

## C 语言精选 200 题

备注：视频解析都上传到（刷题班）或（全程班）或（冲刺班）或（寒暑假班）对应的课表，可以打开小黑课堂 APP，我的课程目录【最下面观看视频讲解】。

1、C 语言源程序名的后缀是（ ）。

- A、.exe
- B、.c
- C、.obj
- D、.cp

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

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

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

5、以下（ ）是不合法的 C 语言常量。”

- A、0xFF
- B、1.2e0.5
- C、2L
- D、'72'

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

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

8、以下选项中合法的用户标识符是( )。

A、long

B、\_2Test

C、3Dmax

D、A.dat

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

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、 $k+=n+1$

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

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

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

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

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

16、以下选项中不属于 C 语言程序运算符的是( )。

- A、<>
- B、sizeof
- C、()
- D、&&

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

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);

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;

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);

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

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

23、设 ch 是 char 型变量，其值为 A，且有下面的表达式

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

上面表达式的值是( )。

A、A

B、a

C、Z

D、z

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

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

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

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

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

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

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

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

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

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

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

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

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)

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++];);

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

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

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

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

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

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

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

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

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

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

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

49、库函数 rand()的功能是产生一个在 0 ~ 32767 之间的随机数。若要用此函数随机产生一个 0 ~ 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

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

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

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

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'

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

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

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']

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

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];`

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

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

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

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

63、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方式是( )。

- A、 \*(w+3)
- B、 \*(w+1)[4]
- C、 \*\* (w+1)
- D、 \*(&w[0][0]+1)

64、若有定义语句：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]

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};

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

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

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'

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");

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

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

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);

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

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

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);

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

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

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

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

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

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,

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

103、C 语言运算对象必须是整型的运算符是( )。

A、 %

B、 /

- C、 =
- D、 <=

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

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

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);

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

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

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

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

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

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

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

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

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

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"

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))

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

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")

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[ ]) {...}

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

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

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

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

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

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

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

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

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,

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,

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('#');
```

```
        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#

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++;

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;

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

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

137、 以下\_\_\_是不合法的 C 语言常量。

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

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

139、 若变量已正确说明为 float 类型， 要通过语句  $\text{scanf}(\text{"\%f\%f\%f"}, \&a, \&b, \&c)$ ; 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

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

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

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

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

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')



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)

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

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

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

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

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

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

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

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

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,

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

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;

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

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

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,

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

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

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；

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

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

165、以下不属于 C 语言位运算符的是( )。

A、!

B、|

C、^

D、~

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

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

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

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、 $p \rightarrow n$

D、 $(*p).n$

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

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

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

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

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

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

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

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

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 ;

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

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

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

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

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

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

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

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

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

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

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、 $x < 10 \ \&\& \ x > 5$

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、 $m = m * 100 + 0.5 / 100.0$

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

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

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;`

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、 $y\%=(x-x\%5)$

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

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

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)`

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"); }
```

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

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、 $x\%2==1$