位的结果值为 1，否则为 0。“按位或”运算符(|)两个相应位

中只要有一个为 1，该位的结果为 1。4 转换为二进制数为 0100，3 转换为二进制数为 0011， $4|3=0111$ 即 7， $4\&3=0000$ ，即 0。故本题答案为 A。

96、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“按位异或”的运算
- D、^表示“按位异或的运算

【解析】：本题的考查点是运算符。

^表示“按位取反”的运算，|表示“逻辑或”的运算，~表示“按位异或”的运算。本题答案为 A。

97、以下叙述中错误的是()。

- A、二进制文件打开后可以先读文件的末尾，而顺序文件不可以
- B、在程序结束时，应当用 fclose 函数关闭已打开的文件
- C、利用 fread 函数从二进制文件中读数据，可以用数组名给数组中所有元素读入数据
- D、不可以用 FILE 定义指向二进制文件的文件指针

【解析】：本题的考查点是对文件的操作。在 C 语言中，可以用 FILE 定义指向二进制文件的文件指针。本题答案为 D。

98、以下叙述中不正确的是()。

- A、C 语言中的文本文件以 ASCII 码形式存储数据
- B、C 语言中对二进制位的访问速度比文本文件快
- C、C 语言中，随机读写方式不用于文本文件
- D、C 语言中，顺序读写方式不用于二进制文件

【解析】：本题主要考查 C 文件的基本概念。

C 语言把文件看作是一个字符(字节)的序列，根据数据组织形式可分为 ASCII 文件(又称文本文件)和二进制文件，即一个 C 文件可以看成是一个字节流

或二进制流，对于流式文件，可以进行顺序读写，也可以进行随机读写，关键在于控制文件的位置指针，由于文本文件要发生字符转换，计算位置时往往会发生混乱，访问速度受到影响，所以随机读写方式不用于文本文件。故本题答案为 D。

99、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

【解析】： 本题考查的是 C 语言中文件的读写。

C 语言中，有两种对文件的存取方式：顺序存取和直接存取，所以选项 A 错误；在对文件进行读、写操作前，先要使用库函数 `fopen` “打开”文件来实现程序中要读、写的文件与实际数据文件间的联系。在进行打开文件操作时可以指定其打开方式，如果使用“r+”方式则在写新数据时，只覆盖新数据所占的空间，其后的老数据并不丢失，所以选项 B 错误；在对文件进行操作后关闭文件是为了防止文件缓冲区中的剩余数据的丢失。本题答案为 D。

100、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、一个内存块的字节数

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、一个内存块的首地址，代表读入数据存放的地址

【解析】： 本题考查的是函数。

`buffer` 是数据块的指针，对 `fread` 来说，它就是内存块的首地址，输入的

数据存入此内存块中。本题答案为 D。

101、结构化程序设计所规定的三种基本控制结构是()。

- A、 输入、处理、输出
- B、 树形、网形、环形
- C、 顺序、选择、循环
- D、 主程序、子程序、函数

【解析】：顺序，选择和循环是结构化程序的三种基本结构。结构化程序设计方法学说出现以后，大多数高级语言都提供这三种控制结构。

本题答案为 C。

102、在 C 语言中，不正确的 int 类型的常数是()。

- A、 32768
- B、 0
- C、 037
- D、 0xAF

【解析】：整型(int) 常量即整常数。C 整常数可用以下三种形式表示：

(1)十进制整数。如 123，-456，0。

(2)八进制整数。以 0 开头的数是八进制数。如 0123 表示八进制数 123，即 $(123)_8$ ，其值为： $1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0$ ，等于十进制数 83。-011 表示八进制数-11，即十进制数-9。

(3)十六进制整数。以 0x 开头的数是 16 进制数。如 0x123，代表 16 进制数 123，即 $(123)_{16} = 1 \times 16^2 + 2 \times 16^1 + 3 \times 16^0 = 256 + 32 + 3 = 291$ 。-0x12 等于十进制数-18。

一个 int 型的常数的值范围为 $-2^{15} \sim (2^{15}-1)$ ，即-32768~32767。故本题选项 A 不是正确的 int 类型的常数。

本题答案为 A。

103、C 语言运算对象必须是整型的运算符是()。

- A、 %

B、 /

C、 =

D、 <=

【解析】：算术运算符包括+、-、*、/、%(模运算符)，其中要求%两侧的数据均为整型，其它算术运算符的运算对象可为任意类型的数据。

此处需注意的是“/”，两个整数相除的结果为整数，如 5/3 的结果为 1，舍去小数部分，如果参加运算的两个数中有一个数为实数，则结果是 double 型，因为所有实数都按 double 型进行运算；

“=”是赋值运算符，它用于将一个表达式的值赋给一个变量。赋值运算并不要求它的运算对象一定是整型，但是当赋值运算符两旁的类型不致，且都是数值型或字符型时，在赋值时要进行类型转换。

“<=”是关系运算符，用关系运算符将两个表达式连接起来的式子称为关系表达式，由此可知，C 语言对关系运算符的运算对象也没有作限制。

本题答案为 A。

104、若变量 a 是 int 类型，并执行了语句：a='A'+1.6；，则正确叙述是（ ）。

A、 a 的值是字符 C

B、 a 的值是浮点型

C、 不允许字符型和浮点型相加

D、 a 的值是字符 'A' 的 ASCII 值加上 1

【解析】：在赋值表达式中，赋值符右边的值的类型会自动转换成赋值符左边的变量的类型。在本题中，先用“A”的 ASCII 码值加上 1.6 得到 66.6，然后再将它强制转换成 int 型，转换后，实数的小数部分全部舍去。

本题答案为 D。

105、以下程序段的输出结果是（ ）。

```
int a=1234;
```

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

A、 12

- B、 34
- C、 1234
- D、 提示出错、无结果

【解析】：本题主要考查对 printf 函数域宽描述符的理解，在这里我们要特别注意：输出数据的实际精度并不主要决定于格式项中的域宽与精度，也不决定于输入的数据精度，而主要决定于数据在机器内的存储精度。

本题答案为 C。

106、以下选项中不属于 C 语言的类型是()。

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

【解析】：C 语言中归纳起来，基本类型数据有以下几种：

```
[signed] char;
unsigned char;
[signed] short [int];
unsigned short [int];
[signed] long [int];
unsigned long [int] ;
float ;
double;
long double;
```

本题答案为 D。

107、若有说明语句：int a, b, c, *d=&c;则能正确从键盘读入三个整数分别赋给变量 a、b、c 的语句是()。

- A、 scanf("%d%d%d", &a,&b, d);

- B、 `scanf("%d%d%d", &a, &b, &d);`
- C、 `scanf("%d%d%d", a, b, d);`
- D、 `scanf("%d%d%d", a, b, *d);`

【解析】：在 `scanf` 语句中，要为某一变量赋值，引用的是该变量的地址。本题中，由于 `d` 为指针变量，所以 `d` 就表示地址，故本题读入数据的形式应写成 `scanf ("%d%d%d", &a, &b, d)`。

本题答案为 A。

108、以下叙述正确的是()。

- A、 可以把 `define` 和 `if` 定义为用户标识符
- B、 可以把 `define` 定义为用户标识符，但不能把 `if` 定义为用户标识符
- C、 可以把 `if` 定义为用户标识符，但不能把 `define` 定义为用户标识符
- D、 `define` 和 `if` 都不能定义为用户标识符

【解析】：`if` 属于系统定义的 32 个关键字之一，所以不可以供用户定义。此处需注意的是不要把 `define` 误认为是宏定义命令，正确的宏定义命令为：`#define`。

本题答案为 B。

109、以下程序的输出结果是()。

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

- A、 5
- B、 4
- C、 6
- D、 不确定

【解析】：本题最重要的是分析 $d=a>c?(a>c?a:c):(b)$ ，运算此表达式的顺序是自左向右，先比较 $a>c$ ，值为假，所以整个表达式的值就为 b 的值，即 4。

本题答案为 B。

110、以下程序中，while 循环的循环次数是()。

```
main()
{
    int i=0;
    while(i<10)
    {
        if(i<1) continue;
        if(i==5)break;
        i++;
    }
    .....
}
```

A、 1

B、 10

C、 6

D、 死循环，不能确定次数

【解析】：题目中，进入循环后，先执行一个条件语句，如果 i 的值小于 1，那么直接进入下一轮循环，因为 i 的初始值是 0，小于 1，故直接进入下一轮循环，又因为 i 的值始终没有改变，所以这个循环成了死循环。

本题答案为 D。

111、以下程序的输出结果是()。

```
main()
{
    int a=0,i ;
```

```
for(i=1; i<5; i++)
{
    switch(i)
    {
        case 0:
        case 3:a+=2;case 1:
        case 2:a+=3;
        default:a+=5;
    }
}

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

- A、 31
- B、 13
- C、 10
- D、 20

【解析】：switch 结构的执行过程为:进入 switch 结构后,对条件表达式进行计算,然后从上至下去找与条件表达式的值相匹配的 case,以此作为入口,执行 Switch 结构中后面的各语句。第一次 for 循环中,switch 结构的条件表达式 i 的值是 1,则从 case 1 开始执行后面的语句,先执行 a+=3, a 的值变成 3,接着执行 a+=5, a 的值变成了 8。然后进入第二次循环,switch 结构条件表达式 i 的值 2,则从 case 2 开始执行后面的语句,结束第二轮循环时 a 的值是 16,在第三轮循环中,switch 结构条件表达式的值是 3,则从 case 3 后面的语句开始执行,a+=2 语句使 a 的值变成了 18,然后执行 a+=3,和+=5,第三轮循环结束时,a 的值是 26。第四轮循环中,switch 结构条件表达的值是 4,从 default 处开始执行,执行一个 a+=5 的操作,a 的值变成了 31,i 的值经修改变成了 5,不再满足循环条件,退出循环。

本题答案为 A

112、以下程序的输出结果是()。


```
main()
{
    int a=4, b=5, c=0, d;
    d=!a&&!b||!c ;
    printf("%d\n", d);
}
```

- A、 1
- B、 0
- C、 非 0 的数
- D、 -1

【解析】：!a 值为 0，故!a&&!b 的值为 0，!c 的值为 1，而 0||1 的值应该是 1。所以答案是 A。

本题答案为 A。

113、以下程序的输出结果是()。

```
#include<stdio.h>
main()
{
    int i=0, a=0;
    while(i<20)
    {
        for(; ;)
        {
            if((i%10)==0) break;
            else i--;
        }
        i+=11; a+=i;
    }
    printf("%d\n", a) ;
}
```

- A、 21

B、 32

C、 33

D、 11

【解析】：本题中，有两个循环语句。首先， $i=0$ ，进入 while 循环，for 语句中对循环没有任何条件，故直接进入 for 循环，因为 $0\%10$ 结果还是 0，所以跳出 for 循环，执行 $i+=11$ ， i 的值变为 11，执行 $a+=i$ ， a 的值变为 11，接着进入下一轮 while 循环，在 for 循环中，因为 i 的值是 11，对 10 取余结果为 1，所以执行 i 自减， i 的值变成 10，进入新一轮 for 循环，因为 $10\%10==0$ ，所以跳出 for 循环，执行 $i=11$ ， i 的值变成 21，执行 $a+=i$ ，也就是将 $21+11$ 的和赋给 a ， $a=32$ ，因为此时 i 的值不再小于 20，所以结束循环，故输出结果是 32。

本题答案为 B。

114、以下程序的输出结果是()。

```
int f()
{
    static int i=0;
    int s=1;
    s+=i; i++;
    return s;
}

main()
{
    int i, a=0;
    for(i=0; i<5; i++) a+=f();
    printf("%d\n", a) ;
}
```

A、 20

B、 24

C、 25

D、 15

【解析】：本题主要考的是对变量存储属性的理解，一个变量被指定为静态变量，在编译时该程序执行结束都存在，而不像动态变量只存在于函数或分程序被调用期间。在函数第一次调用函数结束时，i 的值是 1，返回值 1，第二次调用函数时，i 的值保持为 1，执行自加操作后，i 的值变成了 2，到了第三次调用时，每次调用时，i 的值保持了上次调用结束时的值，再执行自加，值变成了 3。到第五次调用结束时，i 的值是 5。而 s 的值不具备 i 的这种可继承性，每次调用时，它都先被重新赋值为 1，再执行下面的操作。

本题答案为 D。

115、以下程序段的输出结果是：()。

```
char s[]="\\141\\141abc\t" ;
printf("%d\n" , strlen(s)) ;
```

- A、 9
- B、 12
- C、 13
- D、 14

【解析】：本题考的是对转义字符的理解，在 C 语言中规定：用反斜杠开头后面跟一个字母代表一个控制字符；用\\代表字符“\”，用\'代表撇号字符；用反斜杠后面跟 1 到 3 个八进制数代表 ASCII 码为该八进制数的字符；用\x 后面跟 1 到 2 个十六进制数代表 ASCII 码为该十六进制数的字符。

本题答案为 A。

116、以下程序调用 findmax 函数返回数组中的最大值。

```
findmax(int *a, int n)
{
    int *p,*s;
    for(p=a,s=a; p-a<n ; p++)
        if (____) s=p;
    return(*s);
```

```
}  
main()  
{  
    int x[5]={12, 21, 13, 6, 18} ;  
    printf ("%d\n", findmax (x, 5));  
}
```

在下划线处应填入的是()。

- A、 $p > s$
- B、 $*p > *s$
- C、 $a[p] > a[s]$
- D、 $p - a > p - s$

【解析】：findmax 函数求的是数组元素的最大值，首先让指针 p，s 都指向 a[0]，由最后的 return 语句可以知道，最终 s 代表数组元素最大值的地址，要实现这个目的，就要让 s 指针指向的内容与数组元素逐个比较，把较大值的地址放在 s 中，而在 for 循环中，每做一次循环，p 指针自加一次，即指针 p 遍历了整个数组，所以应该在循环语句中，用 p 指针指向的内容与 s 指针指向的内容作比较，并让两者间大值的地址存入 s。

本题答案为 B。

117、以下程序的输出结果是

```
main()  
{  
    char ch[3][5]={"AAAA", "BBB", "CC"} ;  
    printf("%s\n", ch[1]);  
}
```

- A、 "AAAA"
- B、 "BBB"
- C、 "BBBCC"
- D、 "CC"

【解析】：ch[1]是字符串“BBB”的起始地址，该 printf 函数的作用是从给定的地址开始逐个输出字符，直到遇到“\0”为止。

而在为二维数组赋值时，系统都会自动在每个字符串的末尾补上“\0”。所以在输出了三个 B 字符后遇到“\0”，结束输出。

本题答案为 B。

118、若指针 p 已正确定义，要使 p 指向两个连续的整型动态存储单元，不正确的语句是（ ）。

- A、 p=2*(int*)malloc(sizeof(int)) ;
- B、 p=(int*)malloc(2*sizeof(int))
- C、 p=(int*)malloc(2*2)
- D、 p=(int*)calloc(2, sizeof(int))

【解析】：malloc 函数的作用是在内存开辟指定大小的存储空间，并将此存储空间的地址作为函数值带回，它的原型为 void * malloc(unsigned int size)，函数值为指针，这个指针是指向 void 类型，也就是不规定指向任何具体的类型。如果要将此返回地址赋给一个指向特定类型的指针变量，则应进行强制类型转换。

calloc 函数的模型是：void * calloc(unsigned int num, unsigned int size)作用是分配 num 个大小为 size 字节的空间。

本题答案为 A。

119、以下程序的输出结果是（ ）。

```
main()
{
    int x=040;
    printf("%o\n", x<<1);
}
```

- A、 100
- B、 80

C、 64

D、 32

【解析】：040 用二进制表示就是 00100000，左移一位，得到的值应该是 01000000，化成八进制应该是 0100。

本题答案为 A。

120、若要打开 A 盘上的 user 子目录下名为 abc.txt 的文本文件进行读、写操作，下面符合此要求的函数调用是()。

A、 fopen("A:\user\abe.txt", "r")

B、 fopen("A:\\user\\abc.txt", "r+")

C、 fopen("A:\user\abc.txt", "rb")

D、 fopen("A:\\user\\abc.txt", "w")

【解析】：本题考查点是 fopen 函数中文件使用方式的指定。

文件使用方式“r”，表示“只读”方式打开一个字符文件。

文件使用方式“r+”，表示“读写”方式打开一个字符文件。

文件使用方式“rb”，表示以“只读”方式打开一个二进制文件。

文件使用方式“w”，表示以“只写”方式打开一个字符文件。

本题答案为 B。

121、以下选项中函数形参不是指针的是()。

A、 fun(int &p) {...}

B、 fun(int a[10]) {...}

C、 fun(int *a) {...}

D、 fun(int p[]) {...}

【解析】：本题的考查点是形式参数。

当数组名作为实参时，对应的形参除了是指针外，还可以用另外两种形式。即三种形式如下：(1) arrin (int *a); (2) arrin (int a[]); (3) arrin(int a[M])。在第(2)和第(3)种形式中，虽然说明的形式与数组的说明相同，但 C 编译程序都将把 a 处理成第一种指针形式。

故本题答案为 A。

122、以下程序调用 fun 函数求数组中最大值所在元素的下标。

```
#include <stdio. h>

void fun( int *s, int n, int *k )
{
    int i;
    for( i=0, *k=i; i<n;i++ )
        if( s[i] > s[*k] )___;}

main()
{
    int a[5]={1,6,2,8,0}, k;
    fun(a, 5, &k);
    printf("%d %d\n", k, a[k] );
}
```

在横线处应填入的内容是()。

- A、k=i-s
- B、k=i
- C、*k=i-s
- D、*k=i

【解析】： 本题的考查点是函数的调用。

分析程序可知，if 条件句的作用是比较数组中的值大小，依次把最大值所在元素的下标存放在变量 k 中。所以当 $s[i] > s[*k]$ ，要使 a[k] 中存放的是最大元素的下标，即 $*k=i$ 。

故本题的答案为 D。

123、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、60
- B、68
- C、99
- D、108

【解析】：本题的考查点是数组与循环语句。

由(*p)[4]=a可知，p指向a数组的第四个元素的地址，即*p=7。k=k+*(*(p+i)+j)可知，k是累加的*(*(p+i)+j)的值。循环一共进行了2*3=6次，分别累加的是：*(*(p+0)+0)、*(*(p+0)+1)、*(*(p+1)+0)、*(*(p+1)+1)、*(*(p+2)+0)、*(*(p+2)+1)，即*p、*(p+1)、*(p+1)、*(p+2)、*(p+2)、*(p+3)，所以 k=7+9+9+11+11+13=60。

故本题答案为A。

124、有以下程序

```

#include <stdio. h>

int  funa( int a, int b) { return a+b; }
int  funb( int a, int b) { return a-b; }
int  sss( int (*t)(), int x, int y ) { return( (*t)(x,y) );}

main()
{
    int x;

    x=sss(funa,9,3); x+=sss(funb,8,3); printf("%d\n", x);
}

```


程序运行后的输出结果是()。

- A、24
- B、23
- C、17
- D、22

【解析】：本题的考查点是函数的调用。

由程序可知， $x+=sss(\text{funb}, 8, 3)$ 即 $x=x+sss(\text{funb}, 8, 3)=sss(\text{funa}, 9, 3)+sss(\text{funb}, 8, 3)$ 。

所以， $x=\text{funa}(9, 3)+\text{funb}(8, 3)=(9+3)+(8-3)=12+5=17$ 。

故本题答案为 C。

125、有以下程序

```
#include <stdio. h>

void fun(int a, int b,int c)
{a=b; b=c; c=a;}

main()
{
    int a=10, b=20, c=30;
    fun(a, b, c);
    printf("%d, %d, %d\n", c, b, a);
}
```

程序运行后的输出结果是()。

- A、0, 0, 0
- B、10, 20, 30
- C、20, 30, 10
- D、30, 20, 10

【解析】：本题的考查点是函数的调用。

程序中，实参的值传递给形参，在函数中，值进行了交换，但并不改变实际

参数的值，因此 a、b、c 的值仍然是 10、20、30。输出 c、b、a 的值，即为” 30, 20, 10”。

故本题答案为 D。

126、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、50

B、5

C、6

D、60

【解析】：本题的考查点是数组与指针。

程序中，p 和 s 均为指针变量，p = c 实现把数组 c 的首地址赋值给 p，也可以写成 p = &c[0]；s = &c[5] 实现将数组元素 c[5] 的地址赋值给 s，也可以写成 s=c+5，所以 s-p=5。

故本题答案为 B。

127、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、 1.7
- B、 2.0
- C、 -2
- D、 -3

【解析】： 本题的考查点是函数的调用。

函数返回值为 int 型,所以 $3.2-4.1-1.3=-2.1$, 经过四舍五入得返回值为-2。
所以本题答案为 C。

128、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, 1, 4
- B、 1, 2, 4
- C、 1, 4, 8
- D、 4, 4, 4

【解析】： 本题的考查点是 sizeof 关键字。

sizeof 返回对象或者类型所占的内存字节数。char、int、float 类型占用的字节数均为 4。

故本题的答案为 D。

129、有以下程序

```
#include <stdio. h>

main( )
{
    int a = 2, *ptr;
    ptr = &a;
    *ptr = 8;
    a = (*ptr) ++;
    printf("%d, %d\n", a, *ptr);
}
```

程序运行后的输出结果是()。

A、0,4

B、2,4

C、8,9

D、9,9

【解析】： 本题的考查点是指针与数组。

程序中输出的是 a 的值和*ptr 的值。由 ptr = &a 可知，指针变量 ptr 指向的是 a 的地址。由*ptr = 8 和 a = (*ptr) ++得 a 的值为 9。最后程序输出 a 的值和*ptr 的值，因为变量 ptr 指向的是 a 的地址，所以实际输出的都是 a 的值，即输出值为 9,9。

故本题答案为 D。

130、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, 2, 0, 1, 4, 1,
- B、 1, 2, 4, 9, 6, 0,
- C、 2, 9, 0, 0, 1, 4,
- D、 1, 0, 2, 0, 0, 9,

【解析】： 本题的考查点是输入与输出语句。

程序中双重循环共循环 6 次，一共输出 6 个值，分别为：array[0][0], array[0][0] , array[1][0], array[2][0] , array[0][1] , array[1][1] , array[2][1]，即分别为” 1, 0, 2, 0, 0, 9, ”。

故本题答案为 D。

131、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、2, 4, 8,

【解析】：本题的考查点是循环结构与指针。

由程序中 ptr=array 可知，指针变量指向数组 array 的首地址，语句 printf("%d, ", *ptr++) 是输出当前指针指向的元素值，*ptr++向 高地址移动指针 ptr，使指针指向下一个存储单元。一共循环三次，输出 3 个值 array[0]、array[1]、array[2]，即为 1, 1, 3。

故本题答案为 C。

132、有以下程序

```
#include <stdio. h>  
main( )  
{  
    int i, *ptr;  
    int array[4] = {0,1};  
    for (ptr=array, i=0; i<3; i++, ptr++)  
    {  
        if (*ptr == 0)  
            putchar('0');  
        else  
            putchar ('M' + *ptr);  
    }  
}
```

```
printf("\n" );
}
```

程序运行后的输出结果是()。

- A、M0#
- B、#F#
- C、MMM
- D、#N#

【解析】：本题的考查点是数组与指针。

由 $i=0$; $i<3$ 可知，程序共循环 3 次。

第一次， $i=0$ ，由 `int array[4] = {0, 1}` ;语句可知， $*ptr=0$ ，输出值：#；

第二次， $i=1$ ，由 `int array[4] = {0, 1}` ;语句可知， $*ptr=1$ ，输出值：N；

第三次， $i=2$ ，由 `int array[4] = {0, 1}` ;语句可知， $*ptr=0$ ，输出值：#。

所以程序运行后的输出结果为：#N#。

故本题答案为 D。

133、设有定义：`int x=2, *p=&x; float y=3.0; char z='c'` ;，则立即进行以下运算有安全隐患的是()。

- A、`p++`;
- B、`x++`;
- C、`y++`;
- D、`z++`;

【解析】：本题的考查点是指针与指针变量。

A 选项，`*p=&x` 表示，将指针变量指向变量 `x` 的地址。执行 `p++`，会使指针指向意想不到的地方，致使程序失控，严重的将导致系统崩溃。B、C、D 均为变量值的自增，没有安全隐患。

故本题答案为 A。

134、以下对指针变量的操作中，错误的程序段是()。

- A、int x=0, *p; *p=x;
- B、int x=1, *p, *q=&x; p=q;
- C、int *p, *q; q=p=NULL;
- D、int p, *q; q=&p;

【解析】：本题的考查点是指针与指针变量。

A 选项和 D 选项类似，A 选项中 x 为整型变量，p 为指针变量。*p=x 是非法的。正确的应为：p=&x，表示使指针变量 p 指向 x 的地址。即 A 选项错误，D 选项正确。B 选项，把一个指针变量 q 中的地址值赋给另一个指针变量，从而使这两个指针指向同一个地址，即都指向 x 的地址。所以 B 选项正确。C 选项，NULL 的代码值为 0，当执行了以上的赋值语句后，p 和 q 均为空指针。即 C 选项正确。

故本题答案为 A。

135、以下程序的输出结果是()。

```
main()
{
    char a[10]={ '1','2','3','4','5','6','7','8','9',0}, *p;
    int i ;
    i=8;
    p=a+i ;
    printf("%s \n", p-3) ;
}
```

- A、 6
- B、 '6789'
- C、 '6'
- D、 789

【解析】：本题的考查点是通过指针引用数组元素。

指针 p 指向数组中第 9 个元素，即 a[8]='9'。输出时，输出字符串，且指针把 a[8]作为初始位置往前指 3 个元素，也就是共有 4 个字符输出。最后指向 a[5]为'6'，所以输出为 6789。

故本题答案为 B。

136、下列叙述中正确的是()。

- A、 C 语言编译时不检查语法
- B、 语言的子程序有过程和函数两种
- C、 C 语言的函数可以嵌套定义
- D、 C 语言的函数可以嵌套调用

【解析】：选项 A 的错误在于编译过程中是检查语法的，若发现源程序有语法错误，则系统会提示出错信息；选项 B 的错误在于 C 语言中，子程序的作用是由函数来完成的，无过程的概念；选项 c 的错误在于函数不可以嵌套定义，但可以嵌套调用。

本题答案为 D。

137、以下___是不合法的 C 语言常量。

- A、 0xFF
- B、 1.2e0.5
- C、 2L
- D、 ' \72'

【解析】：C 语言中的指数形式，字母 E 之前必须有数字，且 E 后面指数必须为整数。

本题答案为 B。

138、设 x、y、t 均为 int 型变量，则执行语句： $x=y=3$; $t=++x$ || $++y$; 后，y 的值为()。

- A、 不定值
- B、 4
- C、 3
- D、 1

【解析】：C 语言中做逻辑或运算时，自左向右判断各个参加或运算的表达

式，一旦运算到某式的值为非零时，表明整个表达式定为真，余下的语句则不再进行运算。本题中，++x 的值为真，则++y 并没有进行运算，所以 y 值不变。

本题答案为 C。

139、若变量已正确说明为 float 类型，要通过语句 `scanf("%f%f%f", &a, &b, &c);` 给 a 赋予 10.0，b 赋予 22.0，c 赋予 33.0，不正确的输入形式是（ ）。

A、

10

22

33

B、

10.0, 22.0, 33.0

C、

10.0

22.0 33.0

D、

10 22

33

【解析】：C 语言中，数据输入时，必须严格按照 scanf 语句中的格式符来进行输入，不需要分隔符，否则作为有效字符读入。选项 B 中的逗号将被做为有效字符读入。

本题答案为 B。

140、若执行以下程序时从键盘上输入 9，则输出结果是（ ）。

```
main(
```

```
{
```

```
    int n;
```

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

```

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

```

- A、 11
- B、 10
- C、 9
- D、 8

【解析】:n++是在语句结束后再加一,因此,if 语句中相当于判断的是 $9 < 10$,先执行“printf(“%d\n”,n);”,而这时,因为 if 语句被执行,n 的值也被加 1,所以在打印时,n 值为 10,而 else 语句此时并未执行。

本题答案为 B。

141、若 a、b、c1、c2、x、y 均是整型变量,正确的 switch 语句是()。

①

```

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

```

②

```

switch(a*a+b*b)
{
    case 3:
    case 1 : y=a+b; break;
    case 3 : y=b-a; break;
}

```

③

```

switch a

```

```
{  
    case c1 : y=a-b ; break;  
    case c2: x=a*b; break ;  
    default :x=a+b;  
}
```

④

```
switch (a-b)  
{  
    default:y=a*b; break ;  
    case 3:case 4:x=a+b; break ;  
    case 10:case 11 :y=a-b; break ;  
}
```

A、 ①

B、 ②

C、 ③

D、 ④

【解析】：选项 A 的错误在于 switch 语句后面不该加分号；

选项 B 的错误在于 switch 语句要求每一个 case 的常量表达式的值必须互不相同；

选项 C 的错误在于变量 a 应该用括号括起来，case 后面应该是常量表达式，而不是变量。

本题答案为 D。

142、 以下程序执行后 sum 的值是()。

```
main ()
```

```
{
    int i, sum;
    for(i=1; i<6; i++) sum+=i;
    printf("%d\n", sum) ;
}
```

- A、 15
- B、 14
- C、 不确定
- D、 0

【解析】： 变量 sum 在使用时未赋初值，所以无法确定 sum 最终的值。
 本题答案为 C。

143、有以下程序段

```
int x=3;
do
{
    printf("%d ", x--=2);
}while(!(--x));
```

其输出结果是()。

- A、 1
- B、 3 0
- C、 1 -2
- D、 死循环

【解析】： 本题中循环执行了两次，第一次，执行完 printf 语句后，x=1，而--x 是先减后执行 x 所在的表达式，因此(!(--x))的值为非零，所以又执行了一次循环。第二次 x 的值为-2，--x 的值为-3，(!(--x))的值为 0，跳出循环。

本题答案为 C。

144、若变量 c 为 char 类型，能正确判断出 c 为小写字母的表达式是()。

- A、 'a' <= c <= 'z'
- B、 (c >= 'a') || (c <= 'z')
- C、 ('a' <= 'c') and ('z' >= 'c')
- D、 (c >= 'a') && (c <= 'z')

【解析】：(c >= 'a') 和 (c <= 'z') 的关系应该是逻辑与的关系。

本题答案为 D。

145、以下所列的各函数声明中，正确的是()。

- A、 void play(var a:Integer, var b:Integer)
- B、 void play(int a, b)
- C、 void play(int a, int b)
- D、 Sub play(a as integer, b as integer)

【解析】：除选项 C 外，其它各项都没有按照 C 语言中的函数定义规则定义函数。

本题答案为 C。

146、下列程序段的输出结果是()。

```
void fun (int *x, int*y)
{ printf("%d %d", *x,*y) ;*x=3 ;*y=4;}
main()
{
    int x=1,y=2;
    fun(&y,&x);
    printf("%d %d", x,y);
}
```

- A、 2 1 4 3
- B、 1 2 1 2

C、 1 2 3 4

D、 2 1 1 2

【解析】：本题中函数传递的是变量的地址，如果形参变量所指向的存储空间的内容在函数中发生改变，相对应的实参变量所指向的存储空间的内容也就发生改变。

本题答案为 A。

147、下列程序的输出结果是()。

```
main()
{
    char a[10]={9,8,7,6,5,4,3,2,1,0}, *p=a+5;
    printf("%d", *--p);
}
```

A、 非法

B、 a[4]的地址

C、 5

D、 3

【解析】：本题中 `char *p=a+5;`即相当于 `p=&a[5]`，而`--p`，是先要减 1，则 `printf` 语句要输出的值就是 `a[4]`。

本题答案为 C。

148、下列程序的运行结果是()。

```
void fun(int *a,int *b)
{
    int *k;
    k=a;a=b;b=k;
}

main()
```

```
{
    int a=3,b=6,*x=&a,*y=&b;
    fun(x,y);
    printf("%d %d",a,b);
}
```

- A、 6 3
- B、 3 6
- C、 编译出错
- D、 0 0

【解析】：本题中主函数里的 x、y，fun 函数里的 a、b、k，这些都是指针，fun 函数中只是将 a、b 这两个指针交换了位置，而并没有改变主函数中变量 a、b 的值。

本题答案为 B。

149、当调用函数时，实参是一个数组名，则向函数传送的是（ ）。

- A、 数组的长度
- B、 数组的首地址
- C、 数组每一个元素的地址
- D、 数组每个元素中的值

【解析】：当调用函数时，实参是一个数组名，则向函数传送的是数组的首地址，函数中的形参可定义成以下三种形式：

- (1) 形参定义成数组；
- (2) 形参定义成可变长数组；
- (3) 形参定义为指针变量。

本题答案为 B。

150、设有以下说明语句

```
struct ex
```



```
{ int x; float y; char z; } example;
```

则下面的叙述中不正确的是()。

- A、 struct 是结构体类型的关键字
- B、 example 是结构体类型名
- C、 x, y, z 都是结构体成员名
- D、 struct ex 是结构体类型

【解析】：example 是结构体变量名。本题答案为 B。

151、以下存储类型中，只有在使用时才为该类型的变量分配内存的是()。

- A、 auto 和 static
- B、 auto 和 register
- C、 register 和 static
- D、 extern 和 register

【解析】：extern、register、static、auto 分别是定义外部变量、寄存器变量、静态变量、自动变量，其中，自动变量和寄存器变量属于动态存储，调用时临时分配单元；而静态变量和外部变量属于静态存储，在整个程序运行时间都存在。

本题答案为 B。

152、若 fp 是指向某文件的指针，且已读到文件末尾，则库函数 feof(fp) 的返回值是()。

- A、 EOF
- B、 -1
- C、 非零值
- D、 NULL

【解析】：feof(fp) 函数用来检测文件位置指针是否已指向文件的末尾，若已指向末尾，则函数值为非零值，否则函数值为 0。

本题答案为 C。

153、以下程序的输出结果是()。

```
main()
{
    int i,k,a[10],p[3];
    k=5;
    for(i=0;i<10;i++) a[i]=i;
    for(i=0;i<3;i++) p[i]=a[i*(i+1)];
    for (i=0;i<3;i++) k+=p[i]*2;
    printf("%d\n",k);
}
```

- A、 20
- B、 21
- C、 22
- D、 23

【解析】:按照程序的流程运行一遍,我们可以得到, $p[0]=0$, $p[1]=2$, $p[3]=6$;
执行三次 $k+=p[i]*2$, 相当于 $k=5+0*2+2*2+6*2=21$ 。

本题答案为 B。

154、以下程序的输出结果是()。

```
main()
{
    int i, x[3][3]={1,2,3,4,5,6,7,8,9} ;
    for(i=0;i<3;i++)
        printf("%d,", x[i][2-i]);
}
```

- A、 1,5,9,
- B、 1,4,7,
- C、 3,5,7,

D、 3, 6, 9,

【解析】：根据程序的顺序运行一遍，可以得到：

i=0，打印的是 x[0][2]=3;

i=1，打印的是 x[1][1]=5;

i=2，打印的是 x[2][0]=7。

本题答案为 C。

155、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、 13

B、 25

C、 9

D、 12

【解析】：本题的考查点是宏定义。

$D(i+j)=D(2+4)=4*2+4+1=8+4+1=13$ 。

故本题答案为 A。

156、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);
    printf("\n");
}
```

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;

【解析】：本题的考查点是程序。

“>>”是右移运算。

右移时，右端移出的二进制数舍弃，左端（高位）移入的二进制数分两种情况：对于无符号整数和正整数，高位补 0，对于负整数，高位补 1。

struct S 结构体中定义了两个变量，然后定义了一个结构体数组变量。

故 data[0].x=4, data[0].y=3, data[1].x=1, data[1].y=9。

data[o].y>>1=3>>1=1, data[1].y>>1=9>>1=4。

故本题答案为 A。

157、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

【解析】： 本题的考查点是程序。

change(&input,num)函数中 input 变量是地址传递, num 变量是值传递。故 input 的值变成:****, num 的值没有改变, 还是 4。

故本题答案为 A。

158、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、ABC
- B、ABC45
- C、12345
- D、ABC12

【解析】： 本题的考查点是程序。

因为 s1=s2, 故 s1.name=s2.name="ABC" 。故本题答案为 A。

159、有如下程序

```
#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, 3, 3, 3, 0,
- B、3, 5, 10, 4, 0,
- C、3, 1, 2, 0, 0,
- D、3, 2, 2, 2, 0,

【解析】：本题的考查点是程序。

运算符&的作用是：把参加运算的两个数按对应的二进制位分别进行“与”运算，当两个相应的位都为1时，该位的结果为1，否则为0。

当 i=0 时， $\text{array}[i] \& 3 = \text{array}[0] \& 3 = 3 \& 3 = 0011 \& 0011 = 3$;

当 i=1 时， $\text{array}[i] \& 3 = \text{array}[1] \& 3 = 5 \& 3 = 0101 \& 0011 = 1$;

当 i=2 时， $\text{array}[i] \& 3 = \text{array}[2] \& 3 = 10 \& 3 = 1010 \& 0011 = 2$;

当 i=3 时， $\text{array}[i] \& 3 = \text{array}[3] \& 3 = 4 \& 3 = 0100 \& 0011 = 0$;

当 i=4 时， $\text{array}[i] \& 3 = \text{array}[4] \& 3 = 0 \& 3 = 0000 \& 0011 = 0$ 。

故本题答案为 C。

160、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、A

B、EOF

C、ABCDE

D、E

【解析】：本题的考查点是程序。

当 i=0 时，'A' +0='A'，会向 output.txt 文件写入：A；

当 i=1 时，'A' +1='B'，会向 output.txt 文件写入:B，并覆盖 output.txt 文件中原来的值；

当 i=2 时，'A' +2='C'，会向 output.txt 文件写入:C，并覆盖 output.txt 文件中原来的值；

当 i=3 时，'A' +3='D'，会向 output.txt 文件写入:D，并覆盖 output.txt 文件中原来的值；

当 i=4 时，'A' +4='E'，会向 output.txt 文件写入:E，并覆盖 output.txt 文件中原来的值；

故 output.txt 文件中的内容只有 E。

故本题答案为 D。

161、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

【解析】： 本题的考查点是程序。

p1=p2=(char*)malloc(sizeof(char)*10)该语句开辟了内存单元，指针 p1、p2 都指向该片内存单元。p1+1 指向了字符 a 的地址，所以*p2="alloc"。又因为指针 p1、p2 指向的是同一内存单元，故*p1="alloc"。

故本题答案为 A。

162、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;

【解析】： 本题的考查点是结构体变量。

s 和 m 数组的下标是 0-1，因此 D 选项错误；

不能对一个结构体类型的数组直接赋值，因此 B 选项错误；

不能对一个结构体类型的数组中某一具体项赋值，因此 C 选项错误；故本题答案为 A。

163、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、 3,6

B、 1,2

C、 4,7

D、 变量的地址值

【解析】： 本题的考查点是结构体变量。

x[0].b 指向数组 x1 的首地址，x[1].b 指向数组 x2 的首地址。故本题答案为 A。

164、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

【解析】： 本题的考查点是程序。

s+1 指向“Qin”的地址，fun(s+1)函数中为 s+1 指向的内存单元中的内容重新赋值。故 s[1].name="Tong", s[1].age=20。故本题答案为 A。

165、以下不属于 C 语言位运算符的是()。

- A、 !
- B、 |
- C、 ^
- D、 ~

【解析】： 本题考查的是位运算。

C 语言提供六种位运算符，按优先级由高到低的顺序分别为：取反(^)→左移(<<)和右移(>>)→按位与(&)—按位异或(^)→按位或(|)。故本题答案为 A。

166、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

【解析】：本题考查的是 fopen 函数。

fp=fopen("test.txt" "w")为写而打开文本文件，如果指定的文件不存在，系统将在用 fopen 调用中的指定的文件名建立一个新文件。因此 if 条件判断为否，执行 else 语句。

故本题答案为 A。

167、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, 32
- B、16, 32
- C、8, 16
- D、16, 16

【解析】： 本题考查的是宏替换。

$\text{FNB}(a) = a + a + 2 + 2$, $\text{FNA}(\text{FNB}(a)) = 2 + 2 * 2 + 2 = 2 + 4 + 2 = 8$ 。

$\text{FNA}(b) = b * b = 4 * 4$, $\text{FNB}(\text{FNA}(b)) = 4 * 4 + 4 * 4 = 32$ 。

故本题答案为 A。

168、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

【解析】： 本题的考查点是用户定义类型 typedef) 。

C 语言允许用 typedef 说明一种新类型名,说明新类型名的语句一般形式为:

typedef 类型名 标识符;

故本题答案为 A。

169、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[0].n
- B、(*a)->n
- C、p->n
- D、(*p).n

【解析】：本题的考查点是对结构体变量成员的引用。

若已定义了一个结构体变量和基类型为同一结构体类型的指针变量，并使该指针指向同类型的变量则可以用以下三种形式来引用结构体变量中的成员。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

故本题答案为 B。

170、下面函数

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

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 x 复制到字符串 y
- D 将字符串 x 连接到字符串 y 后面

【解析】：本题的考查点是 while() 循环语句。

这个函数的功能是求字符串的长度，需要注意的是，在返回长度时一定要减一，否则计算出的串长度将会多

故本题答案是 A。

171、以下程序的输出结果是()。

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

A 14.000000

B 15.400000

C 13.700000

D 14.900000

【解析】：本题的考查点是 sqrt() 函数的应用。

函数 sqrt(x) 是一个标准的数学函数，用于求 x 的二次方根，知道这一点以后就好做了，将各变量的值带入程序求解即可，另外，程序中的 (double) 是强制类型转换，将原来为 float 型的 y 强制转换成 double 型变量。

故本题答案为 C。

172、以下程序的输出结果是()。

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

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

【解析】：本题的考查点是格式控制字符的应用。

printf()函数:当“格式控制”字符串中连续使用两个“%”符号时,表示要输出字符“%”,所以,a,c 两个变量的值并不带入输出字符串,最后的打印结果应当是 a=%d, b=%d。

故本题答案为 D。

173、以下程序的输出结果是()。

```
#include<stdio.h>

main ()
{
    printf ("%d\n",NULL);
}
```

- A 不确定的(因变量无定义)
- B 0
- C -1
- D 1

【解析】：本题的考查点是对‘ NULL’的理解。

NULL 在 C 编译器中是一个值为 0 的常量,它的定义在头文件“stdio.h”中,本题的程序在一开始就包含了这个头文件,所以,在打印的时候,将打印出 0。

故本题答案为 B。

174、以下程序的输出结果是()。

```
main()
{
```



```

    int k=4, m=1, p;
    p=func(k, m); printf("%d, ", p);
    p=func(k, m); printf("%d\n", p);
}

func(int a, int b)
{
    static int m=0, i=2;
    i+=m+1 ; m=i+a+b;
    return(m);
}

```

A 8, 17

B 8, 16

C 8, 20

D 8, 8

【解析】： 本题的考查点是函数的调用。

将 k 和 m 的值带入函数 func 中，第一次调用完 func() 后，其中 m 和 i 的值要保留，并在第二次调用时使用，这是因为两者都是局部静态变量，最后的结果是 8 和 17。

故本题答案为 A。

175、以下程序的输出结果为()。

```

main()
{
    char *alpha[6]={"ABCD", "EFGH", "IJKL", "MNOP", "QRST", "UVWX"};
    char **p ;
    int i;p=alpha;
    for(i=0; i<4 ; i++)
        printf("%s", p[i]);
}

```

```
printf("\n") ;  
}  
A  ABCDEFGHIJKL  
B  ABCD  
C  ABCDEFGHIJKLMNOP  
D  AEIM
```

【解析】： 本题的考查点是指向指针的指针。

指针数组*alpha[6]实际上相当于一个二维数组，而 p 是一个指向指针的指针，p=alpha 后，引用 p[x]便是引用数组*alpha[6]中的第 x 个数组元素的首地址，所以打印的结果应当为 ABCDEFGHIJKLMNOP。

故本题答案为 C。

176、 以下程序调用 findmax 函数求数组中值最大的元素在数组中的下标，请选择填空：

```
#include <stdio.h>  
  
findmax(int *s,int t,int *k)  
{  
    int p;  
    for(p=0,*k=p;p<t ;p++)  
        if(s[p]>s[*k]) ____;  
}  
  
main()  
{  
    int a[10],i,k;  
    for(i=0;i<10; i++) scanf("%d",&a[i]);  
    findmax(a,10,&k);  
    printf("%d,%d\n",k,a[k]);  
}
```

- A $k=p$
- B $*k=p-s$
- C $k=p-s$
- D $*k=p$

【解析】：本题的考查点是求数组中最大值的下标。

k 是一个指针，它的值是一个地址，要通过它为主函数的变量改变数据，必须使用 $*p$ ，这样就可以把一个下标数赋给 p 指针指向的那个内存单元，所以横线处应填入 $*k=p$ 。

故本题答案为 D。

177、以下的 for 循环：

```
for (x=0, y=0; (y!=123)&&(x<4); x++);
```

- A 是无限循环
- B 循环次数不定
- C 执行 4 次
- D 执行 3 次

【解析】：本题的考查点是 for 语句循环条件的判断。

循环条件是表达式 $(y \neq 123) \&\& (x < 4)$ ，因为 y 值始终为 0，所以表达式左边一定恒为 1，当 $x < 4$ 时，循环可以运行，当 $x=4$ 时，循环终止，所以这个循环一共可以执行 4 次。

故本题答案为 C。

178、若有说明：int i, j=7, *p=&i; 则与 $i=j$; 等价的语句是 ()。

- A $i=*p$;
- B $*p=&j$;
- C $i=&j$;
- D $i=**p$;

【解析】：本题的考查点是指针变量的引用。

p 的地址本身和 i 的地址一样, 所以 p 代表 i 中的数值, 而 &j 是 j 的地址, * &j 就是 j 中的数值, 所以, *p=&j 和 i=j 是等价的。

故本题答案为 B。

179、若有以下说明:

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

则值为 6 的表达式是()。

A *p+6

B *(p+6)

C *p+=5

D p+5

【解析】: 本题的考查点是通过指针引用数组元素。

选项 A, *p+6, 其值等于 1+6=7;

选项 B, *(p+6), 其值等于 a[6]=7;

选项 C, p+=5, 其值等于 *p=*p+5=1+5=6;

选项 D, p+5 是个地址值。

故本题答案为 C。

180、下面程序的输出是

```
main ()
{
    int k=11;
    printf("k=%d, k=%o, k=%x\n", k, k, k);
}
```

A k=11, k=12, k=11

B k=11, k=13, k=13

C k=11, k=013, k=0xb

D k=11, k=13, k=b

【解析】：本题的考查点是格式字符。

在 C 语言格式字符的输出中，“%d”是以带符号的十进制形式输出整数，即 k=11；“%o”是以 8 进制无符号形式输出整数（不输出前导符 0），即 k=13；“%x”是以 16 进制无符号形式输出整数（不输出前导符 0x），即 k=b。

故本题答案为 D。

181、下面程序的输出是（ ）。

```
main()
{
    int t=1;
    fun(fun(t)) ;
}
fun(int h)
{
    static int a[3]={1,2,3} ;
    int k;
    for (k=0;k<3;k++)a[k]+=a[k]-h;
    for (k=0 ;k<3;k++)printf("%d,",a[k]);
    printf("\n") ; return(a[h]);
}
```

- A 1, 3, 5,
1, 5, 9,
- B 1, 3, 5,
1, 3, 5,
- C 1, 3, 5
0, 4, 8
- D 1, 3, 5
-1, 3, 7

【解析】：本题的考查点是函数的反复调用。

在子函数 fun() 中，第一次执行使得 a 数组的 3 个元素的值都翻倍后减一，即为 1, 3, 5，此时 h 的值为 1，a[h] 即 a[1] 的值 3，返回作为 fun() 第二次执行的实参，函数第二次的执行使得各个数组元素的值分别变为 -1, 3, 7。

故本题答案为 D。

182、以下程序的输出结果是()。

```
#include <string.h>

main()
{
    char *a="abcdefghi"; int k;
    fun(a); puts(a) ;
}

fun (char *s)
{
    int x, y ; char c;
    for(x=0, y=strlen(s)-1; x<y ;x++,y--)
        {c=s[y] ;s[y]=s[x] ;s[x]=c;
        }
}

A ihgfedcba
B abcdefghi
C abcdedcba
D ihgfefghi
```

【解析】：本题的考查点是字符串的应用。

观察子函数 fun() 可知，它是实现一个数组首末元素位置互相依次交换。联系主函数，可以看出，是对字符串中的元素依次进行首末位置的对调。

故本题答案为 A。

183、以下程序的输出结果是()。

```
main()
{
    int n[3][3], i, j;
    for(i=0; i<3; i++)
        for(j=0; j<3; j++) n[i][j]=i+j;
    for(i=0; i<2; i++)
        for(j=0; j<2; j++) n[i+1][j+1]+=n[i][j];
    printf("%d\n", n[i][j]);
}
```

A 14

B 0

C 6

D 值不确定

【解析】：本题的考查点是二维数组的运算。

观察程序段可知，第一个 for 循环语句是给数组中的各元素赋值，第二个 for 循环语句是给数组中的元素进行重新赋值，得到数组中的最大元素值为 6。

故本题答案为 C。

184、以下程序的输出结果是()。

```
main()
{
    union {
        char i[2];
        int k ;
    } r;
    r.i[0]=2;r.i[1]=0;
```

```
printf("%d\n", r.k);  
}
```

A 2

B 1

C 0

D 不确定

【解析】：本题的考查点是共用体类型数据的特点。

根据共用体的定义可知:共用体 r 的成员 k 和成员 i[2]是共用同一段内存空间,所以,当程序给 r.i[0]赋值后,实际上,共用体成员 k 的值也确定了,为 2。所以打印输出的结果应当为 2。

故本题答案为 A。

185、设有以下定义和语句,则输出的结果是()。(用 small 模式编译,指针变量占 2 个字节)

```
struct date  
{  
    long *cat;  
    struct date*next ;  
    double dog;  
}too ;  
printf("%d", sizeof(too)) ;
```

A 20

B 16

C 14

D 12

【解析】：本题的考查点是计算结构体的长度。

结构体变量所占内存长度是各成员占的内存长度之和。每个成员分别占有其自己的内存单元。double 占 8 个字节,指针变量占 2 个字节,2+2+8=12,所以,输出结果是 12。

故本题答案为 D。

186、以下程序的输出结果是()。

```
#define f(x) x*x
main()
{
    int a=6, b=2, c;
    c=f(a)/f(b) ;
    printf("%d\n", c) ;
}
```

A 9

B 6

C 36

D 18

【解析】： 本题的考查点是带参数的宏定义。

观察程序段可知,对带参数的宏的展开只是将语句中的宏名后面括号内的实参字符串代替#define 命令行中的形参。本题中 $c=f(a)/f(b)$ 用 `#define f(x) x*x` 代替得 $c=f(6)/f(2)$, 即 $c=6*6/2*2$, 结果为 36。

故本题答案为 C。

187、以下程序的输出结果是()。

```
main()
{
    char *s="12134211"; int v[4]={0,0,0,0} ,k, i ;
    for (k=0 ; s[k] ;k++)
    {
        switch(s[k])
        {
```

```
        case '1' :i=0;
        case '2' :i=1;
        case '3' :i=2;
        case '4' :i=3;
    }
    v[i]++;
}
for(k=0; k<4; k++) printf("%d ", v[k]);
}
```

A 4 2 1 1

B 0 0 0 8

C 4 6 7 8

D 8 8 8 8

【解析】： 本题的考查点是 switch 语句的应用。

在每次执行 switch 后面括号内的表达式 s[k] 时，要先取当前 s[k] 的值与 case 后的字符常量进行比较，找到与之匹配的分支后，即转去执行该分支中的语句。switch 语句中，某分支执行完毕，若未遇到 break 语句，则接着执行剩余的其它分支，直到遇到 break 语句。此题中，因为没有 break 语句，所以总会执行到 i=3，即总是 v[3] 在递加。

故本题答案为 B。

188、有如下程序

```
main()
{
    int n = 9;
    while(n>6)
    {
        n--;
```

```
printf("%d", n);  
}  
}
```

该程序的输出结果是()。

- A 987
- B 876
- C 8765
- D 9876

【解析】：本题考查点是 while 语句循环次数的问题。

符合 $n > 6$ 的 n 只有 7, 8, 9, 所以循环次数为 3。选项 C、D 可以排除。又因 n 在输出之前要执行 $n--$, 所以输出结果只能是 876。

本题答案为 B。

189、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、 $x < 10 \ \&\& \ x > 5$

【解析】：本题考试的是与、或运算符。

A 选项的写法是数字中的写法, 并不是 C 语言中表达式的正确写法, 它的值与 x 无关, 并且恒为真; 与运算符中, 只要有一个为假, 那么整个表达式都为假, 因此与 x 的取值有关, B、D 选项错误; 或运算符中, 两个都为假, 那么整个表达式都为假, 因此也与 x 的取值有关, C 选项错误。

故本题答案为 A。

190、设变量 m 为 float 类型, 变量 n 为 int 类型, 则以下能实现将 m 中的数值保留小数点后两位, 第三位进行四舍五入运算的表达式是()。

- A、 $n = m * 100 + 0.5, m = n / 100.0$
- B、 $m = (m * 100 + 0.5) / 100.0$

C、 $n=m/100+0.5$, $m=n*100.0$

D、 $m=m*100+0.5/100.0$

【解析】：本题考查的是运算表达式。

A 选项中 $m*100$ 是将原数扩大 100 倍，加上 0.5 是对第三位进行四舍五入，并转换为整型赋值给 n，最后除以 100.0 是保留两位小数点并转换为 float 型；B 选项中未执行四舍五入操作，故 B 选项错误；c 选项并不是对第三位进行四舍五入，故 C 选项错误；D 选项中将原数扩大了 100 倍还要+0.0005，与原题不符，故 D 选项错误。

故本题答案为 A。

191、以下叙述正确的是（ ）。

A、C 语言中的语句之间必须用分号作为分隔符

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 程序总是以 main() 作为程序执行的起始行

【解析】：本题考查的是函数。

C 程序中总是从 main() 函数开始执行；main() 函数后面的一对圆括号是不能省略；所有的 C 语句都以分号结尾。故本题答案为 D。

192、C 语言中，最基本的数据类型包括（ ）。

A、整型、实型、字符型

B、整型、字符型、数组

C、整型、实型、逻辑型

D、整型、实型、结构体

【解析】：本题考查的是数据类型。

数据类型分为基本类型、构造类型、指针类型、空类型。

基本类型包括：整型、字符型、实型（单精度型、双精度型）、枚举度型；构造类型包括：数组类型、结构体类型、共用体类型故本题答案为 A。

193、以下非法的定义语句是()。

- A、 unsigned int a=-100u;
- B、 long b=0L;
- C、 short c2=0123;
- D、 int d=0x0;

【解析】： 本题考查的是数据类型。

A 选项中是定义的无符号整型类型，与-100u 不符，故 A 错误。故本题答案为 A。

194、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)

【解析】： 本题考查的是表达式。

A 选项表达式可以分解为 $y=y\%x=5, x=x\%5=2, 5-2=3$, B 选项中分解为 $x=x\%5=2, y=y\%2=0$, 故 B 选项错误; C 选项分解为 $y=y\%(x-x\%5)=2$, 故 C 选项错误, D 选项分解为 $y=y\%(x-x\%5)=2$, 故 D 选项错误。

故本题答案为 A。

195、设有定义

```
int n = 1234;
```

```
double x = 3.1415;
```

则语句

```
printf("%3d,%1.3f\n", n, x);
```

的输出结果是()。

- A、 1234, 3.142
- B、 123, 3.142

C、1234, 3. 141

D、123, 3. 141

【解析】：本题考查的是格式控制。

本题中已定义好了 n、x 的值，%nd，若 n 的值小于该数值的长度，则按该数值输出，若 n 的值大于该数值的长度则在数值的右边补 0；%1.3f 中 3 代表小数位的位数，要遵守四舍五入的规则。

故本题答案为 A。

196、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, 1, 2, 0

B、0, 0, 0, 3

C、1, 1, 2, 0

D、编译有错

【解析】：本题考查的 if 语句。

else 语句前缺少 if 语句，所以编译时会出错。故本题答案为 D。

197、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||x-y>0)

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

【解析】：本题考查的是表达式。

表达式中 0 为假，非 0 为真，因此题目中只有当 $x=y$ 时才会为假，其余都是真。A 选项中，或运算是当两个条件都为假是才为假，只要有一个为真就是真，而只有当 $x=y$ 时，两个条件都为假。

故本题答案为 A。

198、若有定义：

```
float x=1.5;
```

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

则正确的 switch 语句是()。

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

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

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

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

【解析】：本题考查的是 switch 语句。

在 switch 语句中紧跟在 switch 后一对圆括号中的表达式可以是整型表达式及字符型表达式等，case 后的常量表达式的类型必须与 switch 后圆括号中的表达式类型相同。

故本题答案为 A。

199、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、65,68

B、A,68

C、A,B

D、65,66

【解析】：本题考查的是字符。

由题意可知输出的是整型类型，c2-2 是 B，其 ASCII 码值为:66。故本题答案为 D。

200、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==0

B、x/2

C、x%2!=0

D、x%2==1

【解析】：本题考查的是表达式。

由题意可知，x 为大于 1 的奇数故 x%2 的值为 1，故 x%2==0 为假，其值为 0。故本题答案为 A。