

一、判断题：

1. C 程序的基本组成单位是函数 **T**
2. C 程序可以由一个或多个函数组成 **T**
3. C 语言可以不用编译就能被计算机识别执行 **F** (C 语言必须编译成能被计算机识别的机器代码才能执行)
4. main()函数必须放在其它函数之前 **F** (任意位置)
5. 每个 C 程序中都必须要有个 main()函数 **T**
6. 在 C 程序中 main()函数的位置是固定的 **F** (任意位置)
7. C 程序中的 main()函数必须放在程序的开始位置。 **F** (任意位置)
8. C 语言编译时不检查语法 **F** (编译即进行语法检查)
9. C 程序中注释部分可以出现在程序中任意合适的地方 **T**
10. C 程序中一行可以写多条语句 **T**
11. 分号是 C 语句之间的分隔符，不是语句的一部分 **F** (是语句不可缺少的部分)
12. 花括号“{”和“}”只能作为函数体的定界符 **F** (可以转义当作字符)
13. C 程序中的每行只能写一条语句 **F** (一行可有多个语句)
14. C 程序的执行总是从 main 函数开始，在 main 函数结束 **T**
15. C 程序的执行总是从 main 函数开始，在程序的最后一个函数中结束 **F**
16. ++(i+1);是非法的赋值语句 **T** (++只能对变量操作，而 i+1 不是变量)
17. 用户标识符中不可以出现中划线，但可以出现下划线 **T** (用户标识符由字母、数字、下划线组成，且以字母或下划线开头，不能有关键字)
18. 用户标识符中可以出现下划线，但不可以放在用户标识符的开头 **F**
19. C 语言中，Area 与 area 是不同的标识符 **T** (大小写有区别)
20. C 语言中有逻辑类型 **F** (C 语言中既没有逻辑类型也没有集合类型)
21. int a=b=0;是正确的定义语句。 **F** (b 未定义)
22. 1.2E0.5 是符合 C 语言语法的实型常量 **F** (E 后面指数必须为整数)
23. 假定 x 和 y 为 double 型，则表达式 x=2,y=x+3/2 的值是 3.500000 **F** (3/2=1，因为都是整型)
24. 可以用关系运算符对字符串的大小进行比较 **F** (可以用字符串比较函数 strcmp 格式：strcmp(字符数组名 1，字符数组名 2))
25. 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 **F**
26. 设有定义语句：char b= '\123';则变量 b 包括 4 个字符 **F** (八进制 123 对应十进制 83 对应字符 S)
27. 若有定义语句：char s[10]="1234567\0\0";，则 strlen(s)的值是 9 **F** (结束符之前)
28. a=b+c=1 是正确的赋值表达式 **F**
29. 若变量 a、i 已正确定义，且 i 已正确赋值，则 a=a++=5; 是正确的赋值语句 **F**
30. 除逗号运算符外，赋值运算符的优先级最低 **T**
31. C 语言中，运算对象必须是整型数据的运算符是 % **T**
32. 对于单目运算符++、--，它们的运算对象可以是任何变量和常量 **F** (只能是变量)
33. 表达式：10! =9 的值是 true **F** (1)
34. 表达式 1||2||3||4 的值是 1。 **T**
35. 若变量 c 为 char 类型，则表达式(c>= 'a')&&(c<= 'z') 能正确判断出 c 为小写字母 **T**
36. 若变量 c 为 char 类型，则表达式 'a'<=c<= 'z' 能正确判断出 c 为小写字母 **F**
37. sizeof(float)的值是 4 **T**
38. sizeof(double)是一种函数调用 **F** (是一个整型表达式)
39. if(x<y) {x++;y++;} 是正确的 if 语句 **T**
40. if(x>y); 是正确的 if 语句 **T** (空语句)

41. 用 do-while 语句构成的循环,在 while 后的表达式为零时结束循环 **T**
42. do-while 语句构成的循环只能用 break 语句退出 **F**
43. 对 for(表达式 1; ; 表达式 3)可理解为 for(表达式 1; 0; 表达式 3)**F(应为 for(表达式 1; 1; 表达式 3))**
44. break 语句只能用于 switch 语句体中 **F**
45. continue 语句的作用是: 使程序的执行流程跳出包含它的所有循环 **F(continue 是跳出本次循环)**
46. 在循环体内使用 break 语句和 continue 语句的作用相同 **F(break 是跳出当前循环, 即循环结束; continue 是跳出本次循环, 直接进入下一次的循环)(在 switch 中作用相同)**
47. 当程序执行中, 数组元素的下标超出所定义的下标范围时, 系统将给出“下标越界”的出错信息 **F(系统不对数组下标做越界检查)**
48. 假定 int 类型变量占用两个字节, 其有定义: int x[10]={0,2,4};, 则数组 x 在内存中所占字节数是 6**F**
49. 若有定义: int a[2][3];对 a 数组元素正确引用的是 a[2][3]**F**
50. 若有说明 int s[3][4]={0};则只有元素 s[0][0] 可得到初值 0**F**
51. **不能在赋值语句中通过赋值运算符"="对字符型数组进行整体赋值 T**
52. 字符型数组中可以存放字符串 **T**
53. 可以对整型数组进行整体输入、输出 **F**
54. **函数中的形式参数是局部变量 T**
55. 在不同的函数中可以使用相同名字的变量 **T**
56. 在一个函数内定义的变量只在本函数范围内有效 **T**
57. 函数的定义和函数的调用均可以嵌套 **F (定义不能嵌套)**
58. 函数的定义可以嵌套, 但函数的调用不可以嵌套 **F (定义不能嵌套, 调用可以嵌套)**
59. 函数的形参和实参分别占用不同的存储单元 **T**
60. C 语言中, 每个函数都可以被其它函数调用(包括 main 函数)**F(main 函数是不能被任何函数调用)**
61. C 语言中, 在一个函数内部可以定义另一个函数 **F(函数不能嵌套定义)**
62. 实参和与其对应的形参共同占用一个存储单元 **F**
63. 只有当实参和与其对应的形参同名时才共占用存储单元 **F**
64. 若有以下函数调用语句: fun(a+b,(x,y),fun(n+k,d,(a,B)));在此函数调用语句中实参的个数是 6**F(应是 3)**
65. 用户定义的函数中可以没有 return 语句 **T**
66. 用户定义的函数中可以有多条 return 语句, 以便可以调用一次返回多个函数值 **F**
67. 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 **T**
68. 若用数组名作为实参, 则传递给形参的是数组中全部元素的值 **F**
69. C 语言中形参的默认存储类别是自动(auto)**T**
70. C 语言中, 宏替换没有数据类型限制 **T**
71. C 语言中, 宏名必须用大写字母表示 **F**
72. C 程序对预处理命令行的处理是在程序执行的过程中进行的 **F(编译前由预处理器处理)**

二、单选题

1. 以下说法中正确的是(**C**)
 - A) C 语言程序总是从第一个的函数开始执行
 - B) 在 C 语言程序中,要调用的函数必须在 main()函数中定义
 - C) C 语言程序总是从 main()函数开始执行
 - D) C 语言程序中的 main()函数必须放在程序的开始部分

2. 下选项中不是 C 语句的是(C)。

- A) {int i; i++; printf("%d\n",i);}
- B) ;
- C) a=5,c=10
- D) {;}

3. 以下选项中不能作为 C 语言合法常量的是

(A)

- A) 'cd'
- B) 0.1e+6
- C) "\a"
- D) '\011'

4. 若 x,a,b 和 c 均是 int 型变量, 则执行表达式 x=(a=1,b=2)后 x 的结果为(B)。

- A) 1
- B) 2
- C) 3
- D) 不确定

5. 以下能正确定义整型变量 a,b 和 c 并为其赋初值 1 的语句是(D)。

- A) int a=b=c=1; //(b、c 未定义)
- B) int a,b,c=1; // (a、b 未赋值)
- C) a=b=c=1;
- D) int a=1,b=1,c=1;

6. 字符串 "\t065\xff\n" 中的字符数(不算 '\0') 为(D)

- A) 5
- B) 14
- C) 8
- D) 4

7. 对两个静态函数 A 和 B 进行如下初始化:

```
char A[]="ABCDEF";
```

```
char B[]={ 'A','B','C','D','E','F' };
```

则下列叙述正确的是(D)。

- A) A 和 B 完全相同
- B) A 和 B 只是长度相等
- C) A 和 B 不相同, A 是指针数组
- D) A 数组长度比 B 数组长

8. 若有以下定义:

```
char a;int b;
```

```
float c;double d;
```

则表达式 a*b+d-c 值的类型为(D)

- A) float
- B) int
- C) char
- D) double

9. 在 C 语言中, 运算对象必须是整型数的运算符是(A)

- A) %
- B) /
- C) %和\
- D) **

10. 有以下程序

```
main()
```

```
{ int x, y, z;  
  x=y=1;
```

```

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

```

程序运行后的输出结果是

(**C**)

A) 2,3,3

B) 2,3,2

C) 2,3,1

D) 2,2,1

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

A) $x\%2==1$

B) $x/2$

C) $x\%2!=0$

D) $x\%2==0$

12. 以下程序的输出结果是(**C**)。

```

main()
{
    int x=10,y=3;
    printf("%d\n",y=x/y);
}

```

A) 0

B) 1

C) 3

D) 不确定的值

13. 若 a 为 int 类型, 且其值为 3, 则执行完表达式 $a+=a-=a*a$ 后, a 的值是(**C**)

A) -3

B) 9

C) -12

D) 6

14. 有以下程序

```

main()
{
    char a='a',b;
    printf("%c,",++a);
    printf("%c\n",b=a++);
}

```

程序运行后的输出结果是

(**A**)

A) b,b

B) b,c

C) a,b

D) a,c

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

A) 0

(短路特性)

B) 1

C) 2

D) 3

16. 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(**C**)

A) 不定值

B) 4

C) 3

D) 1

17. 有以下程序段

```
int k=0;
```

```
while(k=1)k++;
```

while 循环执行的次数是(A)

A) 无限次

B) 有语法错, 不能执行

C) 一次也不执行

D) 执行 1 次

18. 能正确表示逻辑关系 “ $a \geq 10$ 或 $a \leq 0$ ” 的 C 语言表达式是(D)

A) $a \geq 10$ or $a \leq 10$

B) $a \geq 0 \mid a \leq 10$

C) $a \geq 10 \&\& a \leq 0$

D) $a \geq 10 \mid a \leq 0$

19. 表示关系 $x \leq y \leq z$ 的 c 语言表达式为(A)

A) $(x \leq y) \&\& (y \leq z)$

B) $(x \leq y) \text{AND} (y \leq z)$

C) $(x \leq y \leq z)$

D) $(x \leq y) \& (y \leq z)$

20. 设有: $\text{int } a=1, b=2, c=3, d=4, m=3, n=3$; 执行 $(m=a>b) \mid (n=c>d)$ 后 n 的值为(A)。

A) 0

B) 1

C) 2

D) 3

21. 以下关于逻辑运算符两侧运算对象的叙述中正确的是

(D)

A) 只能是整数 0 或 1

B) 只能是整数 0 或非 0 整数

C) 可以是结构体类型的数据

D) 可以是任意合法的表达式

22. 若 x 和 y 都是 int 型变量, $x=100, y=200$, 且有下列的程序片段:

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

上面程序片段的输出结果是(A)

A) 200

B) 100

C) 100 200

D) 输出格式符不够, 输出不确定的值

23. 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

(B)

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

24. 以下程序的输出结果是(A)

```
main()
```

```
{ int k=17;
```

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

```
}
```

A) 17, 021, 0x11

B) 17, 17, 17

C) 17, 0x11, 021

D) 17, 21, 11

25. x、y、z 被定义为 int 型变量，若从键盘给 x、y、z 输入数据，正确的输入语句是(B)。

A) INPUT x、y、z;

B) scanf("%d%d%d",&x,&y,&z);

C) scanf("%d%d%d",x,y,z);

D) read("%d%d%d",&x,&y,&z);

26. 有以下程序

```
main()
```

```
{
```

```
    int i;
```

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

```
        switch(i)
```

```
        {
```

```
            case 0: printf("%d",i);
```

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

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

```
        }
```

```
}
```

程序运行后的输出结果是(C)

A) 022111

B) 021021

C) 000122

D) 012

27. C 语言中用于结构化程序设计的三种基本结构是(A)

A) 顺序结构、选择结构、循环结构

B) if、switch、break

C) for、while、do-while

D) if、for、continue

28. 在嵌套使用 if 语句时，C 语言规定 else 总是

(C)

A) 和之前与其具有相同缩进位置的 if 配对

B) 和之前与其最近的 if 配对

C) 和之前与其最近的且不带 else 的 if 配对

D) 和之前的第一个 if 配对

29. 设有以下程序片段:

```
switch(X)
```

```
{
```

```
    case 'A': printf("A");
```

```
    case 'B': printf("B");
```

```
    default: printf("error");
```

```
}
```

假设 X='A', 程序输出结果是(D)

A) A

B) B

C) error

D) AError

30. 读程序:

```
main()
```

```
{ int num=0;
```

```
    while (num<=2)
```

```
    { num++; printf("%d  ",num); }
}
```

上面程序的输出结果是(C)

- A) 1
- B) 2 2
- C) 1 2 3
- D) 1 2 3 4

31. 以下程序段的执行结果是(A)。

```
int x=5;
do{printf("%2d\n",x--);}
while(!x);
```

- A) 5
- B) 无任何输出
- C) 4
- D) 陷入死循环

32. 设变量已正确定义，则以下能正确计算 $f=n!$ 的程序段是

(D)

```
A) f=0;
for(i=1;i<=n;i++) f*=i;
B) f=1;
for(i=1;i<n;i++) f*=i;
C) f=1;
for(i=n;i>1;i++) f*=i;
D) f=1;
for(i=n;i>=2;i--) f*=i;
```

33. 以下程序的输出结果是(D)。

```
main()
{
    int a, b;
    for(a=1,b=1;a<=100;a++)
    {
        if(b>=10) break;
        if(b%3==1)
        {
            b+=3;
            continue;
        }
    }
    printf("%d\n",a);
}
```

- A) 101
- B) 6
- C) 5
- D) 4

34. 下列合法的数组定义是(D)。

```
A) int a[]="string";
B) int a[5]={0,1,2,3,4,5};
C) char a="string";
D) char a[]={0,1,2,3,4,5};
```

35. 以下定义语句中，错误的是(D)

```
A) int a[]={1,2};
B) char *a[3];
```

C) `char s[10]="test";`

D) `int n=5,a[n];`

36. 以下正确的数组定义语句是(**D**)。

A) `int y[1][4]={1,2,3,4,5};`

B) `float x[3][]={{1},{2},{3}};`

C) `long s[2][3]={{1},{1,2},{1,2,3}};`

D) `int m[1][4]={4};`

37. 以下程序的输出结果是(**C**)。

```
main()
{
    int i,a[10];
    for(i=9;i>=0;i--)
        a[i]=10-i;
    printf("%d%d%d",a[2],a[5],a[8]);
}
```

A) 258

B) 741

C) 852

D) 369

38. 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(**B**)

A) `a[0][4]`

B) `a[1][3]`

C) `a[0][3]`

D) `a[1][4]`

39. 以下数组定义中不正确的是(**D**)

A) `int a[2][3];`

B) `int b[][3]={0,1,2,3};`

C) `int c[100][100]={0};`

D) `int d[3][]={{1,2},{1,2,3},{1,2,3,4}};`

40. 以下程序的输出结果是(**A**)。

```
main()
{
    int a[4][4]={{1,3,5},{2,4,6},{3,5,7}};
    printf("%d%d%d%d%d\n",a[0][3],a[1][2],a[2][1],a[3][0]);
}
```

A) 0650

B) 1470

C) 5430

D) 输出值不定

41. 以下程序的输出结果是

(**B**)

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

A) "AAAA"

B) "BBB"

C) "BBBCC"

D) "CC"

42. 设有数组定义: `char array []="China";` 则数组 array 所占的空间为(**C**)

A) 4 个字节

B) 5 个字节

- C) 6 个字节
D) 7 个字节
43. 以下选项中，不能正确赋值的是(A)。
- A) `char s1[10];s1="Ctest";`
B) `char s2[]={ 'C','t','e','s','t' };`
C) `char s3[20]="Ctest";`
D) `char *s4="Ctest\n";`
44. 给出以下定义：
`char x[]="abedfeg";`
`char y[]={ 'a','b','c','d','e','f','g' };`
则正确的叙述为(C)。
- A) 数组 x 和数组 y 等价
B) 数组 x 和数组 y 长度相同
C) 数组 x 的长度大于数组 y 的长度
D) 数组 x 的长度小于数组 y 的长度
45. 若要求从键盘读入含有空格字符的字符串，应使用函数
(B)
- A) `getc()`
B) `gets()`
C) `getchar()`
D) `scanf()`
46. 有语句：
`char str1[10],str2[10]={"books"};`
则能将字符串 books 赋给数组 str1 的正确语句是(B)。
- A) `str1={"Books"};`
B) `strcpy(str1,str2);`
C) `str1=str2;`
D) `strcpy(str2,str1);`
47. 以下语句的输出结果是(A)
`printf("%d\n",strlen("\tc\065\xff\n"));`
- A) 5
B) 14
C) 8
D) 输出项不合法，无正常输出
48. 若有语句 `int *point, a=4;` 和 `point=&a;` 下面均代表地址的一组选项是(D)。
- A) `a, point, *&a`
B) `&*a, &a, *point`
C) `*&point, *point, &a`
D) `&a, &*point, point`
49. 设已有定义：`float x;`，则以下对指针变量 p 进行定义且赋初值的语句中正确的是(D)
- A) `float *p = 1024;`
B) `int *p = (float)x;`
C) `float p = &x;`
D) `float *p = &x;`
50. 设有语句：`int a=1,b=2,*p1=&a,*p2=&b;` 以下可使指针 p1 指向变量 b 的赋值语句是(C)。
- A) `p1=*p2`
B) `*p1=p2`
C) `p1=p2`
D) `*p1=*p2`
51. 有以下程序
`main()`

```

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

```

程序运行后的输出结果是(A)

- A) 1,2,3,4,5,6,7,8,9,0,
- B) 2,3,4,5,6,7,8,9,10,1,
- C) 0,1,2,3,4,5,6,7,8,9,
- D) 1,1,1,1,1,1,1,1,1,1,

52. 若已定义 char s[10]; 则在下面表达式中不能表示 s[1]的地址的是(B)。

- A) s+1
- B) s++(s 为常量,不能自增)
- C) &s[0]+1
- D) &s[1]

53. 在以下选项中, 操作不合法的一组是(B)。

- A) int x[6], *p; p=&x[0];
- B) int x[6], *p; *p=x;
- C) int x[6], *p; p=x;
- D) int x[6], p; p=x[0];

54. 以下叙述中正确的是(B)

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

55. 有以下程序

```

main()
{
    char s[]="159",*p;
    p=s;
    printf("%c",*p++);
    printf("%c",*p++);
}

```

程序运行后的输出结果是

(A)

- A) 15
- B) 16
- C) 12
- D) 59

56. 以下语句或语句组中, 能正确进行字符串赋值的是(D)

- A) char *sp; *sp="right!";
- B) char s[10]; s="right!";
- C) char s[10]; *s="right!";
- D) char *sp="right!";

57. 下面程序段的运行结果是(C)。

```

char *p="abcdefgh";
p+=2;
printf("%d\n", strlen(strcpy(p, "ABCD")));

```

- A) 6
- B) 12
- C) 4
- D) 7

58. 下面程序段的运行结果是(B)。

```
char str[ ]= "ABC", *p=str;
printf("%d\n", *(p+3));
```

- A) 67
- B) 0
- C) 字符'C'的地址
- D) 字符'C'

59. 以下程序运行后，输出结果是(C)

```
main()
{ char *s="abcde";
  s+=2;
  printf("%d\n", s);}
```

- A) cde
- B) 字符 c 的 ASCII 码值
- C) 字符 c 的地址
- D) 出错

60. 有以下程序：

```
main()
{ char *p[10]={ "abc", "aabdfig", "dcdbe", "abbd", "cd" };
  printf("%d\n", strlen(p[4]));
}
```

执行后输出结果是(A)

- A) 2
- B) 3
- C) 4
- D) 5

61. 若有语句：char *line[5];，以下叙述中正确的是(A)

- A) 定义 line 是一个数组，每个数组元素是一个基类型为 char 的指针变量
- B) 定义 line 是一个指针变量，该变量可以指向一个长度为 5 的字符型数组
- C) 定义 line 是一个指针数组，语句中的*号称为间址运算符
- D) 定义 line 是一个指向字符型函数的指针

62. 设有如下定义：char *aa[2]={ "abcd", "ABCD" }; 则以下说法中正确的是(D)。

- A) aa 数组元素的值分别是 "abcd" 和 "ABCD"
- B) aa 是指针变量，它指向含有两个数组元素的字符型一维数组
- C) aa 数组的两个元素分别存放的是含有 4 个字符的一维字符数组的首地址
- D) aa 数组的两个元素中各自存放了字符串 "abcd" 和 "ABCD" 的首地址

63. 以下函数调用语句中含有(B)个实参。

```
fun((exp1,exp2),(exp3,exp4,exp5));
```

- A) 1
- B) 2
- C) 4
- D) 5

64. 以下所列的各函数首部中，正确的是(C)

- 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)

65. C 语言中，函数值类型的定义可以缺省，此时函数值的隐含类型是(B)

- A) void
- B) int

- C) float
- D) double

66. 下面的函数调用语句中 func 函数的实参个数是

(A)

```
func(f2(v1, v2), (v3, v4, v5), (v6, max(v7, v8)));
```

- A) 3
- B) 4
- C) 5
- D) 8

67. 有以下程序

```
fun(int x, int y)
{ static int m=0, i=2;
  i+=m+1;    m=i+x+y;    return m;
}
main()
{ int j=1, m=1, k;
  k=fun(j,m);    printf("%d,",k);
  k=fun(j,m);    printf("%d\n",k);
}
```

执行后的输出结果是

(B)

- A) 5, 5
- B) 5, 11
- C) 11, 11
- D) 11, 5

68. 以下函数值的类型是

(A)

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

- A) int
- B) 不确定
- C) void
- D) float

69. 以下程序的输出结果是

(C)

```
fun(int x, int y, int z)
{ z=x*x+y*y; }
main()
{ int a=31;
  fun(5,2,a);
  printf("%d",a);
}
```

- A) 0
- B) 2
- C) 31
- D) 无定值

70. 有以下程序

```
void f(int x,int y)
{ int t;
```

```

    if(x<y){ t=x; x=y; y=t; }
}
main()
{ int a=4,b=3,c=5;
  f(a,b); f(a,c); f(b,c);
  printf("%d,%d,%d\n",a,b,c);
}

```

执行后输出的结果是(D)

- A) 3,4,5
- B) 5,3,4
- C) 5,4,3
- D) 4,3,5

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

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

72. 有以下程序

```

void f(int b[])
{ int i;
  for (i=2; i<6; i++)
    b[i] *= 2;
}
main()
{ int a[10]={1,2,3,4,5,6,7,8,9,10}, i;
  f(a);
  for (i=0; i<10; i++)
    printf("%d,", a[i]);
}

```

程序运行后的输出结果是(B)

- A) 1,2,3,4,5,6,7,8,9,10,
- B) 1,2,6,8,10,12,7,8,9,10,
- C) 1,2,3,4,10,12,14,16,9,10,
- D) 1,2,6,8,10,12,14,16,9,10,

73. 在 C 语言中，函数返回值的类型最终取决于

- (A)
- A) 函数定义时在函数首部所说明的函数类型
- B) return 语句中表达式值的类型
- C) 调用函数时主调函数所传递的实参类型
- D) 函数定义时形参的类型

74. 调用一个函数，此函数中没有 return 语句，下列说法正确的是：该函数(D)。

- A) 没有返回值
- B) 返回若干个系统默认值
- C) 能返回一个用户所希望的函数值
- D) 返回一个不确定的值(函数体可以没有 return 语句，程序执行到末尾为 “}” 时结束，然后返回调用函数，但没有确定的函数值返回)

75. 若有以下调用语句,则不正确的 fun 函数的首部是(D)

```

main()
{
    ...
    int a[50],n;
    ...
}

```

```

        fun(n, &a[9]);
        ...
    }
A) void fun(int m, int x[])
B) void fun(int s, int h[41])
C) void fun(int p, int *s)
D) void fun(int n, int A)
76. 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

(**C**)

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

77. 以下程序的输出结果(**B**)

```

#include<stdio.h>
sub(int x,int y,int * z)
{
  *z=y-x;
}
main()
{
  int a, b, c;
  sub(10,5,&a);
  sub(7,a,&b);
  sub(a,b,&c);
  printf("%d,%d,%d\n", a,b,c);
}

```

- A) 5,2,3
- B) -5,-12,-7
- C) -5,-12,-17
- D) 5,-2,-7

78. 已定义以下函数

```

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

```

fun 函数返回值是

(**B**)

- A) 不确定的值
- B) 一个整数
- C) 形参 p 中存放的值
- D) 形参 p 的地址值

79. 已定义以下函数:

```

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

```

函数的功能是

(**A**)

- 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'

80. 有以下程序

```
#include <stdio.h>
int fun(int a, int b)
{
    if (b==0)    return a;
    else        return (fun(--a, --b));
}
main()
{
    printf("%d\n", fun(4, 2));
}
```

程序的运行结果是(**B**)

- A) 1
- B) 2
- C) 3
- D) 4

81. 以下程序的输出结果是

(**B**)

```
int a,b;
void fun()
{
    a=100; b=200;
}
main()
{
    int a=5, b=7;
    fun();
    printf("%d%d\n", a,b);
}
```

- A) 100200
- B) 57
- C) 200100
- D) 75

82. 以下程序的输出结果是

(**D**)

```
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

83. 下面程序的输出是(**C**)

```
fun3(int x)
```

```
{ static int a=3;
  a+=x;
  return(a);}
```

```
main()
{ int k=2, m=1, n;
  n=fun3(k);
  n=fun3(m);
  printf("%d\n",n);}
```

- A) 3
- B) 4
- C) 6
- D) 9

84. 若有定义: int a=8, b=5, c;, 执行语句 c=a/b+0.4;后, c 的值为(**B**)

- A) 1.4
- B) 1
- C) 2
- D) 2.00000

85. 以下合法的 C 语言赋值语句是(**D**)。

- A) a=b=58
- B) k=int(a+b);
- C) a=58,b=59
- D) --i;

86. 数字字符'0'的 ASCII 值为 48, 若有以下程序

```
main()
{
  char a='1',b='2';
  printf("%c",b++); (输出后 b 为字符 3)
  printf("%d\n",b-a);
}
```

程序运行后的输出结果是(**C**)

- A) 3,2
- B) 50,2
- C) 2,2
- D) 2,50

87. 有以下程序段

```
int a, b, c;
a=10; b=50; c=a;
if (a>b) a=b, b=c; c=a;
printf("a=%d b=%d c=%d\n", a, b, c);
```

程序的输出结果是(**A**)

- A) a=10 b=50 c=10
- B) a=10 b=50 c=30
- C) a=10 b=30 c=10
- D) a=50 b=30 c=50

88. 下面程序段的运行结果是(**C**)

```
char a[] = "language", *p;
p=a;
while (*p != 'u') { printf("%c", *p-32); p++; }
```

- A) LANGUAGE
- B) language
- C) LANG

89. 设有定义: `int a; float b;` 执行 `scanf("%2d%f",&a,&b);` 语句时, 若从键盘输入 876 543.0<回车>, a 和 b 的值分别是

(B)

- A) 876 和 543.000000
- B) 87 和 6.000000
- C) 87 和 543.000000
- D) 76 和 543.000000

90. 设变量定义为 `int A=5,B=6;` 表达式 `(++A==B--)?++A:--B` 的值是(C)。

- A) 5
- B) 6
- C) 7
- D) 8

91. 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;
```

则数值为 4 的表达式是(D)

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

92. 有以下程序

```
#include "stdio.h"
int abc(int u,int v);
main ()
{
    int a=24,b=16,c;
    c=abc(a,b);
    printf("%d\n",c);
}
int abc(int u,int v)
{
    int w;
    while(v)
        { w=u%v; u=v; v=w }
    return u;
}
```

输出结果是(C)

- A) 6
- B) 7
- C) 8
- D) 9

93. 有以下程序

```
main()
{
    char a,b,c,d;
    scanf("%c,%c,%d,%d",&a,&b,&c,&d);
    printf("%c,%c,%c,%c\n",a,b,c,d);
}
```

若运行时从键盘上输入: 6,5,65,66<回车>, 则输出结果是(A)(注: 字符'A'的 ASCII 码为 65)

- A) 6,5,A,B
- B) 6,5,65,66
- C) 6,5,6,5
- D) 6,5,6,6

94. 下列程序的输出结果是(D)

```
main()
{ char *p1,*p2,str[50]="xyz";          xycbBCD
  p1="abcd";                          cd
  p2="ABCD";                          BCD      连接      cdBCD
  strcpy(str+2, strcat(p1+2,p2+1));
  printf("%s",str); }
```

- A) xyabcAB
- B) abcABz
- C) ABabcz
- D) xycdBCD

95. 有以下程序

```
#include <stdio.h>
main()
{ int a[]={1,2,3,4,5,6,7,8,9,10,11,12}, *p=a+5,*q=NULL;
  *q=*(p+5);
  printf("%d %d\n",*p,*q);
}
```

程序运行后的输出结果是(A)

(题目中没有给 q 分配存储单元，只是简单的给它赋了一个值，所以程序的运行结果是 611NULLpointerassignment，也就是运行后报错。)

- A) 运行后报错
- B) 6 6
- C) 6 11
- D) 5 10

96. 有以下程序

```
#include <stdio.h>
main()
{ int i,j, m=55;
  for(i=1;i<=3;i++)
    for(j=3;j<=i;j++) m=m%j;
  printf("%d\n", m);
}
```

程序的运行结果是(B)

- A) 0
- B) 1
- C) 2
- D) 3

97. 有以下程序

```
main()
{ int a=5,b=4,c=3,d=2;
  if(a>b>c)
    printf("%d\n",d);
  else if((c-1>=d)==1)
    printf("%d\n",d+1);
  else
    printf("%d\n",d+2);
}
```

执行后输出的结果是(B)

- A) 2
- B) 3
- C) 4

D) 编译时有错，无结果

98. 若以下选项中的变量已正确定义，则正确的赋值语句是(C)

A) $x1=26.8\%3$;

B) $1+2=x2$

C) $x3=0x12$;

D) $x4=1+2=3$;

99. 在执行以下程序时,如果从键盘上输入 ABCdef<回车>,则输出是(B)

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

```
main()
```

```
{ char    ch;
  while((ch=getchar())!='\n')
  {   if(ch>='A' && ch<='Z') ch=ch+32;
      else if(ch>='a' && ch<='z') ch=ch-32;
      printf("%c",ch);
  }
  printf("\n");
}
```

A) ABCdef

B) abcDEF

C) abc

D) DEF

100. 有以下程序

```
#include <string.h>
```

```
main()
```

```
{
    char  p[]={'a','b','c'}, q[10]={'a','b','c'};
    printf("%d %d\n",strlen(p),strlen(q));
}
```

以下叙述中正确的是

(B)

A) 在给 p 和 q 数组置初值时，系统会自动添加字符串结束符，故输出的长度都为 3

B) 由于 p 数组中没有字符串结束符，长度不能确定；但 q 数组中字符串长度为 3

C) 由于 q 数组中没有字符串结束符，长度不能确定；但 p 数组中字符串长度为 3

D) 由于 p 和 q 数组中都没有字符串结束符，故长度都不能确定

101. 当变量 c 的值不为 2、4、6 时，值也为“真”的表达式是

(B)

A) $(c==2)|| (c==4)|| (c==6)$

B) $(c>=2 \ \&\& \ c<=6)|| (c!=3)|| (c!=5)$

C) $(c>=2 \ \&\& \ c<=6)\&\&!(c\%2)$

D) $(c>=2 \ \&\& \ c<=6)\&\&(c\%2!=1)$

102. 以下程序的输出结果是(A)

```
main()
```

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

A) 2 0

B) 3 1

C) 3 2 1

D) 2 1 0

103. 有以下程序：

```
main()
```

```
{  int x=102, y=012;
    printf("%2d,%2d\n",x,y);
}
```

执行后输出结果是(C)(当域宽小于实际宽度时域宽不起作用，按实际宽度输出)

- A) 10,01
- B) 02,12
- C) 102,10
- D) 02,10

104. 以下程序的输出结果是

(C)

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

- A) 1,5,9
- B) 1,4,7
- C) 3,5,7
- D) 3,6,9

105. 若有以下定义和语句:

```
char s1[ ]="12345",s2[ ]="1234";
printf("%d\n",strlen(strcpy(s1,s2)));
```

则输出结果是(A)

- A) 4
- B) 5
- C) 9
- D) 10

106. 以下不能定义为用户标识符是(D)

- A) man
- B) _0
- C) _int
- D) sizeof

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

- A) scanf("%2d%2d%2d",&i,&j,&k);
- B) scanf("%d %d %d",&i,&j,&k);
- C) scanf("%d,%d,%d",&i,&j,&k);
- D) scanf("i=%d,j=%d,k=%d",&i,&j,&k);

108. 下面程序段中，输出的“*”的个数是(C)。

```
char *s= "\039\tcac";
for( *s !='\0'; s++) printf(" * ");
```

- A) 9
- B) 5
- C) 6
- D) 7

109. 有以下程序

```
point(char *p) {  p+=3;  }
main()
{  char  b[4]={'a','b','c','d'}, *p=b;
    point(p);
    printf("%c\n",*p);
}
```

程序运行后的输出结果是(A)

- A) a
- B) b
- C) c
- D) d

110. 设有 `int x=11`; 则表达式 `(x++ * 1/3)` 的值是(A)

- A) 3
- B) 4
- C) 11
- D) 12

111. 以下能正确定义一维数组的选项是(B)

- A) `int a[5]={0,1,2,3,4,5};`
- B) `char a[]={0,1,2,3,4,5};`
- C) `char a={'A','B','C'};`
- D) `int a[5]="0123";`

112. 执行下述程序片段时的输出结果是(C)。

```
int x,y;
x=13;
y=5;
printf("%d",x%=(y/=2));
```

- A) 3
- B) 2
- C) 1
- D) 0

113. 以下程序执行后 `sum` 的值是(C)。

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

- A) 15
- B) 14
- C) 不确定
- D) 0

114. 以下叙述中正确的是(D)

- A) 输入项可以是一个实型常量,如:`scanf("%f",3.5);`
- B) 只有格式控制,没有输入项,也能正确输入数据到内存,例如:`scanf("a=%d,b=%d");`
- C) 当输入一个实型数据时,格式控制部分可以规定小数点后的位数,例如:`scanf("%4.2f",&f);`
- D) 当输入数据时,必须指明变量地址,例如:`scanf("%f",&f);`

115. 有以下程序

```
main()
{
  char a1='M',a2='m';
  printf("%c\n",(a1, a2));
}
```

以下叙述中正确的是

(B)

- A) 程序输出大写字母 M
- B) 程序输出小写字母 m
- C) 格式说明符不足, 编译出错
- D) 程序运行时产生出错信息

116. 若有说明: `int a[10]`; 则对 `a` 数组元素的正确引用是(D)

- A) a[10]
- B) a[3.5]
- C) a(5)
- D) a[10-10]

117. 请读下面的程序。

```
#include <stdio.h>
main()
{
    int a,b;
    for(a=1,b=1;a<=100;a++)
    {
        if(b>=20) break;
        if(b%3==1)
        {
            b+=3;
            continue;
        }
        b-=5;
    }
    printf("%d\n",a);
}
```

上面程序的输出结果是(**B**)。

- A) 7
- B) 8
- C) 9
- D) 10

118. 有以下程序

```
#include <string.h>
main()
{ char p[20]={'a','b','c','d'},q[]="abc",r[]="abcde";
  strcpy(p+strlen(q),r); strcat(p,q);
  printf("%d %d\n",sizeof(p),strlen(p));
}
```

程序运行后的输出结果是 **abcabcdeabc**

(**C**)

- A) 20 9
- B) 9 9
- C) 20 11
- D) 11 11

119. 以下程序的输出结果是

(**D**)

```
main()
{ int a=3;
  printf("%d\n",(a+=a-=a*a));
}
```

- A) -6
- B) 12
- C) 0
- D) -12

120. 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**)

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


```
}

```

- A) 31
- B) 13
- C) 10
- D) 20

125. 下面函数的功能是(A)

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

- A) 求字符串的长度
- B) 比较两个字符串的大小
- C) 将字符串 x 复制到字符串 y
- D) 将字符串 x 连接到字符串 y 后面

126. 设有定义: int a, *pa=&a; 以下 scanf 语句中能正确为变量 a 读入数据的是(A)

- A) scanf("%d",pa);
- B) scanf("%d",a);
- C) scanf("%d",&pa);
- D) scanf("%d",*pa);

127. 有以下程序

```
main()
{   int  a=1, b=3, c=5;
    int  *p1=&a, *p2=&b, *p=&c;
    *p=*p1*( *p2);
    printf("%d\n",c);
}
```

执行后的输出结果是

(C)

- A) 1
- B) 2
- C) 3
- D) 4

128. 编制一个好的程序, 首先要保证它的正确性和可靠性, 还应强调良好的编程风格, 在选择标识符的名字时应考虑(C)。

- A) 名字长度越短越好, 以减少源程序的输入量
- B) 多个变量共用一个名字, 以减少变量名的数目
- C) 选择含义明确的名字, 以正确提示所代表的实体
- D) 尽量用关键字作名字, 以使名字标准化

129. 请读程序:

```
#include<stdio.h>
main()
{   int a;
    float b, c;
    scanf("%2d%3f%4f",&a,&b,&c);
    printf("\na=%d, b=%f, c=%f\n", a, b, c);
}
```

若运行时从键盘上输入 9876543210<CR>(<CR>表示回车), 则上面程序的输出结果是(C)

- A) a=98, b=765, c=4321
- B) a=10, b=432, c=8765
- C) a=98, b=765.000000, c=4321.000000
- D) a=98, b=765.0, c=4321.0

130. 请读程序:

```
#include<stdio.h>
f(char *s)
{ char *p=s;
while( *p!='\0') p++;
return(p-s);
}
main()
{ printf("%d\n",f("ABCDEF"));}
上面程序的输出结果是( B )
```

- A) 3
- B) 6
- C) 8
- D) 0

131. 有以下程序

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

程序的运行结果是(D)

- A) 331
- B) 41
- C) 2
- D) 1

132. 设变量 x 和 y 均已正确定义并赋值。以下 if 语句中，在编译时将产生错误信息的是 (C)

- A) if (x++);
- B) if (x>y && y!=0);
- C) if (x>0) x--// (分号)
- else y++;
- D) if (y<0) {;}
else x++;

133. 有以下程序

```
main()
{   int  p[7]={11,13,14,15,16,17,18},i=0,k=0;
    while(i<7&& p[i]%2){k=k+p[i];  i++;}
    printf("%d\n",k);
}
```

执行后输出的结果是(D)(注意循环停止的条件)

- A) 58
- B) 56
- C) 45
- D) 24

134. 对于下述 for 循环语句，说法正确的是(B)。

```
int i,k;
for(i=0,k=-1;k=1;i++,k++)
printf("***");
```

- A) 判断循环结束的条件非法

- B) 是无限循环
 C) 只循环一次
 D) 一次也不循环

135. C 语言源程序名的后缀是(B)

- A) .exe
 B) .C
 C) .obj
 D) .cp

136. 在 C 语言中, 如果下面的变量都是 int 类型, 则输出结果是(A)

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

- A) 7
 B) 6
 C) 5
 D) 4

137. 以下关于运算符优先顺序的描述中正确的是(C)

- A) 关系运算符<算术运算符<赋值运算符<逻辑与运算符
 B) 逻辑与运算符<关系运算符<算术运算符<赋值运算符
 C) 赋值运算符<逻辑与运算符<关系运算符<算术运算符
 D) 算术运算符<关系运算符<赋值运算符<逻辑与运算符

138. 下述语句中, (C)中的 if 语句语法是错误的。

- A) if(x>y);
 B) if(x==y) x+=y;
 C) if(x!=y) scanf("%d",&x)/(分号) else scanf("%d",&y);
 D) if(x<y) {x++;y++;}

139. 以下程序的输出结果是(A)

```
#include <stdio.h>
#include <string.h>
main()
{ char str[12]={'s','t','r','i','n','g'};
  printf("%d\n",strlen(str)); }
```

- A) 6
 B) 7
 C) 11
 D) 12

140. 若 x 为 int 型变量, y 是 float 型变量, 输入语句格式为:

```
scanf("x=%d,y=%f",&x,&y)
```

则为使 x=20,y=166.6, 正确的输入是(B)。

- A) 20 166.6 <回车>
 B) x=20,y=166.6 <回车>
 C) 20 <回车> 166.6 <回车>
 D) 20,166.6 <回车>

141. 有以下程序

```
#include <stdio.h>
#include <string.h>
main()
{ char str[ ][20]={"One*World", "One*Dream!"}, *p=str[1];
  printf("%d,", strlen(p)); printf("%s\n", p);
}
```

程序运行后的输出结果是(C)

- A) 9,One*World
- B) 9,One*Dream
- C) 10,One*Dream
- D) 10,One*World

142. 以下程序运行后,如果从键盘上输入:

book <回车>

book <空格><回车>

则输出的结果是(B)

```
#include<string.h>
```

```
main()
```

```
{char a1[80],a2[80],*s1=a1,*s2=a2;
```

```
gets(s1); gets(s2);
```

```
if(! strcmp(s1,s2)) printf("*");
```

```
else printf("#");
```

```
printf("%d\n",strlen(strcat(s1,s2)));
```

```
}
```

- A) *8
- B) #9
- C) #6
- D) *9

143. 若变量已正确定义, 要求程序段完成求 5! 的计算, 不能完成此操作的程序段是(B)

A) for(i=1, p=1; i<=5; i++) p*=i;

B) for(i=1; i<=5;i++){ p=1; p*=i; }

C) i=1; p=1; while(i<=5){ p*=i; i++;}

D) i=1;p=1;do{p*=i; i++; } while(i<=5)

144. 以下程序运行后,输出结果是(B)

```
main()
```

```
{
```

```
int y=18,i=0,j,a[8];
```

```
do
```

```
{ a[i]=y%2;
```

```
i++;
```

```
y=y/2;
```

```
} while(y>=1);
```

```
for(j=i-1;j>=0;j--) printf("%d",a[j]);
```

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

```
}
```

- A) 10000
- B) 10010
- C) 00110
- D) 10100

145. 在一个 C 语言程序中(B)

A) main 函数必须出现在所有函数之前

B) main 函数可以在任何地方出现

C) main 函数必须出现在所有函数之后

D) main 函数必须出现在固定位置

146. 有以下程序

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

```
main()
```

```
{ int c=0,k;
```

```
for (k=1;k<3;k++)
```

```

switch (k)
{default: c+=k;
 case 2: c++;break;
 case 4: c+=2;break;
 }
 printf(“%d\n”,c);
}

```

程序运行后的输出结果是(A)

- A) 3
- B) 5
- C) 7
- D) 9

147. 有以下程序

```

main()
{ char str[][10]={"China","Beijing"},*p=str;
 printf("%s\n",p+10);
}

```

程序运行后的输出结果是(B)

- A) China
- B) Beijing
- C) ng
- D) ing

148. 下列函数定义中，会出现编译错误的是(B)

A) `max(int x, int y, int* z)`

```
{ *z = x>y?x:y; }
```

B) `int max(int x, y)`

```
{
    int z;
    z = x>y?x:y;
    return z;
}
```

C) `max(int x, int y)`

```
{
    int z;
    z = x>y?x:y;
    return(z);
}
```

D) `int max(int x, int y)`

```
{ return (x>y ? x : y); }
```

149. 假定 a 和 b 为 int 型变量,则执行以下语句后 b 的值为(D)

```

a=1;b=10;
do
{b-=a;a++;}
while(b--<0);

```

- A) 9
- B) -2
- C) -1
- D) 8

150. 如果假设

a=2, b=3, x=3.5, y=2.5

那么下面的算术表达式的值是(C)。

`(float)(a+b)/2+(int)x%(int)y`

- A) 2
- B) 3
- C) 3.5
- D) 2.5

151. 以下程序段给数组所有的元素输入数据，请选择正确答案填入(**AD**)。

```
#include <stdio.h>
main()
{
    int a[10],i=0;
    while(i<10)
        scanf("%d",_____);
}
```

- A) a+(i++)
- B) &a[i+1]
- C) a+i
- D) &a[i++]

152. 以下选项中可作为 C 语言合法整数的是(**C**)

- A) 10110B
- B) 0386
- C) 0Xffa
- D) x2a2

153. 有以下程序

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

程序执行后的输出结果是

- (**C**)
- A) 45
- B) 20
- C) 25
- D) 36

三、多项选择题:

1. 以下叙述中正确的是(**BCD**)
 - A) 用户所定义的标识符允许使用关键字
 - B) 用户所定义的标识符应尽量做到“见名知意”
 - C) 用户所定义的标识符必须以字母或下划线开头
 - D) 用户定义的标识符中，大、小写字母代表不同标识
2. 下列定义变量的语句中正确的是(**ABC**)
 - A) int _int;
 - B) double int_;
 - C) char For;
 - D) float US\$;
3. 设有定义：int k=1, m=2; float f=7;，则以下选项中符合 C 语言语法的表达式是(**AB**)
 - A) k=k>=k
 - B) -k++
 - C) k%int(f)
 - D) k<>m

4. 若有定义语句：`double x[5]={1.0,2.0,3.0,4.0,5.0}, *p=x;`则正确引用 `x` 数组元素的是 (**ACD**)
 - A) `*p`
 - B) `x[5]`
 - C) `*(p+1)`
 - D) `*x`
5. 若要求定义具有 10 个 `int` 型元素的一维数组 `a`，则以下定义语句中正确的是 (**ABC**)
 - A) `#define N 10`
`int a[N];`
 - B) `#define n 5`
`int a [2*n];`
 - C) `int a[5+5];`
 - D) `int n=10,a[n];`
6. C 源程序中可以表示的数制是 (**BCD**)
 - A) 二进制
 - B) 八进制
 - C) 十进制
 - D) 十六进制
7. 按照 C 语言规定的用户标识符命名规则，能出现在标识符中的是(**ACD**)
 - A) 大写字母
 - B) 连接符
 - C) 数字字符
 - D) 下划线
8. 以下不合法的字符型常量是(**BD**)
 - A) `'\x13'`
 - B) `'\081'`
 - C) `'\065'`
 - D) `"\n"`
9. 以下合法的字符常量是 (**BC**)
 - A) `'\018'`
 - B) `'\"'`
 - C) `'\\'`
 - D) `'\0xcc'`
10. 以下叙述中正确的是(**ABD**)
 - A) C 语句必须以分号结束
 - B) 复合语句在语法上被看作一条语句
 - C) 空语句出现在任何位置都不会影响程序运行
 - D) 赋值表达式末尾加分号就构成赋值语句
11. 以下合法的赋值语句是(**AD**)
 - A) `n=(i=2,++i);`
 - B) `j++;`
 - C) `++(i+1);`
 - D) `x=j>0;`
12. 以下能正确定义二维数组的选项是 (**AB**)
 - A) `int a[2][2] = {{1}, {2}};`
 - B) `int a[][2] = {1, 2, 3, 4};`

- C) `int a[2][2] = {{1}, {2}, {3}};`
 D) `int a[2][] = {{1, 2}, {3, 4}};`
13. 若有定义: `int aa[8];` 则以下表达式中能代表数组元 `aa[1]` 的地址的是
 (ABD)
 A) `&aa[0]+1`
 B) `&aa[1]`
 C) `&aa[0]++`
 D) `aa+1`
14. 已有定义: `int i,a[10],*p;` , 则不合法的赋值语句是(AB)
 A) `p=100;`
 B) `p=a[5]`
 C) `p=&a[2]+2`
 D) `p=a+2;`
15. 以下能正确进行字符串赋初值的语句是(BCD)
 A) `char str[5]="good!";`
 B) `char str[]="good!";`
 C) `char *str="good!";`
 D) `char str[5]={'g','o','o','d'};`
16. 已知大写字母 A 的 ASCII 码是 65, 小写字母 a 的 ASCII 码是 97。以下能将变量 c 中的大写字母转换为对应小写字母的语句是(ABC)
 A) `c=(c-'A')%26+'a'`
 B) `c=c+32`
 C) `c=c-'A'+'a'`
 D) `c=('A'+c)%26-'a'`
17. 设有以下定义和语句
`char str[20]="Program", *p;`
`p=str;`
 则以下叙述中错误的是(CD)
 A) `*p` 与 `str[0]` 中的值相等
 B) 可以执行 `p++` 操作
 C) 可以执行 `str++` 操作
 D) `str` 数组长度和 `p` 所指向的字符串长度相等
18. 对于下面①, ②两个循环语句, 正确的描述是(AB)。
 ① `while(1);`
 ② `for(;;);`
 A) ①是无限循环
 B) ②是无限循环
 C) ①循环一次
 D) ②循环一次
19. 以下错误的函数定义形式是(BCD)
 A) `double fun (int x , int y)`
 B) `double fun (int x ; int y)`
 C) `double fun (int x , int y) ;`
 D) `double fun (int x , y) ;`
20. 对于基本类型相同的两个指针变量之间, 可以进行的运算是(ABD)
 A) `<`
 B) `=`
 C) `+`
 D) `-`
21. 能把字符串:Hello!赋给数组 b 的语句是(ACD)

- A) `char b[10]={'H','e','l','l','o','!'}`;
 B) `char b[10];b="Hello!"`;
 C) `char b[10];strcpy(b,"Hello!")`;
 D) `char b[10]="Hello!"`;
22. 以下能对二维数组 a 进行正确初始化的语句是(AD)
 A) `int a[2][3]={0}`;
 B) `int a[2][]={{1,2},{0}}`;
 C) `int a[2][3]={{1,2},{3,4},{5,6}}`;
 D) `int a[][3]={1,2,3,4,5,6}`;
23. `sizeof(double)`是(CD)
 A) 一种函数调用
 B) 一个双精度型表达式
 C) 一个整型表达式
 D) `sizeof` 是运算符
24. 设 a 为整型变量, 能正确表达数学关系: $10 < a < 15$ 的 C 语言表达式是(BC)
 A) `10 < a < 15`
 B) `a==11||a==12||a==13||a==14`
 C) `a>10&&a<15`
 D) `(a<10)||(a>15)`
25. C 语言中基本的数据类型包括(ABD)
 A) 整型
 B) 实型
 C) 逻辑型
 D) 字符型
26. 以下对一维整型数组 y 的错误定义是(ABC)。
 A) `int y(10)`;
 B) `int k=10,y[k]`;
 C) `int k`;
 D) `#define SIZE 8`
`int y[SIZE]`;
27. 若已定义: `int a[]={0,1,2,3,4,5,6,7,8,9}`, `*p=a,i`;其中 $0 \leq i \leq 9$, 则对 a 数组元素正确的引用是(ABC)
 A) `a[p-a]`
 B) `*(&a[i])`
 C) `p[i]`
 D) `a[10]`
28. 若 a、b、c、d 都是 int 类型变量且初值为 0, 以下选项中正确的赋值语句是(ACD)。
 A) `a=b=c=100`;
 B) `d++`;
 C) `c+=b`;
 D) `d=(c=22)-(b++)`;
29. 设有定义: `char p[]={'1','2','3'},*q=p`; , 以下能计算出一个 char 型数据所占字节数的表达式是(BCD)
 A) `sizeof(p)`
 B) `sizeof(char)`
 C) `sizeof(*q)`
 D) `sizeof(p[0])`
30. 下列选项中, 合法的 C 语言关键字是(BD)
 A) VAR
 B) char
 C) integer

D) default

31. 若变量 x、y 已正确定义并赋值，以下不正确的表达式是

(BCD)

A) ++x,y=x--

B) x+1=y

C) x=x+10=x+y

D) double(x)/10

32. 以下定义语句中错误的是

(ACD)

A) int a=b=0;

B) char A=65+1,b='b';

C) float a=1,*b=&a,*c=&b;

D) double a=0.0;b=1.1;

33. 已定义 ch 为字符型变量，以下赋值语句中正确的是(BCD)

A) ch='\';

B) ch=62+3;

C) ch=NULL;

D) ch='\xaa';

34. 下列选项中，正确的赋值表达式是(BC)

A) (a+b)++

B) n1=(n2=(n3=0))

C) k=i=j

D) a=b+c=1

35. 以下叙述中错误的是

(ABD)

A) C 程序中的注释只能出现在程序的开始位置和语句的后面

B) C 程序书写格式严格，要求一行内只能写一个语句

C) C 程序书写格式自由，一个语句可以写在多行上

D) 用 C 语言编写的程序只能放在一个程序文件中

36. 指出下面正确的赋值语句(CD)

A) a++

B) a==b;

C) a+=b;

D) a=1,b=1;

37. 下面正确的字符常量为(BD)。

A) "n"

B) '\\'

C) 'ab'

D) 'u'

38. 若程序中定义了以下函数：

```
double myadd(double a,double b)
```

```
{ return (a+b);}
```

并将其放在调用语句之后，则在调用之前应该对该函数进行说明，以下选项中正确的说明是

(BCD)

A) double myadd(double a,b);

B) double myadd(double,double);

C) double myadd(double b,double a);

D) double myadd(double x,double y);

(函数声明中也可以不写形参名只写该形参的声明类型，但不能只写形参名而不写该形参的类型)

39. 下选项中是 C 语句的是(ABD)。

- A) `{int i; i++; printf("%d\n",i);}`
 B) `;`
 C) `a=5,c=10`
 D) `{;}`
40. 以下叙述中错误的是(**AC**)
 A) 预处理命令行必须位于源文件的开头
 B) 在源文件的一行上只能有一条预处理命令
 C) 宏名必须用大写字母表示
 D) 宏替换不占有程序的运行时间
41. 设 `p1` 和 `p2` 是指向同一个 `int` 型一维数组的指针变量, `k` 为 `int` 型变量, 则能正确执行的语句是(**ACD**)
 A) `k=*p1+*p2;`
 B) `p2=k;`
 C) `p1=p2;`
 D) `k=*p1 *(*p2);`
42. 以下叙述中正确的是(**AB**)
 A) C 程序必须由一个或一个以上的函数组成
 B) 函数调用可以作为一个独立的语句存在
 C) 若函数有多个返回值, 必须通过多个 `return` 语句返回
 D) 函数形参的值改变也可以传回给对应的实参
43. 设有以下定义
`int a=0; int *p=&a; char c='A';`
`#define d 2`
 则下面语句中正确的是(**AC**)
 A) `a++;`
 B) `b++;`
 C) `c++;`
 D) `d++;`
44. 关于形参和实参的说法中, 错误的是(**ACD**)。
 A) 形参是虚设的, 所以它始终不占存储单元
 B) 实参与它所对应的形参占用不同的存储单元
 C) 实参与它所对应的形参占用同一个存储单元
 D) 实参与它所对应的形参同名时可占用同一个存储单元
45. 若有说明: `int n=2,*p=&n,*q=p;`, 则以下正确的赋值语句是(**ABC**)
 A) `p=q;`
 B) `*p=*q;`
 C) `n=*q;`
 D) `p=n;`
46. 下列错误的 C 语言标志符的是(**ACD**)。
 A) `Aa+`
 B) `_123`
 C) `a*bc`
 D) `b&cd`
47. 以下叙述中正确的是(**ABD**)
 A) 改变函数形参的值, 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`
48. 下面正确的转义字符是(**AC**)。

- A) '\101'
 B) '\'
 C) '\xaf'
 D) '\018'
49. 以下选项中，合法的字符常量是(AD)
 A) '\t'
 B) '\18'
 C) "n"
 D) '\xaa'
50. 下列叙述中错误的是(AC)
 A) break 语句只能用于 switch 语句
 B) 在 switch 语句中可以不使用 default
 C) break 语句必须与 switch 语句中的 case 配对使用
 D) 在 switch 语句中，不一定使用 break 语句
51. 下面不能正确进行字符串赋值操作的是(ABD)。
 A) char s[5]={"ABCDE"};
 B) char s[5]={'A','B','C','D','E'};
 C) char *s; s="ABCDE";
 D) char *s; scanf("%s", s);
52. t 为 int 类型，进入下面的循环之前，t 的值为 0
 while(t=1)
 { }
 则以下叙述中正确的是(BD)
 A) 循环控制表达式的值为 0
 B) 循环控制表达式的值为 1
 C) 循环控制表达式不合法
 D) 该循环为死循环
53. 下面正确的字符串常量是(BCD)
 A) 'abc'
 B) "1212"
 C) "aaa"
 D) ""
54. 若有说明：int a[3][4]; 则不能对 a 数组元素的正确引用是(ABD)。
 A) a[2][4]
 B) a[1, 3]
 C) a[1][0]
 D) a (2)(1)
55. 以下选项中合法的标识符是
 (ABD)
 A) print
 B) FOR
 C) &a
 D) _00
56. 以下数组定义中正确的是(ABC)
 A) int a[2][3];
 B) int b[][3]={0,1,2,3};
 C) int c[100][100]={0};
 D) int d[3][]={{1,2},{1,2,3},{1,2,3,4}};
57. 下列选项中，能用作标识符的是(AC)
 A) _1234_
 B) a-2

C) `int _2_`

D) `2_int_`

58. 以下为无限循环的语句或语句组是

(**BD**)

A) `n=0;`

`do {++n;} while (n<=0);`

B) `n=0;`

`while (1) {n++;}`

C) `n=10;`

`while (n);`

`{n--;}`

D) `for (n=0, i=1; ; i++) n+=i;`

59. 以下不能正确定义二维数组的是

(**CD**)

A) `int a[2][3];`

B) `int a[][3]={2*3};`

C) `int a[][3]={};`

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

60. 下列标识符中合法的是(**BC**)。

A) `student-name`

B) `_name`

C) `name8`

D) `3_DS`

61. 以下选项中，合法的常量是(**AD**)

A) `1.234e04`

B) `1.234e0.4`

C) `e+4`

D) `1.234e-2`

62. 设有如下程序段

`char s[20]="Beijing", *p;`

`p = s;`

则执行 `p=s;` 语句后，以下叙述错误的是(**BCD**)

A) 可以用 `*p` 表示 `s[0]`

B) `s` 数组中元素的个数和 `p` 所指字符串长度相等

C) `s` 和 `p` 都是指针变量

D) 数组 `s` 中的内容和指针变量 `p` 中的内容相同

63. 以下定义语句中错误的是(**ABD**)

A) `char a='A' b='B';`

B) `float a=b=10.0;`

C) `int a=10,*b=&a;`

D) `float *a,b=&a;`

64. 以下合法的用户标识符是

(**ABD**)

A) `j2_KEY`

B) `Double`

C) `4d`

D) `_8_`

65. 已定义 `c` 为字符型变量，则下列语句中错误的是

(**ABD**)

A) `c = '97';`

B) `c = "97";`

C) `c = 97;`

D) `c = "a";`

66. 若有以下程序

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

```
void f(int n);
```

```
main()
```

```
{ void f(int n);
```

```
  f(5);
```

```
}
```

```
void f(int n)
```

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

则以下叙述中正确的是(BCD)

A) 若只在主函数中对函数 f 进行说明, 则只能在主函数中正确调用函数 f

B) 若在主函数前对函数 f 进行说明, 则在主函数和其后的其它函数中都可以正确调用函数 f

C) 对于以上程序, 编译时系统会提示出错信息: 提示对 f 函数重复说明

D) 函数 f 无返回值, 所以可用 void 将其类型定义为无值型

67. 若 x 和 y 代表整型数, 以下表达式中能正确表示数学关系 $|x-y| < 10$ 的是(ABD)

A) `abs(x-y) < 10`

B) `x-y > -10 && x-y < 10`

C) `!(x-y) < -10 || !(y-x) > 10`

D) `(x-y)*(x-y) < 100`

68. 若已定义: `int a[9], *p=a;` 并在以后的语句中未改变 p 的值, 能表示 `a[1]` 地址的表达式是

(AB)

A) `p+1`

B) `a+1`

C) `a++`

D) `p[1]`

69. 下列函数定义中, 完全正确的是(ACD)

A) `max(int x, int y, int* z)`

```
{ *z = x > y ? x : y; }
```

B) `int max(int x, y)`

```
{
```

```
  int z;
```

```
  z = x > y ? x : y;
```

```
  return z;
```

```
}
```

C) `max(int x, int y)`

```
{
```

```
  int z;
```

```
  z = x > y ? x : y;
```

```
  return(z);
```

```
}
```

D) `int max(int x, int y)`

```
{ return (x > y ? x : y); }
```

70. 请选出可以作为 C 语言用户标识符的一组标识符号

(B)

A) `void define WORD`

B) `a3_b3 _123 IF`

C) `for -abc case`

D) `2a D0 size`

71. 对函数嵌套的叙述中, 错误的叙述为(ACD)。

A) 函数定义可以嵌套, 但函数调用不能嵌套

- B) 函数定义不可以嵌套，但函数调用可以嵌套
C) 函数定义和调用均不能嵌套
D) 函数定义和调用均可以嵌套
72. 设有语句：int a=1,b=2,*p1=&a,*p2=&b; 以下不能使指针 p1 指向变量 b 的赋值语句是(ABD)。
A) p1=*p2
B) *p1=p2
C) p1=p2
D) *p1=*p2
73. 以下不能正确定义字符串的语句是(AC)
A) char str[]={'\064'};
B) char str[]="\x43";
C) char str="";
D) char str[]="\0";
74. 判断字符串 s1 是否大于字符串 s2，不能达到目的的是(ABC)
A) if (s1>s2)
B) if (strcmp (s1, s2))
C) if (strcmp (s2, s1)>0)
D) if (strcmp (s1, s2)>0)
75. 若变量已正确定义并赋值，下面不符合 C 语言语法的表达式是(AD)
A) a:=b+1
B) a=b=c+2
C) (int)18.5%3
D) a=a+7=c+b
76. 下列语句中不符合 C 语言语法的赋值语句是(ACD)。
A) x=(3+b,z)=x+3;
B) x=7+y,y++,z++;
C) x=y+2=x+y+z;
D) x=3+y++=x+3;
77. 设 c1,c2 均是 char 类型变量，则以下正确的函数调用为(AD)。
A) printf("%c,%c",c1,c2);
B) getchar(c1); (该函数无参数)
C) putchar("\");
D) putchar(c1);
78. 以下所列的 C 语言常量中，正确的是(ACD)
A) 0xFF
B) 1.2e0.5
C) 2L
D) '\72'
79. 下列选项中错误的语句是(AC)
A) char s[8]; s={"Bei jing"};
B) char s[8]={"hello"};
C) char s[8]="Bei jing!";
D) char *s; s="Bei jing";
80. 若已定义的函数有返回值，则以下关于该函数调用的叙述中正确的是(ABC)
A) 函数调用可以作为独立的语句存在
B) 函数调用可以作为函数的实参
C) 函数调用可以出现在表达式中
D) 函数调用可以作为函数的形参
81. 已知 ch 是字符型变量，下面错误的赋值语句是(ACD)。

- A) `ch="\\";`
- B) `ch='\xff';`
- C) `ch='123';`
- D) `ch='\08';`

82. 下面各语句行中，不能正确进行**赋字符串**操作的语句行是(**BD**)

- A) `char st[]={"ABCDE"};`
- B) `char s[5]={'A','B','C','D','E'};`
- C) `char *s; s="ABCDE";`
- D) `char *s; scanf("%s",s);`

83. 以下正确的定义语句是(**ABD**)

- 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};`

84. 以下数组定义中正确的是(**ACD**)

- A) `int x[][3]={0};`
- B) `int x[2][3]={{1,2},{3,4},{5,6}};`
- C) `int x[][3]={{1,2,3},{4,5,6}};`
- D) `int x[2][3]={1,2,3,4,5,6};`

85. 以下叙述中正确的是(**ABD**)

- A) C 语言是一种结构化程序设计语言
- B) 结构化程序由顺序、分支、循环三种基本结构组成
- C) 使用三种基本结构构成的程序只能解决简单问题
- D) 结构化程序设计提倡模块化的设计方法