

C 语言程序设计

习题集

目 录

第一章	概 述	1
第三章	数据类型、运算符与表达式.....	2
第四章	最简单的C程序设计	8
第五章	逻辑运算和判断选取控制.....	12
第六章	循环控制	18
第七章	数 组	26
第八章	函 数	39
第九章	编译预处理	52
第十章	指 针	57
第十一章	结构体与共用体	81
第十二章	位运算	91
第十三章	文 件	93
全国计算机等级考试二级笔试试卷.....		100
全国计算机等级考试二级笔试试卷.....		117
全国计算机等级考试二级笔试试卷.....		132

第一章 概 述

1. C 语言程序的基本单位是：
A) 程序行 B) 语句 C) 函数 D) 字符
2. 以下说法中正确的是：
A) C 语言程序总是从第一个函数开始执行；
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；
3. 以下叙述正确的是：
A) 在 C 程序中，`main` 函数必须位于程序的最前面；
B) C 程序的每行中只能写一条语句；
C) C 语言本身没有输入输出语句；
D) 在对一个 C 程序进行编译的过程中，可发现注释中的拼写错误；
4. 以下叙述不正确的是：
A) 一个 C 源程序可由一个或多个函数组成；
B) 一个 C 源程序必须包含一个 `main` 函数；
C) C 程序的基本组成单位是函数；
D) 在 C 程序中，注释说明只能位于一条语句的后面；
5. C 语言规定：在一个源程序中，`main` 函数的位置：
A) 必须在最开始； B) 必须在系统调用的库函数的后面；
C) 可以任意； D) 必须在最后；
6. 一个 C 语言程序是由：
A) 一个主程序和若干子程序组成 B) 函数组成
C) 若干过程组成 D) 若干子程序组成
7. 在 C 语言中，输入操作是由库函数【 】完成的，输出操作是由库函数【 】完成的。
8. 一个 C 源程序中至少应包括一个【 】。
9. 以下叙述中正确的是
A) 构成 C 程序的基本单位是函数
B) 可以在一个函数中定义另一个函数
C) `main()` 函数必须放在其它函数之前
D) 所有被调用的函数一定要在调用之前进行定义

第三章 数据类型、运算符与表达式

1. C 语言中最简单的数据类型包括：
A) 整型、实型、逻辑型 B) 整型、实型、字符型
C) 整型、字符型、逻辑型 D) 整型、实型、逻辑型、字符型
2. 以下选项中属于 C 语言的数据类型是：
A) 复数型 B) 逻辑型 C) 双精度型 D) 集合型
3. 在 C 语言中，不正确的 int 类型的常数是：
A) 32768 B) 0 C) 037 D) 0xAF
4. 在 C 语言中，合法的字符常量是：
A) '\084' B) '\x43' C) 'ab' D) "\0"
5. 请选出作为 C 语言用户标识符的一组标识符：
A) void, define, WORD B) a3_b3, _123, IF
C) For, -abc, Case D) 2a, D0, sizeof
6. 在 C 语言提供的合法的关键字是
A) swith B) cher C) case D) defaulte
7. C 语言中的标识符只能是由字母、数字和下划线三种字符组成，且第一个字符
A) 必须是字母 B) 必须为下划线
C) 必须为字母或下划线 D) 可以是字母、数字或下划线中任一种字符
8. 以下正确的叙述是
A) 在 C 程序中，每行只能写一条语句
B) 若 a 是实型变量，C 程序中允许赋值 a=10，因此实型变量中允许存放整型数
C) 在 C 程序中，无论是整数还是实数，都能被准确无误地表示
D) 在 C 程序中，%是只能用于整数运算的运算符
9. 以下程序的输出结果是 【 】。

```
main ()
{ unsigned short a=65536 ;
  int b ;
  printf ("%d\n", b=a); }
```
10. 数字符号 0 的 ASCII 码十进制表示为 48，数字符号 9 的 ASCII 码十进制表示为
【 】。
11. 若已定义 x 和 y 为 double 类型，则表达式：x=1，y=x+3/2 的值是
A) 1 B) 2 C) 2.0 D) 2.5
12. 字符 (char) 型数据在微机内存中的存储形式是
A) 反码 B) 补码 C) EBCDIC 码 D) ASCII 码

13. 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
14. 已知字母 A 的 ASCII 码为十进制的 65, 下面程序的输出是
- A) 67, D B) B, C
C) C, D D) 不确定的值
- ```
main ()
{ char ch1, ch2;
 ch1='A'+5-'3'; ch2='A'+6-'3';
 printf ("%d, %c\n", ch1, ch2); }
```
15. 下列不正确的转义字符是
- A) '\\ '      B) '\ '      C) '074'      D) '\0'
16. 语句 `printf ("a\bre\hi\y\\bou\n");` 的输出结果是
- A) `a\bre\hi\y\\bou`      B) `a\bre\hi\y\bou`  
C) `re'hi'you`      D) `abre'hi'y\bou`  
(说明: '\b' 是退格符)
17. 下列变量定义中合法的是
- A) `short _a=1-.1e-1;`      B) `double b=1+5e2.5;`  
C) `long do=0xfdaL;`      D) `float 2_and=1-e-3;`
18. 在 C 语言中 (以 16 位 PC 机为例), 5 种基本数据类型的存储空间长度的排列顺序为
- A) `char < int < long int <= float < double`  
B) `char = int < long int <= float < double`  
C) `char < int < long int = float = double`  
D) `char = int = long int <= float < double`
19. 设有以下变量定义, 并已赋确定的值
- ```
char w; int x; float y; double z;
```
- 则表达式: `w*x+z-y` 所求得的数据类型为【 】。
20. 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
- A) float      B) int      C) char      D) double
21. 下面程序的输出是
- A) 0      B) 1      C) 3      D) 不确定的值
- ```
main ()
{ int x=10, y=3;
  printf ("%d\n", y=x/y); }
```
22. 设有如下定义: `int x=10, y=3, z;` 则语句

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

的输出结果是

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

23. C 语言中，运算对象必须是整型数的运算符是

- A) % B) \ C) %和\ D) * *

24. 以下程序的输出结果是

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

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

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

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

26. 若有以下程序段：

```
int c1=1, c2=2, c3;
c3=1.0/c2*c1;
```

则执行后，c3 中的值是

- A) 0 B) 0.5 C) 1 D) 2

27. 在 C 语言中，要求运算数必须是整型的运算符是

- A) % B) / C) < D) !

28. C 语言中运算对象必须是整型的运算符是

- A) %= B) / C) = D) <=

29. 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);

30. 若变量已正确定义并赋值，下面符合 C 语言语法的表达式是

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

31. 若变量 a, i 已正确定义，且 i 已正确赋值，合法的语句是

- A) a= =1 B) ++i; C) a=a++=5; D) a=int (i);

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

- A) c=b=a; B) (a=c) || (b=c);
C) (a=c) && (b=c); D) a=c=b;

33. 下列语句中符合 C 语言语法的赋值语句是

- A) a=7+b+c=a+7; B) a=7+b++=a+7;
C) a=7+b, b++, a+7 D) a=7+b, c=a+7;

34. 设 `int a=12;` 则执行完语句 `a+=a-=a*a` 后，a 的值是

- A) 552 B) 264 C) 144 D) -264

35. 假设在程序中 a , b , c 均被定义成整型 , 并且已赋大于 1 的值 , 则下列能正确表示代数式 $1/abc$ 的表达式是

- A) $1/a*b*c$ B) $1/(a*b*c)$
C) $1/a/b/(float)c$ D) $1.0/a/b/c$

36. 设有如下的变量定义:

```
int i=8 , k , a , b ;  
unsigned long w=5 ;  
double x=1.42 , y=5.2 ;
```

则以下符合 C 语言语法的表达式是

- A) $a+=a-(b=4)*(a=3)$ B) $x\%(-3);$
C) $a=a*3=2$ D) $y=float(i)$

37. 设 a、b、c 为整型数 , 且 $a=2$, $b=3$, $c=4$, 则执行完以下语句:

$a*=16+(b++)-(++c);$ 后, a 的值是【 】。

38. 若有定义: $int a=10, b=9, c=8;$ 执行下列语句后, 变量 b 中的值是【 】。

$c=(a=(b-5)); c=(a\%11)+(b=3);$

39. 若 X 和 Y 都是 int 型变量, $X=100$, $Y=200$, 且有下面的程序片段:

```
printf ("%d", (X , Y)) ;
```

上面程序片段的输出结果是

- A) 200 B) 100
C) 100 200 D) 输出格式符不够, 输出不确定的值

41. 执行下面程序中输出语句后, a 的值是

- A) 65 B) 20 C) 15 D) 10

```
main ()  
{ int a ;  
printf ("%d\n", (a=3*5 , a*4 , a+5)); }
```

42. 设 x、y、z 和 k 都是 int 型变量, 则执行表达式: $x=(y=4, z=16, k=32)$ 后, x 的值为

- A) 4 B) 16 C) 32 D) 52

43. 以下叙述不正确的是

- A) 在 C 程序中, 逗号运算符的优先级最低
B) 在 C 程序中, APH 和 aph 是两个不同的变量
C) 若 a 和 b 类型相同, 在计算了赋值表达式 $a=b$ 后 b 中的值将放入 a 中, 而 b 中的值不变
D) 当从键盘输入数据时, 对于整型变量只能输入整型数值, 对于实型变量只能输入实型数值

44. 以下选项中, 非法的字符常量是

- A) `'\t'` B) `'\17'` C) `"n"` D) `'\xaa'`

45. 若有定义: `int a=8, b=5, c;`, 执行语句 `C=a/b+0.4;` 后, `c` 的值为
A) 1.4 B) 1 C) 2.0 D) 2
46. 若变量 `a` 是 `int` 类型, 并执行了语句: `a= 'A'+1.6;`, 则正确的叙述是
A) `a` 的值是字符 `C`
B) `a` 的值是浮点型
C) 不允许字符型和浮点型相加
D) `a` 的值是字符 '`A`' 的 ASCII 值加上 1。
47. 以下选项中不属于 C 语言的类型的是
A) `signed short int` B) `unsigned long int`
C) `unsigned int` D) `long short`
48. 在 16 位 C 编译系统上, 若定义 `long a;`, 则能给 `a` 赋 40000 的正确语句是
A) `a=20000+20000;` B) `a=4000*10;`
C) `a=30000+10000;` D) `a=4000L*10L;`
49. 以下程序的输出结果是【 】。
- ```
main()
{
 int a=0;
 a+=(a=8);
 printf("%d\n", a);
}
```
50. 以下选项中合法的实型常数是  
A) `5E2.0`                      B) `E-3`                      C) `.2E0`                      D) `1.3E`
51. 以下选项中合法的用户标识符是  
A) `long`                      B) `_2Test`                      C) `3Dmax`                      D) `A. dat`
52. 已知大写字母 `A` 的 ASCII 码值是 65, 小写字母 `a` 的 ASCII 码是 97, 则用八进制表示的字符常量 '`\101`' 是  
A) 字符 `A`                      B) 字符 `a`                      C) 字符 `e`                      D) 非法的常量
53. 设 `a` 和 `b` 均为 `double` 型变量, 且 `a=5.5`、`b=2.5`, 则表达式 `(int) a+b/b` 的值是  
A) `6.500000`                      B) `6`                      C) `5.500000`                      D) `6.000000`
54. 若有以下程序:
- ```
main()
{
    int k=2, i=2, m;
    m=(k+=i*=k); printf( "%d,%d\n", m, i);
}
```
- 执行后的输出结果是

A) 8, 6

B) 8, 3

C) 6, 4

D) 7, 4

55. 与数学式子 $\frac{3x^n}{2x-1}$ 对应的 C 语言表达式是

A) $3*x^n(2*x-1)$

B) $3*x**n(2*x-1)$

C) $3*pow(x, n)*(1/(2*x-1))$

D) $3*pow(n, x)/(2*x-1)$

56. 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) $k+=n+1$

第四章 最简单的 C 程序设计

1. putchar 函数可以向终端输出一个
 - A) 整型变量表达式
 - B) 实型变量值
 - C) 字符串
 - D) 字符或字符型变量值
2. printf 函数中用到格式符 %5s, 其中数字 5 表示输出的字符占用 5 列。如果字符串长度大于 5, 则输出按方式【 】; 如果字符串长度小于 5, 则输出按方式【 】。
 - A) 从左起输出该字符串, 右补空格
 - B) 按原字符串长从左向右全部输出
 - C) 右对齐输出该字符串, 左补空格
 - D) 输出错误信息
3. 以下说法正确的是
 - A) 输入项可以为一个实型常量, 如 scanf ("%f ", 3.5);
 - B) 只有格式控制, 没有输入项, 也能进行正确输入, 如 scanf ("a=%d, b=%d");
 - C) 当输入一个实型数据时, 格式控制部分应规定小数点后的位数, 如 scanf ("% 4.2f ", &f);
 - D) 当输入数据时, 必须指明变量的地址, 如 scanf ("%f ", &f);
4. 若有以下定义和语句:

```
int u=010 , v=0x10 , w=10 ;
printf ("%d , %d , %d\n" , u , v , w) ;
```

则输出结果是:
 - A) 8 , 16 , 10
 - B) 10 , 10 , 10
 - C) 8 , 8 , 10
 - D) 8 , 10 , 10
5. 以下程序的输出结果是
 - A) 17, 021, 0x11
 - B) 17, 17, 17
 - B) 17, 0x11, 021
 - D) 17, 21, 11

```
main ( )
{ int k=17;
  printf ("%d, %o, %x \n", k, k, k); }
```
6. 下面程序的输出是

```
main ( )
{ unsigned int a=32768 ;
  printf ("a=%d\n" , a) ; }
```

 - A) a=-32768
 - B) a=32767
 - C) a=-32767
 - D) a=-1

7. 下列程序执行后的输出结果是

- A) -32767 B) FFFE C) -1 D) -32768

```
main ()
{ char x=0xFFFF ; printf ("%d \n", x--) ; }
```

8. 有如下程序

```
main ()
{ int y=3 , x=3 , z=1 ;
  printf (" %d %d\n", (++x , y++) , z+2) ; }
```

运行该程序的输出结果是

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

9. 下列程序的运行结果是

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

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

10. 下列程序的输出结果是

- A) 3 B) 3.2 C) 0 D) 3.07

```
main ()
{ double d=3.2 ; int x , y ;
  x=1.2 ; y= (x+3.8) /5.0;
  printf ("%d \n", d*y); }
```

11. 下列程序执行后的输出结果是

- A) G B) H C) I D) J

```
main ()
{ int x='f' ; printf ("%c \n", 'A'+ (x-'a'+1)); }
```

12. 语句: printf ("%d", (a=2) && (b= -2)); 的输出结果是

- A) 无输出 B) 结果不确定 C) -1 D) 1

13. 下列程序执行后的输出结果是 (小数点后只写一位)

- A) 6 6 6.0 6.0 B) 6 6 6.7 6.7
C) 6 6 6.0 6.7 D) 6 6 6.7 6.0

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

14. 若有以下定义和语句

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

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

则输出结果是:

A) 2 , M B) 3 , E

C) 2 , E D) 输出项与对应的格式控制不一致 , 输出结果不确定

15. 若想通过以下输入语句给 a 赋于 1, 给 b 赋于 2, 则输入数据的形式应该是【 】。

```
int a, b;
```

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

16. 若有定义: int x , y ; char a , b , c ; 并有以下输入数据 (此处<CR> 代表换行符 , u 代表空格):

1u2<CR> AuBuC<CR>

则能给 x 赋整数 1 , 给 y 赋整数 2 , 给 a 赋字符 A , 给 b 赋字符 B , 给 c 赋字符 C 的正确程序段是

A) scanf ("x=%d y+%d", &x, &y); a=getchar ();

b=getchar (); c=getchar ();

B) scanf ("%d %d", &x , &y); a=getchar ();

b=getchar (); c=getchar () ;

C) scanf ("%d%d%c%c%c", &x , &y , &a , &b , &c) ;

D) scanf ("%d%d%c%c%c%c%c%c", &x, &y, &a, &a, &b, &b, &c, &c);

17. 请读程序:

```
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> (表示回车), 则输出结果是

A) A=98, B=765, C=4321

B) A=98, B=765.000000, C=4321.000000

C) A=98, B=765.0, C=4321.0

D) A=98.000000, B=765.000000, C=4321.000000

18. 若想通过以下输入语句使 a=5.0 , b=4 , c=3 , 则输入数据的形式是【 】。

...

```
int b , c ; float a ;
```

```
scanf ("a=%f, b=%d, c=%d", &a, &b, &c);
```

...

19. 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

20. 以下程序的输出结果是【 】。

```
main()
{   int  a=177;
    printf("%o\n", a);
}
```

21. 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

- 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);`

22. 设有定义：`long x=-123456L;`，则以下能够正确输出变量 x 值的语句是

- A) `printf("x=%d\n" , x);` B) `printf("x=%ld\n" , x);`
C) `printf("x=%8dL\n" , x);` D) `printf("x=%LD\n" , x);`

23. 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

第五章 逻辑运算和判断选取控制

1. 能正确表示逻辑关系：“ $a \geq 10$ 或 $a \leq 0$ ”的 C 语言表达式是
A) $a \geq 10$ or $a \leq 0$ B) $a \geq 0 | a \leq 10$
C) $a \geq 10 \ \&\& a \leq 0$ D) $a \geq 10 \ || a \leq 0$
2. 为表示关系 $x \geq y \geq z$ ，应使用 C 语言表达式
A) $(x \geq y) \ \&\& (y \geq z)$ B) $(x \geq y) \ \text{AND} (y \geq z)$
C) $(x \geq y \geq z)$ D) $(x \geq y) \ \& (y \geq z)$
3. 表示条件： $10 < X$ 或 $X < 0$ 的 C 语言表达式是【 】。
4. 条件“ $x > 20$ 或 $x < -100$ ”的 C 语言表达式是【 】。
5. 设 $a=5$ ， $b=6$ ， $c=7$ ， $d=8$ ， $m=2$ ， $n=2$ ，执行 $(m=a>b) \ \&\& (n=c>d)$ 后 n 的值为
A) 0 B) 1 C) 2 D) 7
6. 下面程序的输出是
A) 003 B) 012 C) 103 D) 112

```
main ()
{   int    a= -1 , b=4 , k ;
    k= (a++<=0) && (! (b--<=0));
    printf ("%d%d%d\n", k , a , b); }
```
7. 设 $\text{int } A=3$ ， $B=4$ ， $C=5$ ；，则下列表达式中，值为 0 的表达式是
A) $A \ \&\& B$ B) $A \leq B$
C) $A | | B + C \ \&\& B$ D) $! ((A < B) \ \&\& ! C | | 1)$
8. 有以下程序

```
main ()
{   int    a , b , c=246 ;
    a=c/100%9 ; b= (-1) && (-1) ;
    printf ("%d ; %d\n" , a , b) ; }
```

输出结果是

A) 2 ; 1 B) 3 ; 2 C) 4 ; 3 D) 2 ; -1
9. 设 $\text{int } x=1$ ， $y=1$ ；表达式 $(!x || y--)$ 的值是
A) 0 B) 1 C) 2 D) -1
10. 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) ture B) false C) 0 D) 1

11. 表示“整数 x 的绝对值大于 5”时值为“真”的 C 语言表达式是 【 】。

12. 逻辑运算符两侧运算对象的数据类型

- A) 只能是 0 或者 1
B) 只能是 0 或非 0 正数
C) 只能是整型或字符型数据
D) 可以是任何类型的数据

13. 以下关于运算符优先顺序的描述中正确的是

- A) 关系运算符 < 算术运算符 < 赋值运算符 < 逻辑运算符
B) 逻辑运算符 < 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 < 逻辑运算符

14. 若 int K=3;，且有下列的程序片段：

```
if (K<=0) printf ("####");  
else printf ("&&&&");
```

上面程序片段的输出结果是

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

15. 设 char CH ;，其值为 A，且有下列的表达式：

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

则表达式的值是

- A) A B) a C) Z D) z

16. 有如下程序

```
main ()  
{ int a=2 , b=-1 , c=2 ;  
  if (a<b)  
    if (b<0) c=0;  
  else c++ ;  
  printf ("%d\n", c); }
```

该程序的输出结果是

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

17. 两次运行下面的程序，如果从键盘上分别输入 6 和 4，则输出结果是

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

A) 7 和 5 B) 6 和 3 C) 7 和 4 D) 6 和 4

18. 假定所有变量均已正确说明，下列程序段运行后 x 的值是

```
a=b=c=0 ; x=35 ;
if (!a) x-- ;
else if (b) ;
    if (c) x=3 ;
    else x=4 ;
```

A) 34 B) 4 C) 35 D) 3

19. 与 $y = (x > 0 ? 1 : x < 0 ? -1 : 0)$ 的功能相同的 if 语句是

A) if (x>0) y=1;	B) if (x)
else if (x<0) y=-1;	if (x>0) y=1;
else y=0;	else if (x<0) y=-1;
	else y=0;
C) y=-1 ;	D) y=0;
if (x)	if (x>=0)
if (x>0) y=1;	if (x>0) y=1;
else if (x==0) y=0;	else y=-1;
else y=-1;	

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

A) $a < > 0$ B) !a C) a=0 D) a

22. 以下程序的输出结果是

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

```
main ( )
{ int a= -1 , b=1 ;
  if (++a) printf ("%d %d\n", a, b);
  else printf ("%d %d\n", b, a); }
```

23. 有如下程序

```
main ( )
{ float x=2.0, y;
  if (x<0.0) y=0.0;
  else if (x<10.0) y=1.0/x;
  else y=1.0;
  printf ("%f\n", y); }
```

该程序的输出结果是

A) 0.000000 B) 0.250000 C) 0.500000 D) 1.000000

24. 请读程序:

```
main ( )
```



```

{ float x , y ;
  scanf ("%f ", &x);
  if (x<0.0) y=0.0 ;
  else if ((x<5.0) && (x!=2.0)) y=1.0/ (x+2.0);
      else if (x<10.0) y=1.0/x;
          else y=10.0;
  printf ("%f\n", y); }

```

若运行时从键盘上输入 2.0 (表示回车), 则上面程序的输出结果是

A) 0.000000 B) 0.250000 C) 0.500000 D) 1.000000

25. 若有以下定义:

```

float x ;
int a , b ;

```

则正确的 switch 语句是

- A) switch (x)
 - { case 1.0: printf ("*\n");
 - case 2.0: printf ("**\n"); }
- B) switch (x)
 - { case 1, 2: printf ("*\n");
 - case 3: printf ("**\n"); }
- C) switch (a+b)
 - { case 1: printf ("\n") ;
 - case 1+2: printf ("**\n"); }
- D) switch (a+b);
 - { case 1: printf ("*\n");
 - case 2: printf ("**\n"); }

26. 有如下程序

```

main ()
{ int x=1 , a=0 , b=0 ;
  switch (x)
  { case 0: b++;
    case 1: a++ ;
    case 2: a++; b++; }
  printf (" a=%d, b=%d\n", a, b); }

```

该程序的输出结果是

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

27. 假定 w、x、y、z、m 均为 int 型变量, 有如下程序段:

```

w=1; x=2; y=3; z=4;
m=(w<x)?w; x; m=(m<y)?m;y; m=(m<z)?m; z;

```

则该程序运行后，m 的值是

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

28. 整型变量 x 和 y 的值相等、且为非 0 值，则以下选项中，结果为零的表达式是

- A) $x \parallel y$ B) $x \mid y$ C) $x \& y$ D) $x \wedge y$

29. 以下程序的输出结果是

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

- A) 5 B) 4 C) 6 D) 不确定

30. 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

31. 若从键盘输入 58, 则以下程序输出的结果是 **【 】**。

```
main()
{   int  a;
    scanf("%d",&a);
    if(a>50) printf("%d",a);
    if(a>40) printf("%d",a);
    if(a>30) printf("%d",a);
}
```

32. 有以下程序

```
main()
{   int a=15,b=21,m=0;
    switch(a%3)
    {   case 0:m++;break;
        case 1:m++;
        switch(b%2)
        {   default:m++;
            case 0:m++;break;
        }
    }
}
```

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

程序运行后的输出结果是

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

33. 以下程序运行后的输出结果是_____。

```
main()  
{ int x=10, y=20, t=0;  
  if(x==y) t=x; x=y; y=t;  
  printf( "%d,%d \n" , x, y);  
}
```

第六章 循环控制

1. 若输入字符串: abcde<回车>, 则以下 while 循环体将执行【 】次。

```
While ((ch=getchar ()) = 'e') printf ("*");
```

2. 有如下程序

```
main ()
{ int n=9;
  while (n>5) { n-- ; printf ("%d", n); }
```

该程序段的输出结果是

- A) 987 B) 876 C) 8765 D) 9876
3. 运行以下程序后, 如果从键盘上输入 china#<回车>, 则输出结果为

A) 2, 0 B) 5, 0 C) 5, 5 D) 2, 5

```
# include<stdio.h>
main ( )
{ int v1=0, v2=0;
  char ch;
  while ((ch=getchar ( )) != ' # ' )
  switch (ch)
  { case 'a':
    case 'h':
    default: v1+ +;
    case '0': v2+ +; }
  printf ("%d, %d\n", v1, v2); }
```

4. 请读程序:

```
main ()
{ int num=0 ;
  while (num<=2) { num++ ; printf ("%d\n", num); }}
```

上面程序的输出结果是

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

5. 请读程序:

```
#include "math.h"
main ()
{ float x , y , z ;
```

```
scanf ("%f %f ", &x, &y) ;
z=x/y;
while (1)
{ if (fabs (z) >1.0) {x=y; y=z; z=x/y; }
  else break; }
printf ("%f\n", y); }
```

若运行时从键盘上输入 3.6 2.4<CR>, 则输出结果是

- A) 1.500000 B) 1.600000 C) 2.000000 D) 2.400000

6. 以下程序的功能是：从键盘上输入若干个学生的成绩，统计并输出最高成绩和最低成绩，当输入负数时结束输入。请填写。

```
main ()
{ float x , amax , amin ;
  scanf ("%f ", &x) ;
  amax=x ; amin=x ;
  while (【          】)
  { if (x>amax) amax=x;
    if (【          】) amin=x;
    scanf ("%f ", &x);      }
  printf ("\namax=%f\namin=%f\n", amax, amin); }
```

7. 定义如下变量：

```
int n=10;
```

则下列循环的输出结果是

```
while (n>7)
{ n--; printf ("%d\n", n); }
```

A) 10	B) 9	C) 10	D) 9
9	8	9	8
8	7	8	7
		7	6

8. 以下程序段的输出结果是

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

A) 1 B) 3 0 C) 1 -2 D) 死循环

9. 以下叙述正确的是

- A) do-while 语句构成的循环不能用其它语句构成的循环来代替。
 B) do-while 语句构成的循环只能用 break 语句退出。
 C) do-while 语句构成的循环，在 while 后的表达式为非零时结束循环。
 D) 用 do-while 语句构成的循环，在 while 后的表达式为零时结束循环。

10. 以下程序的输出结果是

- A) 20 B) 31 C) 321 D) 210

```
main ( )
{   int  n=4;
    while (n-->0) printf ("%d ", --n); }
```

11. 设有如下程序段:

```
int  i=0 ,  sum=1 ;
do { sum+ = i++ ; } while (i<6) ;
printf ("%d\n" ,  sum) ;
```

上述程序段的输出结果是【 】。

12. 执行下面程序片段的的结果是

```
int  x=23 ;
do { printf ("%2d" , x-- ) ; } while (!x) ;
```

- A) 打印出 321 B) 打印出 23
C) 不打印任何内容 D) 陷入死循环

13. 假定 a 和 b 为 int 型变量 ; 则执行以下语句后 b 的值为

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

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

14. 以下循环体的执行次数是【 】。

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

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

15. 以下程序的输出结果是

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

```
main ( )
{   int  i;
    for (i='A'; i < 'I' ; i++ , i++)
        printf ("%c", i+32) ;
    printf (" \n") ; }
```

16. 以下程序的输出结果是

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

```
main ( )
{   int  x=10 , y=10 , i ;
    for (i=0; x>8; y=++i)
```

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

17. 请读程序:

```
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

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

18. 有如下程序

```
main ( )
{ int i , sum ;
  for (i=1; i<=3; sum++) sum+= i;
  printf (" %d\n", sum); }
```

该程序的执行结果是

A) 6 B) 3 C) 不确定 D) 0

19. 若 X 是 int 型变量, 且有下面的程序片段:

```
for (X=3; X<6; X++)
printf ((X%2) ? ("**%d") : ("##%d\n"), X);
```

上面程序片段的输出结果是

A) **3 B) ##3 C) ##3 D) **3##4
 ##4 **4 **4##5 **5
 **5 ##5

20. 设 j 为 int 型变量 ; 则下面 for 循环语句的执行结果是

```
for ( j=10; j>3; j--)
{ if (j%3) j--;
  --j; --j;
  printf ("%d " ; j); }
```

A) 63 B) 74 C) 62 D) 73

21. 下面程序的功能是: 计算 1 到 10 之间奇数之和及偶数之和, 请填空。

```
#include <stdio.h>
main ( )
{ int a , b , c , i ;
  a=c=0;
  for (i=0; i<10; i+=2)
```

```

    { a+=i; 【          】 ; c+=b; }
    printf ("偶数之和=%d\n", a);
    printf ("奇数之和=%d\n", c-11); }

```

22. 下面程序的功能是：输出 100 以内能被 3 整除且个位数为 6 的所有整数，请填空。

```

#include <stdio.h>
main ()
{   int   i ,   j ;
    for (i=0; 【          】 ; i++)
    {   j=i*10+6;
        if ( 【          】 ) continue;
        printf ("%d", j); }
}

```

23. 执行下面的程序后，a 的值为

```

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; }
}

```

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

24. 下面程序的输出是

A) 741 B) 852 C) 963 D) 875421

```

main ()
{   int   y=9;
    for ( ; y>0; y--)
    {   if (y%3==0) { printf ("%d", --y); continue; }
    }
}

```

25. 设 x 和 y 均为 int 型变量，则执行下面的循环后，y 值为

```

for (y=1 , x=1; y<=50; y++)
{   if (x>=10) }break;
    if (x%2==1) { x+=5; continue; }
    x-=3;
}

```

A) 2 B) 4 C) 6 D) 8

26. 下面有关 for 循环的正确描述是

A) for 循环只能用于循环次数已经确定的情况

- B) for 循环是先执行循环体语句，后判断表达式
C) 在 for 循环中，不能用 break 语句跳出循环体
D) for 循环的循环体语句中，可包含多条语句，但必须用花括号括起来
27. 以下正确的描述是

- A) continue 语句的作用是结束整个循环的执行
B) 只能在循环体内和 switch 语句体内使用 break 语句
C) 在循环体内使用 break 语句或 continue 语句的作用相同
D) 从多层循环嵌套中退出时，只能使用 goto 语句

28. 有以下程序

```
main ()
{ int i, j;
  for (j=10; j<11; j++)
    for (i=9; i<=j-1; i++) printf ("%d" ; j);
}
```

输出结果是

- A) 11 B) 10 C) 9 D) 10 11

29. 下列程序的运算结果是

- A) 4 B) 2 C) 1 D) 0

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

30. 以下描述正确的是

- A) goto 语句只能用于退出多层循环
B) switch 语句中不能出现 continue 语句
C) 只能用 continue 语句来终止本次循环
D) 在循环中 break 语句不能独立出现

31. 以下程序的输出结果是

- A) ##### B) ##### C) ***** D) #####

```
main ()
{ int i;
  for (i=1; i <6 ; i++)
  { if (i%2) { printf ("#"); continue; }
    printf ("*"); }
  printf ("\n");
}
```

32. 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

33. 以下程序的输出结果是【 】。

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

34. 以下程序的功能是：按顺序读入 10 名学生 4 门课程的成绩，计算出每位学生的平均分并输出，程序如下：

```
main()
{ int n,k;
  float score ,sum,ave;
  sum=0.0;
  for(n=1;n<=10;n++)
  { for(k=1;k<=4;k++)
    { scanf( "%f" ,&score); sum+=score;}
    ave=sum/4.0;
    printf( "NO%d:%f\n" ,n,ave);
  }
}
```

上述程序运行后结果不正确，调试中发现有一条语句出现在程序中的位置不正确。这条语句是

A) sum=0.0; B) sum+=score;
C) ave=sum/4.0; D) printf("NO%d:%f\n" ,n,ave);

35. 有以下程序段

```
int n=0,p;  
do{scanf("%d",&p);n++;}while(p!=12345 && n<3);
```

此处 do—while 循环的结束条件是

- A) P 的值不等于 12345 并且 n 的值小于 3
- B) P 的值等于 12345 并且 n 的值大于等于 3
- C) P 的值不等于 12345 或者 n 的值小于 3
- D) P 的值等于 12345 或者 n 的值大于等于 3

36. 有以下程序:

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

程序运行时, 如果从键盘输入: Y? N? <回车>, 则输出结果为 [10] 。

37. 以下函数的功能是计算 $s=1+\frac{1}{2!}+\frac{1}{3!}+\cdots+\frac{1}{n!}$, 请填空。

```
double fun(int n)  
{ double s=0.0, fac=1.0; int i;  
for(i=1, i<=n; i++)  
{ fac=fac [12] ;  
s=s+fac;  
}  
return s;  
}
```


8. 有如下程序

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

该程序的输出结果是

- A) 不确定的值 B) 2 C) 1 D) 0

9. 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

10. 阅读下列程序:

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

下述程序运行后输出结果是

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

11. 以下程序的功能是: 从键盘上输入若干个学生的成绩, 统计计算出平均成绩, 并输出低于平均分的学生成绩, 用输入负数结束输入。请填空。

```
main ()
{ float x[1000], sum=0.0, ave, a;
  int n=0, i;
  printf ("Enter mark: \n"); scanf ("%f", &a);
  while (a=0.0&& n)
  { sum+ 【                      】 ;
    x[n]= 【                      】;
    n++; scanf ("%f", &a); }
  ave= 【                      】;
```

```
printf ("Output: \n");
printf ("ave=%f\n", ave);
for ( i=0; i<n; i++)
    if (【          】) printf ("%d  ", x[i]); }
```

12. 对以下说明语句的正确理解是

```
int a[10]={ 6, 7, 8, 9, 10} ;
```

- A) 将 5 个初值依次赋给 a[1] 至 a[5]
- B) 将 5 个初值依次赋给 a[0] 至 a[4]
- C) 将 5 个初值依次赋给 a[6] 至 a[10]
- D) 因为数组长度与初值的个数不相同, 所以此语句不正确

13. 有如下程序

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

该程序的输出结果是

- A) 18 B) 19 C) 20 D) 21

14. 有定义如下:

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

则下面语句的输出结果是

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

- A) 159 B) 147 C) 357 D) 369

15. 阅读下列程序:

```
#include<stdio.h>
main ( )
{ int i , j , row , col , m ;
  static int a[3][3]={ {100, 200, 300} , {28, 72, -30} ,
                      {-850, 2, 6}};

  m=a[0][0] ;
  for (i=0 ; i<3 ; i++)
    for (j=0 ; j<3 ; i++)
      if (a[i][j]<m) { m=a[i][j] ; row=i ; col=j ; }
  printf ("%d , %d , %d\n" , m , row , col) ;
}
```

上述程序的输出结果是【 】。

16. 以下能对二维数组 a 进行正确初始化的语句是

- A) int a[2][]={{1, 0, 1}, {5, 2, 3}};

- B) `int a[][3]={ {1, 2, 3}, {4, 5, 6} };`
 C) `int a[2][4]={ {1, 2, 3}, {4, 5}, {6} };`
 D) `int a[][3]={ {1, 0, 1} {}, {1, 1} };`
17. 合法的数组定义是
 A) `int a[]="string";` B) `int a[5]={0, 1, 2, 3, 4, 5};`
 C) `char s="string";` D) `char a[]={0, 1, 2, 3, 4, 5};`
18. 若有说明: `int a[3][4]={0};`; 则下面正确的叙述是
 A) 只有元素 `a[0][0]` 可得到初值 0
 B) 此说明语句不正确
 C) 数组 `a` 中各元素都可得到初值, 但其值不一定为 0
 D) 数组 `a` 中每个元素均可得到初值 0
19. 若有说明: `int a[3][4];`; 则数组 `a` 中各元素
 A) 可在程序的运行阶段得到初值 0
 B) 可在程序的编译阶段得到初值 0
 C) 不能得到确定的初值
 D) 可在程序的编译或运行阶段得到初值 0
20. 若有以下的定义: `int t[3][2];`; 能正确表示 `t` 数组元素地址的表达式是
 A) `&t[3][2]` B) `t[3]` C) `*t[1]` D) `t[2]`
21. 不能把字符串: `Hello!` 赋给数组 `b` 的语句是
 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. 以下不能对二维数组 `a` 进行正确初始化的语句是
 A) `int a[2][3]={0};`
 B) `int a[][3]={ {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. 若有定义和语句:
`char s[10];`
`s="abcd"; printf ("%s\n", s);`
 则结果是 (以下 `u` 代表空格)
 A) 输出 `abcd` B) 输出 `a` C) 输出 `abcduuuuu` D) 编译不通过
24. 设有数组定义: `char array []="China";` 则数组 `array` 所占的空间为
 A) 4 个字节 B) 5 个字节 C) 6 个字节 D) 7 个字节
25. 下述对 C 语言字符数组的描述中错误的是
 A) 字符数组可以存放字符串
 B) 字符数组中的字符串可以整体输入、输出

- C) 可以在赋值语句中通过赋值运算符“=”对字符数组整体赋值
D) 不可以用关系运算符对字符数组中的字符串进行比较
26. 下列描述中不正确的是
A) 字符型数组中可以存放字符串
B) 可以对字符型数组进行整体输入、输出
C) 可以对整型数组进行整体输入、输出
D) 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
27. 请读程序片段（字符串内没有空格字符）：

```
printf ("%d\n", strlen ("ATS\N012\1\\"));
```

 上面程序片段的输出结果是
 A) 11 B) 10 C) 9 D) 8
28. 若有以下程序片段：

```
char str[ ]="ab\n012\\\"";
printf ("%d\n", strlen (str)) ;
```

 上面程序片段的输出结果是
 A) 3 B) 4 C) 6 D) 12
29. 函数调用：strcat (strcpy (str1 , str2) , str3) 的功能是
 A) 将串 str1 复制到串 str2 中后再连接到串 str3 之后
 B) 将串 str1 连接到串 str2 之后再复制到串 str3 之后
 C) 将串 str2 复制到串 str1 中后再将串 str3 连接到串 str1 之后
 D) 将串 str2 连接到串 str1 之后再将串 str1 复制到串 str3 中
30. 以下函数把 b 串连接到 a 串的后面，并返回 a 中新字符串的长度。请填空。

```
Strcen (char a[], char b[])
{ int num=0, n=0;
  while (*(a+num) != 【                      】) num++;
  while (b[n]) {*(a+num)=b[n]; num++; 【                      】 ; }
  a[num]=0;
  return (num);
}
```
31. 下列程序执行后的输出结果是
 A) you&me B) you C) me D) err

```
#include "string.h"
main ()
{ char arr[2][4];
  strcpy (arr, "you"); strcpy (arr[1], "me");
  arr[0][3]='&';
  printf ("%s \n", arr); }
```
32. 请选出以下语句的输出结果


```
printf ("%d\n" , strlen ("\t\"065\xff\n")) ;
```

- A) 5 B) 14 C) 8 D) 输出项不合法，无正常输出

33. 给出以下定义：

```
char x[ ]="abcdefg";
```

```
char y[ ]={'a' , 'b' , 'c' , 'd' , 'e' , 'f' , 'g'};
```

则正确的叙述为

- A) 数组 X 和数组 Y 等价
 B) 数组 x 和数组 Y 的长度（空间）相同
 C) 数组 X 的长度（空间）大于数组 Y 的长度（空间）
 D) 数组 X 的长度（空间）小于数组 Y 的长度（空间）

34. 下面程序的功能是：将字符数组 a 中下标值为偶数的元素从小到大排列，其它元素不变。请填空。

```
#include <stdio.h>
#include <string.h>
main ()
{ char a[ ]="c language" , t ;
  int i , j , k;
  k=strlen (a);
  for (i=0 ; i<=k-2 ; i+=2)
  for (j=i+2 ; j<=k ; 【           】 )
    if ( 【           】 )
      { t=a[i] ; a[i]=a[j] ; a[j]=t ; }
  puts (a) ; printf ("\n") ; }
```

35. 请选出以下程序段的输出结果

```
#include<stdio.h>
main ()
{ char s1[10] , s2[10] , s3[10] , s4[10] ;
  scanf ("%s%s" , s1 , s2) ; gets (s3) ; gets (s4) ;
  puts (s1) ; puts (s2) ; puts (s3) ; puts (s4) ; }
```

输入数据如下：（此处<CR>代表回车符）

```
aaaa bbbb<CR>
```

```
cccc dddd<CR>
```

- | | | | |
|---------|---------|---------|-----------|
| A) aaaa | B) aaaa | C) aaaa | D) aaaa |
| bbbb | bbbb | bbbb | bbbb |
| cccc | cccc | dddd | |
| cccc | dddd | dddd | cccc dddd |

36. 下面程序的运行结果是

- A) 12ba56 B) 6521 C) 6 D) 62

37. 设有下列程序:

上述程序运行后，如果从键盘上输入（在此<CR>代表回车符）：

则程序的输出结果是【 】。

A) 6385 B) 69825 C) 63825 D) 693825

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

```

main ( )
{
    int num[26] , i ;
    char c ;
    for (i=0 ; i<26 ; i++) num[i]=0 ;
    while ( 【          】 !=' #' )
        if ( isupper (c)) num[c-65]+=1 ;
    for (i=0 ; i<26 ; i++)
        if (num[i])
            printf ("%c: %d\n" , 【          】 , num[i]) ;
}

```

41. 运行下面的程序，如果从键盘上输入：

ab <回车>

c <回车>

def <回车>则输出结果为

A) a B) a C) ab D) abcdef

b b c

c c d

d d

e

f

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

```
#define N 6
```

```
main ( )
```

```
{ char c[N];
```

```
int i=0 ;
```

```
for ( ; i<N ; c[i]=getchar ( ), i++)
```

```
for ( i=0; i<N; i++) putchar (c[i]);
```

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

```
}
```

42. 以下程序可以把从键盘上输入的十进制数（long 型）以二到十六进制形式输出，请填空。

```
main ( )
```

```
{ char b[16]={ '0' , '1' , '2' , '3' , '4' , '5' , '6' , '7' , '8' ,
               '9' , 'A' , 'B' , 'C' , 'D' , 'E' , 'F' };
```

```
int c[64] , d , i=0 , base ;
```

```
long n ;
```

```
printf ("enter a number: \n"); scanf ("%ld", &n);
```

```
printf ("enter new base: \n"); scanf ("%d", &base);
```

```

do { c[i]=【          】 ; i++ ; n=n/base ; } while (n!=0);
printf ("transmite new base: \n");
for (--i ; i>=0 ; --i)
    { d=c[i] ; printf ("%c", b【          】 ) ; }
}

```

43. 执行下列程序时输入: 123<空格>456<空格>789<回车>, 输出结果是

- A) 123, 456, 789 B) 1, 456, 789
C) 1, 23, 456, 789 D) 1, 23, 456

```

main ()
{ char s[100]; int c, i;
  scanf ("%c", &c);
  scanf ("%d", &i);
  scanf ("%s", s);
  printf ("%c, %d, %s \n", c, i, s); }

```

44. 以下程序用来对从键盘上输入的两个字符串进行比较, 然后输出两个字符串中第一个不相同字符的 ASCII 码之差。例如: 输入的两个字符串分别为 abcdef 和 abceef, 则输出为-1。请填空。

```

main ()
{ char str[100] , str2[100] , c ;
  int i , s ;
  printf ("\n input string 1: \n") ; gest (str1) ;
  printf ("\n input string 2: \n") ; gest (str2) ;
  i=0 ;
  while ((str1[i]==str2[i]) && (str1[i]!=【          】))
  i++ ;
  s=【          】 ;
  printf ("%d\n", s) ; }

```

45. 以下程序输出的结果是

```

main ()
{ char w[][10]={"ABCD" , "EFGH" , "IJKL" , "MNOP"} ;
  int k ;
  for (k=1 ; k<4 ; k++ ) printf ("%c\n" , w[k][k]) ;
}

```

- | | | | |
|---------|---------|--------|------|
| A) ABCD | B) ABCD | C) EFG | D) F |
| FGH | EFG | JK | K |
| KL | IJ | O | P |
| M | | | |

46. 下面程序的输出是

```
#include <stdio.h>
main ()
{ char ss[10]="12345";
  strcat (ss, "6789");
  gets (ss);
  printf ("%s\n", ss);
}
```

输入 ABC<CR> (表示回车)

A) ABC B) ABC9 C) 123456ABC D) ABC456789

47. 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

A) 3 B) 6 C) 10 D) 20

48. 若有定义语句: char s[100], d[100]; int j=0, i=0;, 且 s 中已赋字符串, 请填空以实现字符串拷贝。(注: 不得使用逗号表达式)

```
while([i]){ d[j]= 【                      】; j++;}
d[j]=0;
```

49. 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

50. 以下程序的输出结果是

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

51. 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

52. 以下程序的输出结果是

```

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

```

A) "AAAA" B) "BBB" C) "BBBCC" D) "CC"

53. 以下程序的输出结果是

```

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

```

A) 3 B) 4 C) 1 D) 9

54. 以下程序的输出结果是【13】。

```

main()
{   char   s[]="abcdef";
    s[3]= '\0';
    printf("%s\n",s);
}

```

55. 以下程序中函数 sort 的功能是对 a 所指数组中的数据进行由大到小的排序

```

void sort(int a[],int n)
{   int   i,j,t;
    for(i=0;i<n-1;i++)
        for(j=i+1,j<n;j++)

```

```

        if(a[i]<a[j])    {t=a[i];a[i]=a[j];a[j]=t;}
    }
    main()
    { int aa[10]={1,2,3,4,5,6,7,8,9,10}, i;
      sort(&aa[3], 5);
      for(i=0;i<10;i++)  print( "%d, ", aa[i]);
      printf( "\n" );
    }

```

程序运行后的输出结果是

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

56. 有以下程序

```

main()
{ char a[]={ 'a' , 'b' , 'c' , 'd' , 'e' , 'f' , 'g' , 'h' , '\0' };
  int i, j;
  i=sizeof(a);   j=strlen(a);
  printf( "%d,%d\b" i, j);
}

```

程序运行后的输出结果是

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

57. 有以下程序

```

main()
{ int aa[4][4]={ {1,2,3,4}, {5,6,7,8}, {3,9,10,2}, {4,2,9,6} };
  int i, s=0
  for(i=0;i<4;i++)  s+=aa[i][1];
  printf( "%d\n", s);
}

```

程序运行后的输出结果是

- A) 11 B) 19 C) 13 D) 20

58. 以下程序中函数 f 的功能是将 n 个字符串按由大到小的顺序进行排序。

```

#include <string.h>
void f(char p[][10], int n)
{ char t[20];   int i, j;
  for(i=0; i<n-1; i++)
    for(j=i+1; j<n; j++)
      if(strcmp(p[i], p[j])<0)
        { strcpy(t, p[i]); strcpy(p[i], p[j]); strcpy(p[j], t); }
}

```

```
}  
main()  
{ char p[][10]={ "abc", "aabdfg", "abbd", "dcdbbe", "cd" };int i;  
  f(p,5);  printf( "%d\n",strlen(p[0]));  
}
```

程序运行后的输出结果是

- A) 6 B) 4 C) 5 D) 3

第八章 函 数

1. 函数 pi 的功能是根据以下近似公式求 π 值:

$$(\pi * \pi) / 6 = 1 + 1 / (2 * 2) + 1 / (3 * 3) + \dots + 1 / (n * n)$$

现在请你在下面的函数中填空, 完成求 π 的功能。

```
#include "math.h"
double pi (long n)
{ double s=0.0; long i;
  for (i=1; i<=n; i++) s=s+【          】;
  return (sqrt (6*s)); }
```

2. 若函数 fun 的类型为 void , 且有以下定义和调用语句:

```
#define M 50
main ()
{ int a[M] ;
  ...
  fun (a) ;
  ... }
```

定义 fun 函数首部可以用三种不同的形式 , 请写出这三种形式:【 】、
【 】、【 】。

(注意: ①形参的名字请用 q , ②使用同一种风格)。

3. 函数 fun 的功能是: 使一个字符串按逆序存放, 请填空。

```
void fun (char str[ ])
{ char m ; int i , j ;
  for (i=0 , j=strlen (str) ; i<【          】 ; i++ , j--)
  { m=str[i];
    str[i]=【          】;
    str[j-1]=m; }
  printf ("%s\n", str); }
```

4. 设在主函数中有以下定义和函数调用语句, 且 fun 函数为 void 类型; 请写出 fun 函数的首部【 】。要求形参名为 b。

```
main ()
{ double s[10][22];
  int n ;
  ...
  fun (s); }
```

... }

5. 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

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

6. 下面程序的输出结果是

```
#include<stdio.h>
main ()
{ int i=2 , p ;
  p=f (i , i+1) ;
  printf ("%d" , p) ;
}
int f (int a , int b)
{ int c ;
  if (a>b) c=1 ;
  else if (a==b) c=0 ;
  else c=-1 ;
  return (c) ; }
```

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

7. 请读程序:

```
# include < stdio.h>
FUNC ( int A , int B)
{ int C ; C=A+B ; return C ; }
main ( )
{ int X=6 , Y=7 , Z=8 , R ;
  R=FUNC ((X--, Y++, X+Y), Z--);
  printf ("%d\n", R); }
```

上面程序的输出结果是【 】。

8. 以下正确的说法是

建立函数的目的之一是

- A) 提高程序的执行效率
- B) 提高程序的可读性
- C) 减少程序的篇幅
- D) 减少程序文件所占内存

9. 以下正确的说法是

- A) 用户若需调用标准库函数，调用前必须重新定义
- B) 用户可以重新定义标准库函数，若如此，该函数将失去原有含义
- C) 系统根本不允许用户重新定义标准库函数
- D) 用户若需调用标准库函数，调用前不必使用预编译命令将该函数所在文件包到用户源文件中，系统自动去调

10. 设有如下的函数

```
ggg (float x)
{ printf ("\n%d" , x*x) ; }
```

则函数的类型

- A) 与参数 x 的类型相同
- B) void
- C) int
- D) 无法确定

11. 下面程序的输出是【 】。

```
unsigned fun6 (unsigned int num)
{ unsigned int k=1;
  do { k *=num%10; num/=10; } while (num);
  return (k); }

main ()
{ unsigned int n=26;
  printf ("%d\n", fun6 (n)); }
```

12. 有以下程序

```
void fun (int a , int b , int c)
{ a=456 ; b=567 ; c=678 ; }

main ()
{ int x=10 , y=20 , z=30 ;
  fun (x , y , z) ;
  printf ("%d ; %d; %d \n" , x , y, z) ; }
```

输出结果是

- A) 30 ; 20 ; 10
- B) 10 ; 20 ; 30
- C) 456 ; 567 ; 678
- D) 678 ; 567 ; 456

13. 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) ; }
```

该程序的输出的结果是

- A) 12 B) 13 C) 14 D) 15

14. 有以下函数

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

该函数的返回值是

- A) 无确切的值 B) 形参 p 中存放的地址值
C) 一个临时存储单元的地址 D) 形参 p 自身的地址值

15. 以下正确的说法是

在 C 语言中，函数调用时，

- A) 实参和与其对应的形参各占用独立的存储单元
B) 实参和与其对应的形参共用一个的存储单元
C) 只有当实参和与其对应的形参同名时才共用存储单元
D) 形参是虚拟的，不占用存储单元

16. C 语言规定，简单变量做实参时，它和对应形参之间的数据传递方式是

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

17. 若调用一个函数，且此函数中没有 return 语句，则正确的说法是

该函数

- 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

18. 以下不正确的说法是

C 语言规定：

- A) 实参可以是常量、变量或表达式
B) 形参可以是常量、变量或表达式
C) 实参可以为任意类型
D) 形参应与其对应的实参类型一致

19. 以下说法正确的是

- A) 定义函数时，形参的类型说明可以放在函数体内
B) return 后边的值不能为表达式
C) 如果函数值的类型与返回值的类型不一致，以函数值的类型为准
D) 如果形参与实参的类型不一致，以实参类型为准

20. C 语言允许函数值类型缺省定义，此时该函数值隐含的类型是

- A) float 型 B) int 型 C) long 型 D) double 型

21. C 语言规定，函数返回值的类型是由

- A) return 语句中的表达式类型决定
 B) 调用该函数时的主调函数类型所决定
 C) 调用该函数时系统临时决定
 D) 在定义该函数时所指定的函数类型所决定
22. 有如下函数调用语句
`func (rec1, rec2+rec3, (rec4, rec5));`
 该函数调用语句中, 含有的实参个数是
 A) 3 B) 4 C) 5 D) 有语法错
23. 以下错误的描述是
 函数调用可以
 A) 出现在执行语句中 B) 出现在一个表达式中
 C) 做为一个函数的实参 D) 做为一个函数的形参
24. 下列程序执行后的输出结果是
 A) hello B) hel C) hlo D) hlm
- ```

void func1 (int i);
void func2 (int i);
char st[]="hello, friend!";
void func1 (int i)
{ printf ("%c", st[i]);
 if (i<3) { i+=2 ; func2 (i) ; }
}
void func2 (int i)
{ printf ("%c", st[i]);
 if (i<3) { i+=2 ; func1 (i) ; }}
main ()
{ int i=0 ; func1 (i) ; printf ("\n"); }

```
25. C 语言规定, 程序中各函数之间  
 A) 既允许直接递归调用也允许间接递归调用  
 B) 不允许直接递归调用也不允许间接递归调用  
 C) 允许直接递归调用不允许间接递归调用  
 D) 不允许直接递归调用允许间接递归调用
26. 以下正确的描述是  
 在 C 语言程序中  
 A) 函数的定义可以嵌套, 但函数的调用不可以嵌套  
 B) 函数的定义不可以嵌套, 但函数的调用可以嵌套  
 C) 函数的定义和函数的调用均不可以嵌套  
 D) 函数的定义和函数的调用均可以嵌套
27. 下面程序的输出结果是【                      】。

```

fun (int x)
{ int p ;
 if (x==0 || x==1) return (3);
 p=x-fun (x-2);
 return p; }
main ()
{ printf ("%d\n", fun (4)); }

```

28. 下面程序的输出是【                      】。

```

long fun5 (int n)
{ long s ;
 if ((n==1) || (n==2)) s=2;
 else s=n+fun5 (n-1);
 return (s); }
main ()
{ long x ;
 x=fun5 (4);
 printf ("%ld\n", x) ; }

```

29. 有如下程序

```

long fib (int n)
{ if (n > 2) return (fib (n-1) + fib (n-2));
 else return (2);
}
main ()
{ printf ("%d\n", fib (3)); }

```

该程序的输出结果是

A) 2          B) 4          C) 6          D) 8

30. 若输入 3 个整数 3，2，1，则下面程序的输出结果是【                      】。

```

include <stdio.h>
void sub (int n , int uu[])
{ int t ;
 t=uu[n--] ; t+=3*uu[n] ;
 n=n++ ;
 if (t>=10) { uu[n++]=t/10 ; uu[n]=t%10 ; }
 else uu[n]= t ; }
main ()
{ int i , n , aa[10]={0 , 0 , 0 , 0 , 0 , 0 }
 scanf ("%d%d%d" , &n , &aa[0] , &aa[1]) ;
 for (i=1 ; i<n ; i++) sub (i , aa) ;

```

```

 for (i=0 ; i<=n ; i++) printf ("%d" , aa[i]) ;
 printf ("\n") ; }

```

31. 用数组名作为函数调用的实参, 传递给形参的是  
 A) 数组的首地址                      B) 数组第一个元素的值  
 C) 数组中全部元素的值              D) 数组元素的个数
32. 若使用一维数组名作为函数的实参, 则以下正确的说法是  
 A) 必须在主调函数中说明此数组的大小  
 B) 实参数组类型与形参数组类型可以不匹配  
 C) 在被调函数中, 不需要考虑形参数组的大小  
 D) 实参数组名与形参数组名必须一致

33. 下列程序执行后输出的结果是

```

#include <stdio.h>
f (int a)
{ int b=0 ;
 static int c=3 ;
 a=c++ , b++ ;
 return (a) ; }
main ()
{ int a=2 , i , k ;
 for (i=0 ; i<2 ; i++) k=f (a++) ;
 printf ("%d\n", k); }
A) 3 B) 0 C) 5 D) 4

```

34. 以下叙述中不正确的是  
 A) 在 C 中, 函数中的自动变量可赋初值, 每调用一次, 赋一次初值。  
 B) 在 C 中调用函数时, 实际参数和对应形参在类型上只需赋值兼容。  
 C) 在 C 中, 外部变量的隐含类别是自动存储类别。  
 D) 在 C 中, 函数形参可以说明为 register 变量。

35. 下面程序的输出是

```

A) 3 B) 4 C) 6 D) 9
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); }
```

36. 以下程序的输出结果是【                      】。

```
void fun ()
{ static int a=0 ;
 a+=2; printf (" %d", a); }
main ()
{ int cc;
 for (cc=1 ; cc<4 ; cc++) fun () ;
 printf (" \n"); }
```

37. 以下程序的输出结果是

A) 699          B) 669          C) 61515          D) 6615

```
int d=1;
fun (int p)
{ static int d=5;
 d+=p; printf ("%d ", d);
 return (d); }
main ()
{ int a=3 ; printf ("%d \n", fun (a+fun (d))) ; }
```

38. 以下正确的说法是

如果在一个函数中的复合语句中定义了一个变量，则该变量

- A) 只在该复合语句中有效          B) 在该函数中有效  
C) 在本程序范围内均有效          D) 为非法变量

39. C 语言中，函数的隐含存储类别是

A) auto          B) static          C) extern          D) 无存储类别

40. 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 语言所有函数都是外部函数

41. 下程序运行后，输出结果是

A) 8, 15          B) 8, 16          C) 8, 17          D) 8, 8

```
func (int a, int b)
{ static int m=0 , i=2 ;
 i+=m+1 ; m=i+a+b ;
 return (m);
}
main ()
{ int k=4 , m=1 , p ;
```



```

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

```

42. 下列程序执行后输出的结果是

```

int d=1;
fun (int p)
{ int d=5 ;
 d +=p ++ ;
 printf ("%d", d) ;
}
main ()
{ int a=3 ;
 fun (a) ; d += a ++ ;
 printf ("%d\n", d) ; }
A) 8 4 B) 9 6 C) 9 4 D) 8 5

```

43. 下面程序的输出是

```

A) 1 B) 2 C) 7 D) 10
int m=13;
int fun2 (int x , int y)
{ int m=3 ;
 return (x*y-m) ;
}
main ()
{ int a=7 , b=5 ;
 printf ("%d\n", fun2 (a, b) /m) ; }

```

44. 以下叙述中不正确的是

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

45. 以下对 C 语言函数的有关描述中 ， 正确的是

- A) C 调用函数时，只能把实参的值传给形参，形参的值不能回传给实参
- B) C 函数既可以嵌套定义又可以递归调用
- C) 函数必须有返回值 ， 否则不能使用函数
- D) C 程序中有调用关系的所有函数必须放在同一个源程序文件中

46. 下面程序的输出结果是【 】。

```

include<stdio.h>
main ()
{ int a=3 , b=2 , c=1 ;

```

```

 c-=++b ; b*=a+c ;
 { int b=5 , c=12 ; c/=b*2 ; a-=c ;
 printf (" %d , %d , %d , " , a , b , c) ; a+=--c ; }
 printf (" %d , %d , %d , " , a , b , c) ;
 }

```

47. 若已定义: int a[10], i;, 以下 fun 函数的功能是: 在第一个循环中给前 10 个数组元素依次赋 1、2、3、4、5、6、7、8、9、10; 在第二个循环中使 a 数组前 10 个元素中的值对称折叠, 变成 1、2、3、4、5、5、4、3、2、1。请填空。

```

fun(int a[])
{ int i;
 for(i=1; i<=10; i++) 【 】=i;
 for(i=0; i<5; i++) 【 】=a[i];
}

```

48. 以下函数的功能是: 求 x 的 y 次方, 请填空。

```

double fun(double x, int y)
{ int i;
 double z;
 for(i=1, z=x; i<y; i++) z=z* 【 】;
 return z;
}

```

49. 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); }

```

则以下叙述中不正确的是

- 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

50. 在 C 语言中, 形参的缺省存储类是

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

50. 以下程序的输出的结果是

```
int x=3;
main()
{ int i;
 for (i=1;i<x;i++) incre();
}
ncre()
{ staic int x=1;
 x*=x+1;
 printf(" %d", x);
}
```

A) 3 3                      B) 2 2                      C) 2 6                      D) 2 5

51. 以下程序的输出结果是【                    】。

```
main()
{ int x=0;
 sub(&x, 8, 1);
 printf("%d\n", x);
}
sub(int *a, int n, int k)
{ if(k<=n) sub(a, n/2, 2*k);
 *a+=k;
}
```

52. 有以下程序

```
void fun(char *c, int d)
{ *c=*c+1; d=d+1;
 printf("%c, %c, ", *c, d);
}
main()
{ char a=' A' , b=' a' ;
 fun(&b, a); printf("%c, %c\n" , a, b);
}
```

程序运行后的输出结果是

A) B, a, B, a                      B) a, B, a, B                      C) A, b, A, b                      D) b, B, A, b

53. 以下程序中函数 reverse 的功能是将 a 所指数组中的内容进行逆置。

```
void reverse(int a[], int n)
{ int i, t;
 for (i=0; i<n/2; i++)
```

```

 { t=a[i]; a[i]=a[n-1-i];a[n-1-i]=t;}
}
main()
{ int b[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; int i, s=0;
 reverse(b, 8);
 for(i=6; i<10; i++) s+=b[i];
 printf("%d\n", s);
}

```

程序运行后的输出结果是

- A) 22                      B) 10                      C) 34                      D) 30

54. 有以下程序

```

int a=3;
main()
{ int s=0;
 { int a=5; s+=a++; }
 s+=a++;printf("%d\n", s);
}

```

程序运行后的输出结果是

- A) 8                      B) 10                      C) 7                      D) 11

55. fun 函数的功能是：首先对 a 所指的 N 行 N 列的矩阵，找出各行中的最大的数，再求这 N 个最大值中的最小的那个数作为函数值返回。请填空。

```

#include <stdio.h>
#define N 100
int fun(int(*a)[N])
{ int row,col,max,min;
 for(row=0;row<N;row++)
 { for(max=a[row][0],col=1;col<N;col++)
 if(_____)max=a[row][col];
 if(row==0)min=max;
 else if(_____)min=max;
 }
 return min;
}

```

56. 下面程序的运行结果是：\_\_\_\_\_。

```

void swap(int *a, int *b)
{ int *t;
 t=a; a=b; b=t;
}

```

```
}
main()
{ int x=3, y=5, *p=&x, *q=&y;
 swap(p, q);
 printf("%d%d\n" , *p, *q);
}
```

## 第九章 编译预处理

1. 下面叙述中不正确的是

- A) 预处理命令行都必须以#号开始
- B) 在程序中凡是以#号开始和语句行都是预处理命令行
- C) C 程序在执行过程中对预处理命令行进行处理
- D) 以下是正确的宏定义

```
#define IBM_PC
```

2. 以下叙述中正确的是

- A) 程序的一行上可以出现多个有效的预处理命令行
- B) 使用带参的宏时，参数的类型应与宏定义时的一致
- C) 宏替换不占用运行时间，只占用编译时间
- D) 在以下定义中 C R 是称为“宏名”的标识符

```
define C R 045
```

3. 有关宏替换的叙述不正确的是

- A) 宏替换不占用运行时间
- B) 宏名无类型
- C) 宏替换只是字符替换
- D) 宏名必须用大写字母表示

4. C 语言的编译系统对宏命令的处理是

- A) 在程序运行时进行的
- B) 在程序连接时进行的
- C) 和 C 程序中的其它语句同时进行编译的
- D) 在对源程序中其它成份正式编译之前进行的

5. 在宏定义 #define PI 3.14159 中，用宏名 PI 代替一个

- A) 单精度数
- B) 双精度数
- C) 常量
- D) 字符串

6. 以下程序的输出结果是

- A) 9
- B) 6
- C) 36
- D) 18

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

7. 执行下面的程序后，a 的值是

```
#define SQR(X) X*X
main ()
{ int a=10, k=2, m=1;
```

```
a/=SQR (k+m) /SQR (k+m);
printf ("%d\n", a); }
```

A) 10      B) 1      C) 9      D) 0

8. 以下说法中正确的是

- A) #define 和 printf 都是 C 语句
- B) #define 是 C 语句，而 printf 不是
- C) printf 是 C 语句，但#define 不是
- D) #define 和 printf 都不是 C 语句

9. 请读程序：

```
#include< stdio.h>
#define SUB (X, Y) (X) * Y
main ()
{ int a=3, b=4;
 printf ("%d\n", SUB (a++, b++)); }
```

上面程序的输出结果是

A) 12    B) 15      C) 16    D) 20

10. 请选出以下程序段的输出结果

```
#include<stdio.h>
#define MIN (x , y) (x) < (y) ? (x): (y)
main ()
{ int i , j , k ;
 i=10 ; j=15 ;
 k=10*MIN (i , j) ;
 printf ("%d\n" , k) ; }
```

A) 15      B) 100      C) 10      D) 150

11. 下列程序执行后的输出结果是

A) 6      B) 8      C) 10      D) 12

```
#define MA (x) x* (x-1)
main () { int a=1 , b=2 ; printf ("%d \n", MA (1+a+b)); }
```

12. 请选出以下程序段的输出结果

```
#include<stdio.h>
#define MIN (x , y) (x<y) ? (x): (y)
main ()
{ int i , j , k ;
 i=10 ; j=15 ;
 k=10*MIN (i , j) ;
 printf ("%d\n" , k) ; }
```

A) 15      B) 100      C) 10      D) 150

13. 请选出以下程序段的输出结果

```
#include<stdio.h>
#define MIN (x , y) (x<y? x: y)
main ()
{ int i , j , k ;
 i=10 ; j=15 ;
 k=10*MIN (i , j) ;
 printf ("%d\n" , k) ; }
```

A) 15      B) 100      C) 10      D) 150

14. 有如下程序

```
#define N 2
#define M N+1
#define NUM 2*M+1
main ()
{ int i ;
 for (i=1 ; i<=NUM ; i++) printf ("%d\n", i); }
```

该程序中的 for 循环执行的次数是

A) 5      B) 6      C) 7      D) 8

15. 设有以下宏定义:

```
#define N 3
#define Y (n) ((N+1)*n)
```

则执行语句:  $z=2 * (N+Y (5+1))$ ; 后, z 的值为

A) 出错      B) 42      C) 48      D) 54

16. 以下程序运行后, 输出结果是

A) 49.5      B) 9.5      C) 22.0      D) 45.0

```
#include<stdio.h>
#define PT 5.5
#define S (x) PT*x*x
main ()
{ int a=1 , b=2 ; printf ("%4.1f\n", S (a+b)) ; }
```

17. 阅读下列程序:

```
include <stdio.h>
define MAX_COUNT 4
void fun () ;
main ()
{ int count ;
 for (count=1 ; count<=MAX_COUNT ; count++) fun () ; }
void fun ()
```



```

{ static int i ;
 i+=2 ; printf ("%d" , i) ; }

```

上述程序的输出结果是【                      】。

18. 在“文件包含”预处理语句的使用形式中，当#include 后面的文件名用“ ”（双引号）括起来时，寻找包含文件的方式是

- A) 直接按系统设定的标准方式搜索目录
- B) 先在源程序所在目录搜索，再按系统设定的标准方式搜索
- C) 仅仅搜索源程序所在目录
- D) 仅仅搜索当前目录

19. 以下正确的描述是

- A) C 语言的预处理功能是指完成宏替换和包含文件的调用
- B) 预处理指令只能位于 C 源程序文件的首部
- C) 凡是 C 源程序中行首以“#”标识的控制行都是预处理指令
- D) C 语言的编译预处理就是对源程序进行初步的语法检查

20. 以下程序的输出结果是

```

#define M(x, y, z) x*y+z
main()
{ int a=1, b=2, c=3;
 printf("%d\n" , M(a+b, b+c, c+a));
}

```

- A) 19                      B) 17                      C) 15                      D) 12

21. 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

22. 设有如下宏定义

```

#define MYSWAP(z, x, y) {z=x; x=y; y=z;}

```

以下程序段通过宏调用实现变量 a、b 内容交换, 请填空。

```

float a=5, b=16, c;

```

```

MYSWAP(【 】 , a, b);

```

23. 程序中头文件 type1.h 的内容是:

```

#define N 5

```

```

#define M1 N*3

```

程序如下:

```

#define "type1.h"

```

```
#define M2 N*2
main()
{ int i;
 i=M1+M2; printf("%d\n", i);
}
```

程序编译后运行的输出结果是：

- A) 10                      B) 20                      C) 25                      D) 30

## 第十章 指 针

1. 若 `int x, *pb;` , 则正确的赋值表达式是

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

2. 有如下程序段

```
int *p, a=10, b=1;
```

```
p=&a; a=*p+b;
```

执行该程序段后, `a` 的值为

- A) 12      B) 11      C) 10      D) 编译出错

3. 若有以下定义和语句:

```
double r=99, *p=&r;
```

```
*p=r;
```

则以下正确的叙述是

- A) 以下两处的`*p` 含义相同, 都说明给指针变量 `p` 赋值  
B) 在“`double r=99, *p=&r;`”中, 把 `r` 的地址赋值给了 `p` 所指的存储单元  
C) 语句“`*p=r;`”把变量 `r` 的值赋给指针变量 `p`  
D) 语句“`*p=r;`”取变量 `r` 的值放回 `r` 中

4. 下列程序的输出结果是

- A) 4      B) 6      C) 8      D) 10

```
int b=2;
int func(int *a)
{ b += *a; return (b); }
main()
{ int a=2, res=2;
 res += func(&a);
 printf("%d\n", res); }
```

5. 下面函数用来求出两个整数之和, 并通过形参传回两数相加之和值, 请填空。

```
int add(int x, int y, 【 】 z)
{ 【 】 =x+y; }
```

6. 下面程序的输出是

- A) 23      B) 24      C) 25      D) 26

```
void prtv(int *x)
{ printf("%d\n", ++*x); }
main()
{ int a=25; prtv(&a); }
```

7. 下面程序的输出结果是 【                      】。

```
void as (int x , int y , int *cp , int *dp)
{ *cp=x+y; *dp=x-y; }
main ()
{ int a=4 , b=3 , c , d ;
 as (a , b , &c , &d);
 printf ("%d %d\\", c, d); }
```

8. 请选出以下程序的输出结果

```
#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
```

9. 有以下程序

```
int fun (int x , int y , int *cp , int *dp)
{ *cp=x+y; *dp=x-y; }
main ()
{ int a , b , c , d ;
 a=30; b=50;
 fun (a , b , &c , &d) ;
 printf ("%d ; %d\\n" , c , d) ; }
```

输出结果是

- A) 50 ; 30      B) 30 ; 50      C) 80 ; -20      D) 80 ; 20
10. 要求函数的功能是交换 x 和 y 中的值, 且通过正确调用返回交换结果. 能正确执行此功能的函数是
- A) funa (int \*x, int \*y) { int \*p; \*p=\*x; \*x=\*y; \*y=\*p; }
- B) fund (int x, int y) { int t; t=x; x=y; y=t; }
- C) func (int \*x, int \*y) { \*x=\*y; \*y=\*x; }
- D) fund (int \*x, int \*y) { \*x=\*x+\*y; \*y=\*x-\*y; \*x=\*x-\*y; }
11. 若有说明: long \*p , a; 则不能通过 scanf 语句正确给输入项读入数据的程序段是
- A) \*p=&a; scanf ("%ld", p);
- B) p= (long \*) malloc (8); scanf ("%ld", p);
- C) scanf ("%ld", p=&a);

D) scanf ("%ld", &a);

12. 以下程序执行后, a 的值是

- A) 4      B) 1      C) 0      D) 运行时出错, 无定值

```
main ()
{
 int a, k=4, m=6, *p1=&k, *p2=&m;
 a=p1= &m;
 printf ("%d\n", a);
}
```

13. fun1 函数的调用语句为: fun1 (&a, &b, &c); 它将三个整数按由大到小的顺序调整后依次放入 a, b, c, 三个变量中, a 中放大数。请填空。

```
void fun2 (int *x, int *y)
{
 int t;
 t=*x; *x=*y; *y=t;
}
void fun1 (int *pa, int *pb, int *pc)
{
 if 【 】 ;
 if 【 】 ;
 if (*pc>*pb) fun2 (【 】);
}
```

14. 对于类型相同的两个指针变量之间, 不能进行的运算是

- A) <      B) =      C) +      D) -

15. 若有以下定义, 则不移动指针 p, 且通过指针 p 引用值为 98 的数组元素的表达式是【 】。

```
int w[10]={23, 54, 10, 33, 47, 98, 72, 80, 61}, *p=w;
```

16. 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 数组元素不正确的引用上

- A) a[p-a]      B) \*(&a[i])      C) p[i]      D) a[10]

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

- A) p+1      B) a+1      C) a++      D) ++p

18. 若有以下说明:

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

则数值为 6 的表达式是

- A) \*p+6      B) \*(p+6)      C) \*p+=5      D) p+5

19. 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, 则不能正确执行的语句是

- A) k=\*P1+\*P2;      B) p2=k;      C) P1=P2;      D) k=\*P1 \* (\*P2);

20. 下面程序的输出是

- A) 3      B) 4      C) 1      D) 2

```
main ()
```

```

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

```

21. 若有以下定义:

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

则值为 3 的表式是

- A) p+=2 , \* (p++)      B) p+=2 , \*++p  
C) p+=3 , \*p++      D) p+=2 , ++\*p

22. 若有以下定义和语句:

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

则不能表示 a 数组元素的表达式是

- A) \*p      B) a[10]      C) \*a      D) a[p-a]

23. 阅读程序:

```

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

```

该程序有两个 printf 语句, 如果第一个 printf 语句输出的是 194, 则第二个 printf 语句的输出结果是

- A) 203    B) 204      C) 1a4    D) 1a6

24. 请选出正确的程序段

- |                  |               |
|------------------|---------------|
| A) int *p;       | B) int *s, k; |
| scanf ("%d", p); | *s=100;       |
| ...              | ...           |
| C) int *s, k;    | D) int *s, k; |
| char *p, c;      | char *p, e;   |
| s=&k;            | s=&k;         |
| p=&c;            | p=&c;         |
| *p='a';          | s=p;          |
| ...              | *s=1;         |
|                  | ...           |

25. 有如下说明

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

则数值为 9 的表达式是

- A) \*p+9      B) \* (p+8)      C) \*p+=9      D) p+8

26. 已知指针 p 的指向如下图所示, 则执行语句 \*--p; 后\*p 的值是

- A) 30      B) 20      C) 19      D) 29

a[0] a[1] a[2] a[3] a[4]

|    |    |    |    |    |
|----|----|----|----|----|
| 10 | 20 | 30 | 40 | 50 |
|----|----|----|----|----|

↑

p

27. 执行以下程序后, y 的值是

```
main ()
{ int a[]={2, 4, 6, 8, 10};
 int y=1, x, *p;
 p=&a[1];
 for (x=0; x<3; x++)
 y += * (p + x);
 printf ("%d\n", y); }
```

A) 17 B) 18 C) 19 D) 20

28. 设有如下定义:

```
int arr[]={6 , 8 , 9 , 10};
int * ptr;
```

则下列程序段的输出结果为

```
ptr=arr ;
* (ptr+2) +=2 ;
printf ("%d ; %d\n" , *ptr , * (ptr+2));
```

A) 8 ; 10 B) 6 ; 8 C) 7 ; 9 D) 6 ; 10

29. 下列程序执行后的输出结果是

```
A) 6 B) 7 C) 8 D) 9
void func (int *a, int b[])
{ b[0]=*a+6; }
main ()
{ int a , b[5];
 a=0; b[0]=3;
 func (&a, b); printf ("%d \n", b[0]); }
```

30. 下面程序把数组元素中的最大值放入 a[0] 中。则在 if 语句中的条件表达式应该是

A) p>a B) \*p>a[0] C) \*p>\*a[0] D) \*p[0]> \*a[0]

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

31. 以下函数返回 a 所指数组中最小的值所在的下标值

```
fun (int *a , int n)
{ int i , j=0 , p ;
 p=j ;
 for (i=j ; i<n ; i++)
 if (a[i]<a[p])【 】
 return (p); }
```

在下划线处应填入的是

- A) i=p      B) a[p]=a[i]      C) p=j      D) p=i

32. 以下程序的运行结果是

- A) 运行后报错      B) 6 6      C) 6 12      D) 5 5

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

33. 要求函数的功能是在一维数组 a 中查找 x 值；若找到则返回所在的下标值，否则返回 0；数列放在 a[1] 到 a[n] 中。不能正确执行此功能的函数是

- A) funa (int \*a , int n , int x)  
    { \*a=x; while (a[n]!=x) n--; return n; }
- B) funb (int \*a , int n , int x)  
    { int k; for (k=1; k<=n; k++)  
        if (a[k]==x) return k; return 0; }
- C) func (int a[ ] , int n , int x)  
    { int \*k; a[0]=x; k=a+n; while (\*k!=x) k--; return k-n; }
- D) fund (int a[ ] , int n , int x)  
    { int k=0; do{ k++; } while ((k<n+1) && (a[k]!=x)); }

34. 下面程序输出数组中的最大值，由 s 指针指向该元素。

```
main ()
{ int a[10]={6, 7, 2, 9, 1, 10, 5, 8, 4, 3}, *p, *s;
 for (p=a, s=a; p-a<10; p++)
 if (【 】) s=p;
 printf ("The max: %d", *s); }
```

则在 if 语句中的判断表达式应该是

- A) p>s    B) \*p>\*s    C) a[p]>a[s]    D) p-a>p-s

35. 若有以下定义和语句：

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



则不能表示 a 数组元素的表达式是

- A) \*p      B) a[10]      C) \*a      D) a[p-a]

36. 若有以下调用语句, 则不正确的 fun 函数的首部是

- 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)

```
main ()
{ ...
 int a[50], n;
 ...
 fun (n, &a[9]);
 ... }
```

37. 请选出以下程序的输出结果

```
#include<stdio.h>
fun (int *s, int n1, int n2)
{ int i, j, t;
 i=n1; j=n2;
 while (i<j)
 { t= *(s+ i) ;
 *(s+i) = *(s+j) ;
 *(s+j) =t ;
 i++ ; j-- ; } }
main ()
{ int a[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 0}, i, *p=a ;
 fun (p, 0, 3) ;
 fun (p, 4, 9) ;
 fun (p, 0, 9) ;
 for (i=0; i<10; i++) printf ("%d", *(a+i)) ;
 printf ("\n") ; }
A) 0 9 8 7 6 5 4 3 2 1 B) 4 3 2 1 0 9 8 7 6 5
C) 5 6 7 8 9 0 1 2 3 4 D) 0 9 8 7 6 5 1 2 3 4
```

38. 请读程序:

```
#include <stdio.h>
int A[]={2, 4, 6, 8};
main ()
{ int l, *P=A;
 for (l=0; l<4; l++) A[l]= *P++;
 printf ("%d\n", A[2]); }
```

上面程序输出结果是

A) 6          B) 8          C) 4          D) 2

39. 下面程序的输出是【                      】。

```
#define PR (ar) printf ("%d", ar)
main ()
{ int j , a[]={1, 3, 5, 7, 9, 11, 13, 15}, *p=a+5;
 for (j=3; j; j--)
 { switch (j)
 { case 1:
 case 2: PR (*p++); break;
 case 3: PR (* (--p)); }
 }
}
```

40. 请读程序:

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

上面程序输出结果是

A) 720          B) 120          C) 24          D) 6

41. 请读程序:

```
#include< stdio.h>
void FUN (int *S)
{ static int J=0;
 do {S[J] +=S[J+1]; }while (++J<2); }
main ()
{ int K, A[10]={ 1, 2, 3, 4, 5};
 for (K=1; K<3; K++) FUN (A);
 for (K=0; K<5; K++) printf ("%d", A[K]); }
```

上面程序输出结果是

A) 34756      B) 23445          C) 35745      D) 12345

42. 请选出以下程序的输出结果

```
#include<stdio.h>
main ()
{ int a[]={ 1 , 2 , 3 , 4 } , i ;
 int x=0 ;
```

```

 for (i=0 ; i<4 ; i++)
 { sub (a , x) ; printf ("%d" , x) ; }
 printf ("\n") ; }
sub (int *s , int y)
{ static int t=3 ;
 y=s[t] ; t-- ; }
A) 1 2 3 4 B) 4 3 2 1 C) 0 0 0 0 D) 4 4 4 4

```

43. 若有以下定义和语句:

```

int w[2][3] , (*pw) [3] ; pw=w ;

```

则对 w 数组元素非法引用是

A) \* (w[0]+2)      B) \*pw+1) [2]      C) pw[0][0]      D) \* (pw[1]+2)

44. 若有以下说明和语句 , 请选出哪个是对 c 数组元素的正确引用

```

int c[4][5] , (*cp) [5] ;
cp=c ;
A) cp+1 B) * (cp+3) C) * (cp+1) +3 D) * (*cp+2)

```

45. 执行以下程序段后 , m 的值为

```

int a[2][3]={ {1 , 2 , 3} , {4 , 5 , 6} };
int m , *p ;
p=&a[0][0];
m= (*p) * (* (p+2)) * (* (p+4));
A) 15 B) 14 C) 13 D) 12

```

46. 下面程序的输出是

```

A) 60 B) 68 C) 99 D) 108
main ()
{ int a[3][4]={1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23};
 int (*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) ; }

```

47. 设有以下定义的语句:

```

int a[3][2]={10 , 20 , 30 , 40 , 50 , 60} , (*p) [2];
p=a;

```

则\* (\* (p+2) +1) 值为【                      】。

48. 若有以下说明:

```

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

```

则数值为 4 的表达式是

A) \*w[1]+1      B) p++, \* (p+1)

C) w[2][2]                      D) p[1][1]

49. 下列程序执行后的输出结果是

A) 3              B) 6              C) 9              D) 随机数

```
main ()
{ int a[3][3], *p, i;
 p=&a[0][0];
 for (i=p; i<9; i++) p[i]=i+1;
 printf ("%d \n", a[1][2]); }
```

50. 以下程序的输出结果是

A) 1              B) 4              C) 7              D) 5

```
int a[3][3]={1, 2, 3, 4, 5, 6, 7, 8, 9, }, *p;
main ()
{ p= (int *) malloc (sizeof (int));
 f (p , a); printf ("%d \n", *p); }
f (int *s , int p[][3])
{ *s=p[1][1]; }
```

51. 下面程序输出是

A) 23              B) 26              C) 33              D) 36

```
int aa[3][3]={ {2}, {4}, {6} };
main ()
{ int i , *p=&aa[0][0];
 for (i=0; i<2; i++)
 { if (i==1) aa[i][i+1]=*p+1;
 else ++p;
 printf ("%d", *p); }}
```

52. 若有以下定义:

char s[20]="programming" , \*ps=s ;

则不能代表字符 o 的表达式是

A) ps+2              B) s[2]              C) ps[2]              D) ps+=2 , \*ps

53. 设有如下的程序段:

```
char str[]="Hello" ;
char *ptr ;
ptr=str ;
```

执行上面的程序段后 , \*(ptr+5) 的值为

A) 'o'              B) '\0'              C) 不确定的值              D) 'o' 的地址

54. 设已有定义: char \*st="how are you"; 下列程序段中正确的是

A) char a[11], \*p; strcpy (p=a+1, &st[4]);  
B) char a[11]; strcpy (++a, st);

- C) char a[11]; strcpy (a, st);  
D) char a[ ], \*p; strcpy (p=&a[1], st+2);
55. 下列程序的输出结果是【                      】。
- ```
main ()
{ char b[ ]="ABCDEFGH";
  char *chp=&b[7];
  while (--chp>&b[0])
    putchar (*chp);
  putchar ('\n'); }
```
56. 设有如下程序段，运行后的输出结果为
- ```
static char str[]="Beijing";
printf ("%d\n", strlen (strcpy (str, "China")));
```
- A) 5          B) 7          C) 12          D) 14
57. 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) 4 B) 5 C) 9 D) 10
58. 设有如下函数定义：
- ```
int f (char *s)
{ char *p=s;
 while (*p!='\0') p++;
 return (p-s); }
```
- 如果在主程序中用下面的语句调用上述函数，则输出结果为
- ```
printf ("%d\n", f ("goodbey!"));
```
- A) 3 B) 6 C) 8 D) 0
59. 以下程序的输出结果是
- ```
sub1 (char a, char b) { char c; c=a; a=b; b=c; }
sub2 (char *a, char b) { char c; c=*a; *a=b; b=c; }
sub3 (char *a, char *b) { char c; c=*a; *a=*b; *b=c; }
main ()
{ char a, b;
 a='A'; b='B'; sub3 (&a, &b);
 putchar (a); putchar (b);
 a='A'; b='B'; sub2 (&a, b);
 putchar (a); prtchar (b);
 a='A'; b='B'; sub1 (a, b);
 putchar (a); putchar (b); }
```

A) BABBAB      B) ABBBBBA      C) BABABA      D) BAABBA

60. 下列程序中字符串中各单词之间有一个空格，则程序的输出结果是【      】。

```
#include <string.h>
main ()
{ char str1[]="How do you do", *p1=str1;
 strcpy (str1+strlen (str1) /2, "es she");
 printf ("%s \n", p1); }
```

61. 以下程序的输出结果是

A) AfghdEFG      B) Abfhd      C) Afghd      D) Afgd

```
#include <string.h>
main ()
{ char *p1 , *p2 , str[50]="ABCDEFGH";
 p1="abcd"; p2="efgh";
 strcpy (str+1, p2+1);
 strcpy (str+3, p1+3);
 printf ("%s", str); }
```

62. 下列程序的输出结果是【      】。

```
#include <string.h>
main ()
{ char b[30] , *chp ;
 strcpy (&b[0], "CH");
 strcpy (&b[1], "DEF");
 strcpy (&b[2], "ABC");
 printf ("%s \n", b); }
```

63. 下面程序的输出是

```
main ()
{ char *s="121" ;
 int k=0 , a=0 , b=0 ;
 do { k++ ;
 if (k%2==0) {a=a+s[k]-'0' ; continue ; }
 b=b+s[k]-'0' ; a=a+s[k]-'0' ; } while (s[k+1]) ;
 printf ("k=%d a=%d b=%d\n" , k , a , b) ;
```

A) k=3 a=2 b=3      B) k=3 a=3 b=2  
C) k=2 a=3 b=2      D) k=2 a=2 b=3

64. 以下程序的功能是：从键盘上输入一行字符，存入一个字符数组中，然后输出该字符串，请填空。

```
include "ctype.h"
```

```
include "stdio.h"
main ()
{ char str[81], *sptr;
 int i;
 for (i=0; i<80; i++)
 { str[i]=getchar ();
 if (str[i]=='\n') break; }
 str[i]= 【 】;
 sptr=str;
 while (*sptr) putchar (*sptr 【 】);
 }
}
```

65. 有以下程序

```
main ()
{ char a[]="programming" , b[]="language";
 char *p1 , *p2 ;
 int i;
 p1=a; p2=b;
 for (i=0; i<7; i++)
 if (* (p1+i) == * (p2+i))
 printf ("%c" ; * (p1+i)); }
```

输出结果是

- A) gm      B) rg      C) or      D) ga

66. 下面函数的功能是

```
sss (char *s , char *t)
{ while ((*s) && (*t) && (*t++== *s++)) ;
 return (*s-*t) ; }
```

- A) 求字符串的长度      B) 比较两个字符串的大小  
C) 将字符串 s 复制到字符串 t 中      D) 将字符串 s 接续到字符串 t 中

67. 下面各语句行中，不能正确进行字符串赋值操作的语句是

- A) char ST[5]={"ABCDE"};  
B) char S[5]={'A' , 'B' , 'C' , 'D' , 'E'};  
C) char \*S; S="ABCDE";  
D) char \*S; scanf ("%S", S);

68. 下列函数的功能是

```
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 Y        D) 将字符串 X 连接到字符串 Y 后面

69. 下面程序的输出是

- A) -1      B) 0      C) 1      D) 不确定的值

```
#include <string.h>
main ()
{ char *p1 = "abc", *p2 = "ABC", str[50] = "xyz";
 strcpy (str+2, strcat (p1, p2));
 printf ("%n", str); }
```

70. 下面程序的输出结果是

```
F (char *S)
{ char *P=S;
 while (*P != '\0') P++;
 return (P-S); }
main ()
{ printf ("%d\n", F ("ABCDEF")); }
```

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

71. 请读程序:

```
#include <stdio.h>
#include <string.h>
main ()
{ char *s1="ABCDEF", *s2="aB";
 s1++; s2++;
 printf ("%d\n", strcmp (s1, s2)); }
```

上面程序的输出结果是

- A) 正数    B) 负数              C) 零    D) 不确定的值

72. 请读程序

```
#include<stdio.h>
char FUN (char *C)
{ if (*C<='Z' && *C>='A') *C - ='A' - 'a';
 return *C; }
main ()
{ char S[81], *P=S;
 gets (S);
 while (*P)
 { *P=FUN (P); putchar (*P); P++; }
 putchar ('\n'); }
```

若运行时从键盘上输入 OPEN THE DOOR (表示回车), 则输出结果是



- A) oPEN tHE door                      B) open the door  
C) OPEN THE DOOR                      D) Open The Door

73. 以下程序的输出结果是

```
#include "ctype.h"
void space (char *str)
{ int i , t ;
 char ts[81];
 for (i=0, t=0 ; str[i]!='\0' ; i+=2)
 if (! isspace (* (str+i)) && (* (str+i) !='a'))
 ts[t++]=toupper (str[i]) ;
 ts[t]='\0' ; strcpy (str , ts) ; }
```

main ( )

```
{ char s[81]="abcdefg" ;
```

```
 space (s); puts (s); }
```

- A) abcdeg      B) bcde      C) ABCDE      D) BCDE

74. 请读程序:

```
#include "stdio.h"
void fun (char *s)
{ char a[10];
 strcpy (a, "STRING"); s=a; }
main ()
{ char *p=NULL;
 fun (p); printf ("%s\n", p); }
```

上面程序的输出结果是

- A) STRING    B) STRING      C) STRING    D) 不确定的值

75. 以下程序的功能是：将无符号八进制数字构成的字符串转换为十进制整数。例如，输入的字符串为：556，则输出十进制整数 366。请填空。

```
#include <stdio.h>
main ()
{ char *p, s[6];
 int n;
 p=s;
 gets (p);
 n=*p-'0';
 while (【 】 !='\0') n=n*8+*p-'0';
 printf ("%d \n", n); }
```

76. 以下程序的输出结果是

- A) ihgfedcba      B) abcdefghi      C) abcdedeba      D) ihgfefghi

```
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 ; }}
```

77. 以下程序的输出结果是

- A) 4 2 1 1      B) 0 0 0 8      C) 4 6 7 8      D) 8 8 8 8

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

78. 以下函数用来在 w 数组中插入 x，w 数组中的数已按由小到大顺序存放，n 所指存储单元中存放数组中数据的个数，插入后数组中的数仍有序。请填空。

```
void fun (char *w , char x , int *n)
{ int i , p ;
 p=0 ;
 w[*n]=x ;
 while (x<w[p]) 【 】 ;
 for (i=*n ; i>=p ; i--) w[i]=【 】 ;
 w[p]=x ; ++ *n ; }
```

79. 以下程序的输出结果是

- A) 6      B) 6789      C) '6'      D) 789

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

```
printf ("%s\n", p-3); }
```

80. 设有如下定义:

```
int (*ptr) ();
```

则以下叙述中正确的是

- A) ptr 是指向一维组数的指针变量
- B) ptr 是指向 int 型数据的指针变量
- C) ptr 是指向函数的指针 ; 该函数返回一个 int 型数据
- D) ptr 是一个函数名 ; 该函数的返回值是指向 int 型数据的指针

81. 若有函数 max (a , b), 并且已使函数指针变量 p 指向函数 max , 当调用该函数时, 正确的调用方法是

- A) (\*p) max (a , b) ;
- B) \*pmax (a , b) ;
- C) (\*p) (a , b) ;
- D) \*p (a , b) ;

82. 已有函数 max (a, b), 为了让函数指针变量 p 指向函数 max, 正确的赋值方法是

- A) p=max;
- B) \*p=max;
- C) p=max (a, b);
- D) \*p=max (a, b);

83. 已有定义 int (\*p) (); 指针 p 可以

- A) 代表函数的返回值
- B) 指向函数的入口地址
- C) 表示函数的类型
- D) 表示函数返回值的类型

84. 若有以下说明和定义

```
fun (int *c) { }
main ()
{ int (*a) () =fun, *b (), w[10], c;
 ...
}
```

在必要的赋值之后, 对 fun 函数的正确调用语句是

- A) a=a (w);
- B) (\*a) (&c);
- C) b=\*b (w);
- D) fun (b);

85. 下面程序的输出结果是 【                      】。

```
funa (int a, int b) { return a+b; }
funb (int a, int b) { return a-b; }
sub (int (*t) (), int x, int y)
{ return ((*t) (x, y)); }
main ()
{ int x , (*p) ();
 p=funa;
 x=sub (p, 9, 3);
 x+=sub (funb, 8, 3);
 print ("%d\n", x); }
```

86. 若有以下的说明和语句:

```
main ()
```

则以一选项中能正确表示 t 数组元素地址的表达式是

87. 若有以下的说明和语句, 则在执行 for 语句后,  $*(*(pt+1)+2)$  表示的数组元素是

```
int t[3][3], *pt[3], k;
for (k=0; k<3; k++) pt[k]=&t[k][0]; ...
```

```
main ()
{ char ch[2][5]={ "6937", "8254" }, *p[2];
 int i, j, s=0;
 for (i=0; i<2; i++) p[i]=ch[i];
 for (i=0; i<2; i++)
 for (j=0; p[i][j]>' \0'; j+=2) s=10*s+p[i][j]- '0';
 printf ("%d\n", s); }
```

A) 69825      B) 63825      C) 6385      D) 693825

```
int **pp , *p , a=10 , b=20 ;
pp=&p ; p=&a ; p=&b ; printf ("%d , %d\n" , *p , **pp) ;
```

输出结果是:

90. 下面程序的输出结果是 【           】

91. 若有下面的程序片段:

则对数组元素的错误引用是

92. 以下程序的输出结果是

74

```

 { **s=p[1][1]; }
main ()
{ int a[2][3]={1, 3, 5, 7, 9, 11}, *p;
 p= (int*) malloc (sizeof (int));
 fut (&p, a);
 printf ("%d\n", *p); }
A) 1 B) 7 C) 9 D) 11

```

93. 下列程序的输出结果是

```

main ()
{ int a[5]={2, 4, 6, 8, 10}, *p, * *k;
 p=a; k=&p;
 printf ("%d", * (p++));
 printf ("%d\n", * *k); }
A) 4 4 B) 2 2 C) 2 4 D) 4 6

```

94. 设有如下程序:

```

#include<stdio.h>
main ()
{ int **k , *j , i=100 ;
 j=&i ; k=&j ;
 printf ("%d\n" , **k) ; }

```

上面程序的输出结果是

A) 运行错误      B) 100      C) i 的地址      D) j 的地址

95. 设有如下程序:

```

include<stdio.h>
main (int argc , char *argv[])
{ while (--argc>0)
 printf ("%s" , argv[argc]) ;
 printf ("\n") ; }

```

假定上述程序经编译和连接后生成可执行文件 PROG. EXE , 如果在 DOS 提示符下键入 PROG ABCD EFGH IJKL <CR> , 则输出结果为【                      】。

96. 若以下程序所生成的可执行文件名 FILE1. EXE; 当打入以下命令执行该程序时:

FILE1 CHINA BEIJING SHANGHAI 程序的输出结果是

A) CHINA BEIJING SHANGHAI                      B) C B S  
C) FILE1 CHINA BEIJING                      D) F C B

```

main (int argc, char *argv[])
{ while (argc-->0)
 { ++argv; printf ("%s", *argv); }}

```

97. 请读程序:

若上面的程序编译`连接后生成可执行文件 ABC.EXE，则输入以下命令行  
abc file1 file2<CR> (表示回车)

98. 设有如下定义:

则以下说法中正确的是

- 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着含有 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‘a’和‘A’的地址

A) C 语言允许 main 函数带形数，且形参个数和形参名均可由用户指定

B) C 语言允许 main 函数带形参，形参名只能是 argc 和 argv

C) 当 main 函数带有形参时，传给形参的值只能从命令行中得到

D) 有说明:main (int argc, char \*argv), 则形参 argc 的值必须大于 1

A) `i=*p;`      B) `p*=&j;`      C) `i=&j;`      D) `i=**p;`

```
main()
{
 int a, b, k=4, m=6, *p1=&k, *p2=&m;
 a=p1==&m;
 b=(*p1) / (*p2)+7;
 printf("a=%d\n", a);
 printf("b=%d\n", b);
}
```

102. 以下程序运行后的输出结果是 【                      】。

103. 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);

104. 若定义: int a=511, \*b=&a;, 则 printf ("%d\n", \*b); 的输出结果为

- A) 无确定值      B) a 的地址      C) 512      D) 511

105. 以下程序的输出结果是

```
char cchar(char ch)
{ if (ch>= 'A' &&ch<= 'Z') ch=ch- 'A'+ 'a';
 return ch;
}
main()
{ char s[]="ABC+abc=defDEF", *p=s;
 while(*p)
 { *p=cchar(*p);
 p++;
 }
 printf ("%s\n", s);
}
A) abc+ABC=DEFdef B) abc+abc=defdef
C) abcaABCDEFdef D) abcabcdefdef
```

106. 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

107. 若有以下定义和语句:

```
int s[4][5], (*ps)[5];
ps=s;
```

则对 s 数组元素的正确引用形式是

- A) ps+1                      B) \*(ps+3)                      C) ps[0][2]                      D) \*(ps+1)+3

108. 以下程序的输出结果是

```
#include <stdio.h>
#include <string.h>
main()
{ char b1[8]="abcdefg", b2[8], *pb=b1+3;
 while (--pb>=b1) strcpy(b2, pb);
 printf("%d\n", strlen(b2));
}
```

- A) 8                      B) 3                      C) 1                      D) 7

109. 在说明语句: int \*f(); 中, 标识符 f 代表的是

- A) 一个用于指向整型数据的指针变量  
B) 一个用于指向一维数组的行指针  
C) 一个用于指向函数的指针变量  
D) 一个返回值为指针型的函数名

110. 不合法的 main 函数命令行参数表示形式是

- A) main(int a, char \*c[])  
B) main(int arc, char \*\*arv)  
C) main(int argc, char \*argv)  
D) main(int argv, char \*argc[])

111. 以下程序的输出结果是【                      】。

```
main()
{ char *p="abcdefgh", *r;
 long *q;
 q=(long*)p;
 q++;
 r=(char*)q;
 printf("%s\n", r);
}
```

112. 以下程序中, select 函数的功能是: 在 N 行 M 列的二维数组中, 选出一个最大值作为函数值返回, 并通过形参传回此最大值所在的行下标。请填空。

```
#define N 3
#define M 3
select(int a[N][M], int *n)
{ int i, j, row=1, colum=1;
```



```

 for (i=0; i<N; i++)
 for (j=0; j<M; j++)
 if (a[i][j]>a[row][column]) {row=i; column=j;}
 *n= 【 】;
 return 【 】;
}
main()
{ int a[N][M]={9, 11, 23, 6, 1, 15, 9, 17, 20}, max, n;
 max=select(a, &n);
 printf("max=%d, line=%d\n", max, n);
}

```

113. mystrlen 函数的功能是计算 str 所指字符串的长度, 并作为函数值返回。请填空。

```

int mystrlen(char *str)
{ int i;
 for (i=0; 【 】 != '\0'; i++);
 return(【 】);
}

```

114. 有以下程序

```

#include <stdlib.h>
main()
{ char *p, *q;
 p=(char*)malloc(sizeof(char)*20); q=p;
 scanf("%s%s", p, q); printf("%s%s\n", p, q);
}

```

若从键盘输入: abc def<回车>, 则输出结果是:

A) def def            B) abc def            C) abc d            D) d d

115. 设有定义: int n, \*k=&n; 以下语句将利用指针变量 k 读写变量 n 中的内容, 请将语句补充完整。

```

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

```

116. 有以下程序

```

#include <string.h>
main(int argc, char *argv[])
{ int i, len=0;
 for (i=1; i<argc; i++) len+=strlen(argv[i]);
 printf("%d\n", len);
}

```

}

程序编译连接后生成的可执行文件是 `ex1.exe`，若运行时输入带参数的命令行是：

`ex1 abcd efg 10`<回车>

则运行的结果是：

- A) 22                      B) 17                      C) 12                      D) 9

117. 有以下程序

```
int fa(int x)
{ return x*x; }
int fb(int x)
{ return x*x*x; }
int f(int (*f1)(), int (*f2)(), int x)
{ return f2(x)-f1(x); }
main()
{ int i;
 i=f(fa, fb, 2); printf("%d\n", i);
}
```

程序运行后的输出结果是

- A) -4                      B) 1                      C) 4                      D) 8

## 第十一章 结构体与共用体

1. 说明一个结构体变量时系统分配给它的内存是

- A) 各成员所需要内存量的总和
- B) 结构体中第一个成员所需内存量
- C) 成员中占内存量最大者所需的容量
- D) 结构中最后一个成员所需内存量

2. C 语言结构体类型变量在程序执行期间

- A) 所有成员一直驻留在内存中
- B) 只有一个成员驻留在内存中
- C) 部分成员驻留在内存中
- D) 没有成员驻留在内存中

3. 设有以下说明语句

```
struct stu { int a ; float b ; } stutype ;
```

则下面的叙述不正确的是

- A) struct 是结构体类型的关键字
- B) struct stu 是用户定义的结构体类型
- C) stutype 是用户定义的结构体类型名
- D) a 和 b 都是结构体成员名

4. 设有以下结构类型说明和变量定义, 则变量 a 在内存所占字节数是【           】。

```
struct stud { char num[6]; int s[4]; double ave; } a, *p;
```

5. 程序中有下面的说明和定义

```
struct abc { int x; char y; }
struct abc s1, s2;
```

则会发生的情况是

- A) 编译出错
- B) 程序将顺利编译、连接、执行
- C) 能顺利通过编译、连接、但不能执行
- D) 能顺利通过编译、但连接出错

6. 有如下定义

```
struct person { char name[9]; int age; };
struct person class[10]={ "Johu", 17, "Paul", 19, "Mary", 18, "Adam",
16};
```

根据上述定义, 能输出字母 M 的语句是

- A) printf (" %c\n", class[3].name);

- B) printf (" %c\n", class[3].name[1]);
- C) printf (" %c\n", class[2].name[1]);
- D) printf (" %c\n", class[2].name[0]);

7. 下面程序的输出是

- A) 0      B) 1      C) 3      D) 6

```
main ()
{ struct cmplx { int x; int y; } cnum[2]={1, 3, 2, 7};
 printf ("%d\n", cnum[0].y/cnum[0].x*cnum[1].x); }
```

8. 下列程序的输出结果是

- A) 5      B) 6      C) 7      D) 8

```
struct abc { int a, b, c; };
main ()
{ struct abc s[2]={{1, 2, 3}, {4, 5, 6}};
 int t;
 t=s[0].a+s[1].b;
 printf ("%d \n", t); }
```

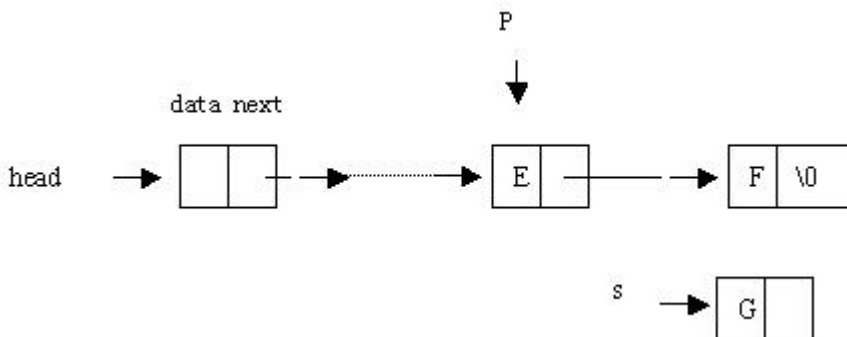
9. 有以下程序

```
struct stu
{ int num; char name[10]; int age; };
void fun (struct stu *p)
{ printf ("%s\n", (*p).name); }
main ()
{ struct stu students[3]={{9801, "Zhang", 20} , {9802, "Wang", 19} ,
 {9803, "Zhao", 18} };
 fun (students+2); }
```

输出结果是

- A) Zhang      B) Zhao      C) Wang      D) 18

10. 若已建立如下图所示的单向链表结构，



在该链表结构中，指针 p、s 分别指向图中所示结点，则不能将 s 所指的结点插入



```

 { p=【 】 malloc (sizeof (struct list));
 p->data=ch;
 p->next=p; q=p;
 ch=getchar (); }
 p->next='\0';
 }
 }

```

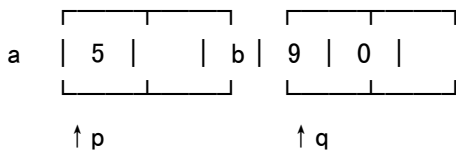
15. 有以下结构体说明和变量的定义，且如下图所示指针 p 指向变量 a，指针 q 指向变量 b。则不能把结点 b 连接到结点 a 之后的语句是

A) a.next=q;                      B) p.next=&b;  
C) p->next=&b;                      D) (\*p).next=q;

```

struct node { char data; struct node *next; } a, b, *p=&a, *q=&b;
data next data next

```



16. 当说明一个共用体变量时系统分配给它的内存是
- A) 各成员所需要内存量的总和  
B) 共用体中第一个成员所需内存量  
C) 成员中占内存量最大者所需的容量  
D) 共用体中最后一个成员所需内存量
17. 以下对 C 语言中共用体类型数据的叙述正确的是
- A) 可以对共用体变量名直接赋值  
B) 一个共用体变量中可以同时存放其所有成员  
C) 一个共用体变量中不可以同时存放其所有成员  
D) 共用体类型定义中不能出现结构体类型的成员
18. C 语言共用体类型变量在程序运行期间
- A) 所有成员一直驻留在内存中  
B) 只有一个成员驻留在内存中  
C) 部分成员驻留在内存中  
D) 没有成员驻留在内存中
19. 字符 '0' 的 ASCII 码的十进制数为 48，且数组的第 0 个元素在低位，则以下程序的输出结果是

```

#include <stdio.h>
main ()
{ union { int i[2]; long k; char c[4]; } r, *s=&r;
 s->i[0]=0x39; s->i[1]=0x38;
 printf ("%c\n", s->c[0]) }

```

A) 39    B) 9    C) 38    D) 8

20. 若有以下说明和定义语句，则变量 w 在内存中所占的字节数是【           】。

```
union aa { float x, y ; char c[6]; };
struct st { union aa v; float b[5]; double ave; } w;
```

21. 设有以下定义：

```
#include "alloc.h"
typedef union { long i ; int k[5] ; char c ; }DATE ;
struct date
{ int cat ; DATE cow ; double dog ; }too ;
DATE max ;
```

则下列语句的执行结果是

```
printf ("%d", sizeof (struct date)+sizeof (max));
```

A) 25    B) 30    C) 18    D) 8

22. 已知字符 0 的 ASCII 码为十六进制的 30，下面程序的输出是

A) 6    B) 9    C) 0    D) 3

```
main ()
{ union { unsigned char c; unsigned int i[4]; }z;
 z.i[0]=0x39; z.i[1]=0x36;
 printf ("%c\n", z.c); }
```

23. 变量 a 所占内存字节数是

A) 4    B) 5    C) 6    D) 8

```
union U
{ char st[4] ; int i ; long l ; } ;
struct A { int c; union U u; } a ;
```

24. 以下对枚举类型名的定义中正确的是

A) enum a={one , two , three} ;  
B) enum a {one=9 , two=-1 , three} ;  
C) enum a={"one" , "two" , "three"} ;  
D) enum a {"one" , "two" , "three"} ;

25. 设有如下枚举类型定义：

```
enum language { Basic=3 , Assembly , Ada=100 , COBOL , Fortran} ;
```

枚举量 Fortran 的值为

A) 4    B) 7    C) 102    D) 103

26. 下面程序的输出是

A) 0 1 2 3    B) 0 4 0 10    C) 0 4 5 15    D) 1 4 5 15

```
main ()
{ enum team { my , your=4 , his , her=his+10};
```

```
printf ("%d%d%d%d\n", my, your, his, her); }
```

27. 以下选项中，能定义 s 为合法的结构体变量的是

A) typedef struct abc

```
{ double a;
 char b[10];
} s;
```

B) struct

```
{ double a;
 char b[10];
} s;
```

C) struct ABC

```
{ double a ;
 char b[10] ;
} ABC s ;
```

D) typedef ABC

```
{ double a;
 char b[10] ;
} ABC s ;
```

28. 下面对的叙述中不正确的是

A) 用 typedef 可以定义各种类型名，但不能用来定义变量

B) 用 typedef 可以增加新类型

C) 用 typedef 只是将已存在的类型用一个新的标识符来代表

D) 使用 typedef 有利于程序的通用和移植

29. 下面程序的输出是

A) 32      B) 16      C) 8      D) 24

```
typedef union { long x[2]; int y[4]; char z[8]; }MYTYPE;
MYTYPE them;
main ()
{ printf ("%d\n", sizeof (them)); }
```

30. 以下各选项企图说明一种新的类型名，其中正确的是

A) typedef v1 int;

B) typedef v2=int;

C) typedefv1 int v3;

D) typedef v4: int;

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

A) 12

B) 23

C) 14

D) 32

32. 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

33. 以下程序段用于构成一个简单的单向链表，请填空。

```

struct STRU
{ int x, y ;
float rate;
【 】 p;
} a, b;
a.x=0; a.y=0; a.rate=0; a.p=&b;
b.x=0; b.y=0; b.rate=0; b.p=NULL;

```

34. 若有如下结构体说明：

```

struct STRU
{ int a, b ; char c; double d;
 struct STRU p1,p2;
};

```

请填空，以完成对 t 数组的定义，t 数组的每个元素为该结构体类型

【                      】 t[20];

35. 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));

36. 若有下面的说明和定义：

```

struct test
{ int m1; char m2; float m3;
 union uu {char u1[5]; int u2[2];} ua;
} myaa;

```

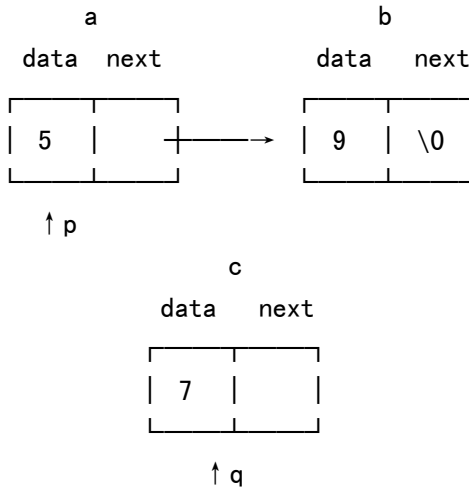
则 sizeof(struct test )的值是

A) 12                      B) 16                      C) 14                      D) 9

37. 若以下定义:

```
struct link
{
 int data;
 struct link *next;
} a, b, c, *p, *q;
```

且变量 a 和 b 之间已有如下图所示的链表结构:



指针 p 指向变量 a, q 指向变量 c。则能够把 c 插入到 a 和 b 之间并形成新的链表的语句组是:

- A) a.next=c; c.next=b; B) p.next=q; q.next=p.next;  
C) p->next=&c; q->next=p->next; D) (\*p).next=q; (\*q).next=&b;

38. 设有以下说明语句

```
typedef struct
{
 int n;
 char ch[8];
} PER;
```

则下面叙述中正确的是

- A) PER 是结构体变量名 B) PER 是结构体类型名  
C) typedef struct 是结构体类型 D) struct 是结构体类型名

39. 以下定义的结构体类型拟包含两个成员, 其中成员变量 info 用来存入整形数据; 成员变量 link 是指向自身结构体的指针. 请将定义补充完整。

```
struct node
{
 int info;
 【 】 link;
}
```

40. 有以下程序

```

struct STU
{ char num[10]; float score[3]; };
main()
{ struct stu s[3]={ { "20021", 90, 95, 85}, { "20022", 95, 80, 75},
{ "20023", 100, 95, 90}}, *p=s;
 int i; float sum=0;
 for(i=0; i<3, i++)
 sum=sum+p->score[i];
 printf("%6.2f\n", sum);
}

```

程序运行后的输出结果是

- A) 260.00      B) 270.00      C) 280.00      D) 285.00

41. 设有如下定义:

```

struct sk
{ int a;
 float b;
}data;
int *p;

```

若要使 P 指向 data 中的 a 域, 正确的赋值语句是

- A) p=&a;      B) p=data.a;      C) p=&data.a;      D) \*p=data.a;

42. 有以下程序

```

#include <stdlib.h>
struct NODE
{ int num; struct NODE *next; };
main()
{ struct NODE *p, *q, *r;
 p=(struct NODE*)malloc(sizeof(struct NODE));
 q=(struct NODE*)malloc(sizeof(struct NODE));
 r=(struct NODE*)malloc(sizeof(struct NODE));
 p->num=10; q->num=20; r->num=30;
 p->next=q; q->next=r;
 printf("%d\n", p->num+q->next->num);
}

```

程序运行后的输出结果是

- A) 10      B) 20      C) 30      D) 40

43. 若有以下说明和定义

```

typedef int *INTEGER;

```

INTEGER p, \*q;

以下叙述正确的是

- A) P 是 int 型变量
- B) p 是基类型为 int 的指针变量
- C) q 是基类型为 int 的指针变量
- D) 程序中可用 INTEGER 代替 int 类型名

44. 下面程序的运行结果是：\_\_\_\_\_。

```
typedef union student
{
 char name[10];
 long sno;
 char sex;
 float score[4];
} STU;
main()
{
 STU a[5];
 printf("%d\n", sizeof(a));
}
```

## 第十二章 位运算

1. 语句: `printf ("%d \n", 12 &012);` 的输出结果是  
A) 12      B) 8      C) 6      D) 012
2. 设 `int b=2;` 表达式 `(b>>2) / (b>>1)` 的值是  
A) 0      B) 2      C) 4      D) 8
3. 执行下面的程序段  

```
int x=35, b;
char z='A';
b=((x&15) && (z<'a'));
```

后, `b` 的值为  
A) 0      B) 1      C) 2      D) 3
4. 设二进制数 `a` 是 00101101, 若想通过异或运算 `a^b` 使 `a` 的高 4 位取反, 低 4 位不变, 则二进制数 `b` 应是【                  】。
5. 下面程序的输出是  
A) 100      B) 160      C) 120      D) 64  

```
main ()
{ char x=040;
printf ("%d\n", x=x<<1); }
```
6. 设有如下定义: `int x=1, y=-1;` 则语句: `printf ("%d\n", (x--&++y));` 的输出结果是  
A) 1      B) 0      C) -1      D) 2
7. 在位运算中, 操作数每右移一位, 其结果相当于  
A) 操作数乘以 2      B) 操作数除以 2  
C) 操作数除以 4      D) 操作数乘以 4
8. 在位运算中, 操作数每左移一位, 其结果相当于  
A) 操作数乘以 2      B) 操作数除以 2  
C) 操作数除以 4      D) 操作数乘以 4
9. 设位段的空间分配由右到左, 则以下程序的运行结果是  

```
struct bit
{ unsigned a: 2;
 unsigned b: 3;
 unsigned c: 4;
 int i;
} date;
```

```
main ()
{
 date.a=8; date.b=2;
 printf ("%d\n", date.a+date.b);
}
```

A) 语法错误      B) 2      C) 5      D) 10

10. 设位段的空间分配由右到左，则以下程序的运行结果是【      】。

```
struct bit
{
 unsigned a: 2 ;
 unsigned b: 3 ;
 unsigned c: 4 ;
 int i ;
} date ;

main ()
{
 date.a=1; date.b=2; date.c=3; i=0;
 printf ("%d\n", date);
}
```

11. 以下程序的输出结果是

```
main()
{
 char x=040;
 printf ("%0\n", x<<1);
}
```

A) 100      B) 80      C) 64      D) 32

12. 有以下程序

```
main()
{
 unsigned char a,b,c;
 a=0x3; b=a|0x8; c=b<<1;
 printf("%d%d\n", b,c);
}
```

程序运行后的输出结果是

A) - 11    12      B) - 6    - 13      C) 12    24      D) 11    22

## 第十三章 文 件

1. 系统的标准输入文件是  
A) 键盘      B) 显示器      C) 软盘      D) 硬盘
2. 以下可作为函数 `fopen` 中第一个参数的正确格式是  
A) `c:\user\text.txt`      B) `c:\user\text.txt`  
C) `"c:\user\text.txt"`      D) `"c:\\user\\text.txt"`
3. 若要用 `fopen` 函数打开一个新的二进制文件, 该文件要既能读也能写, 则文件方式字符串应是  
A) `"ab+"`      B) `"wb+"`      C) `"rb+"`      D) `"ab"`
4. 若用 `"a+"` 方式打开一个已存在的文件, 则以下叙述正确的是  
A) 文件打开时, 原有文件内容不被删除, 位置指针移到文件末尾, 可作添加和读操作  
B) 文件打开时, 原有文件内容不被删除, 位置指针移到文件开头, 可作重写和读操作  
C) 文件打开时, 原有文件内容被删除, 只可作写操作  
D) 以上各种说法皆不正确
5. 当顺利执行了文件关闭操作时, `fclose` 函数的返回值是  
A) -1      B) TRUE      C) 0      D) 1
6. 已知函数的调用形式: `fread (bufferb, size, countb, fp)`; 其中 `buffer` 代表的是  
A) 一个整型变量, 代表要读入的数据项总数  
B) 一个文件指针, 指向要读的文件  
C) 一个指针, 指向要读入数据的存放地址  
D) 一个存储区, 存放要读的数据项
7. `fscan` 函数的正确调用形式是  
A) `fscan (fp, 格式字符串, 输出表列);`  
B) `fscan (格式字符串, 输出表列, fp)`  
C) `fscan (格式字符串, 文件指针, 输出表列);`  
D) `fscan (文件指针, 格式字符串, 输出表列);`
8. `fgetc` 函数的作用是从指定文件读入一个字符, 该文件的打开方式必须是  
A) 只写      B) 追加      C) 读或读写      D) 答案 B 和 C 都正确
9. 若调用 `fputc` 函数输出字符成功, 则其返回值是  
A) EOF      B) 1      C) 0      D) 输出的字符
10. 若要用下面的程序片段使指针变量 `P` 指向一个存储整型变量的动态存储单元:

```
int *P;
P = _____ malloc (sizeof (int));
```

则应填入

A) int    B) int \*    C) ( \* int )    D) (int \*)

11. 下面的程序调用 getone 函数开辟一个动态存储单元，调用 assone 函数把数据输入此动态存储单元，调用 outone 函数输出此动态存储单元中的数据。请填空。

```
include <stdio.h>
getone (int **s)
{ *s= (int *) malloc (sizeof (int)) ; }
assone (int *a)
{ scanf ("%d", 【 】) ; }
outone (int *b)
{ printf ("%d\n", 【 】) ; }
main ()
{ int *p ;
 getone (&p) ; assone (p) ; outone (p) ; }
```

12. 请读程序:

```
void fun (float *p1 , float *p2 , float *s)
{ s= (float *) malloc (1, sizeof (float));
 *s= *p1 + * (p2++) ; }
main ()
{ float a[2]={1.1, 2.2}, b[2]={10.0, 20.0}, *s=a;
 fun (a, b, s);
 printf ("%f\n", *s); }
```

上面程序的输出结果是

A) 11.100000    B) 12.100000    C) 21.100000    D) 1.100000

13. 若要使指针 p 指向一个 double 类型的动态存储单元，请填空。

p=【                      】 malloc (sizeof (double));

14. 若 fp 是指向某文件的指针，且已读到该文件的末尾，则 C 语言函数 feof ( fp) 的返回是

A) EOF    B) -1    C) 非零值    D) NULL

15. 下面的程序用来统计文件中字符的个数，请填空。

```
main ()
{ FILE *fp;
 long num=0;
 if ((fp=fopen ("fname.dat", "r")) == 【 】)
 { printf ("Can't open file! \n"); exit (0); }
 while 【 】
```



```

 { fgetc (fp); num++; }
printf ("num=%d\n", num);
【 】}

```

16. 以下程序的功能是：从键盘上输入一个字符串，把该字符串中的小写字母转换为大写字母，输出到文件 test.txt 中，然后从该文件读出字符串并显示出来。请填空。

```

main ()
{ FILE *fp;
 char str[100] ; int i=0 ;
 if ((fp=fopen ("text.txt", 【 】)) ==NULL)
 { printf ("can't open this file.\n"); exit (0); }
 printf ("input astring: \n"); gest (str);
 while (str[i])
 { if (str[i]>='a' &&str[i]<='z')
 str[i]= 【 】;
 fputc (str[i], fp); i++; }
 fclose (fp);
 fp=fopen ("test.txt", 【 】) ;
 fgets (str, 100, fp) ;
 printf ("%s\n", str) ;
 fclose (fp); }

```

17. 设有如下程序：

```

include <stdio.h>
main (int argc , char *argv[])
{ FILE *fp ;
 void fc () ;
 int i=1 ;
 while (--argc>0)
 if ((fp=fopen (argv[i++] , "r")) ==NULL)
 { printf ("Cannot open file! \n") ; exit (1) ; }
 else { fc (fp) ; fclose (fp) ; } }
void fc (FILE *ifp)
{ char c ;
 while ((c=getc (ifp)) !=' #') putchar (c-32) ; }

```

上述程序经编译、连接后生成可执行文名为 cpy.exe。假定磁盘上有三个文本文件，其文件名和内容分别为：

| 文件名 | 内容    |
|-----|-------|
| a   | aaaa# |
| b   | bbbb# |

```
c cccc#
```

如果在 DOS 下键入

```
cpy a b c<CR>
```

则程序输出【 】。

18. 在 C 程序中, 可把整型数以二进制形式存放到文件中的函数是

- A) fprintf 函数                      B) fread 函数  
C) fwrite 函数                        D) fputc 函数

19. 假定下列程序的可执行文件名为 prg.exe, 则在该程序所在的子目录下输入命令行:

prg hello good<回车>后, 程序的输出结果是

- A) hello good      B) hg      C) hel      D) hellogood

```
main () (int argc, char *argv[])
{ int i;
 if (argc<=0) return;
 for (i=1; i<argc; i++) printf ("%c", *argv[i]); }
```

20. 以下函数 creatlist 用来建立一个带头节点的单链表, 链表的结构如下图所示, 新的结点总是插入在链表的末尾。链表的头指针作为函数值返回, 链表最后一个节点的 next 域放入 NULL, 作为链表结束标志。data 为字符型数据域, next 为指针域。读入时字符以#表示输入结束 (#不存入链表)。请填空。

```
struct node
{ char data;
 struct node * next; };
...
```

```
【 】 creatlist ()
{ struct node * h, * s, * r; char ch;
 h= (struct node *) malloc (sizeof (struct node));
 r=h;
 ch=getchar ();
 { s= (struct node *) malloc (sizeof (struct node));
 s-data=【 】;
 r-next=s; r=s;
 ch=getchar (); }
 r->next=【 】 ;
 return h; }
```

21. 标准函数 fgets (s , n , f) 的功能是

- A) 从文件 f 中读取长度为 n 的字符串存入指针 s 所指的内存  
B) 从文件 f 中读取长度不超过 n-1 的字符串存入指针 s 所指的内存  
C) 从文件 f 中读取 n 个字符串存入指针 s 所指的内存  
D) 从文件 f 中读取长度为 n-1 的字符串存入指针 s 所指的内存

22. 若有如下图所示五个连续的 int 类型的存储单元并赋值如下图，a[0]的地址小于a[4]的地址。p 和 s 是基类型为 int 的指针变量。请对以下问题进行填空。

a[0] a[1] a[2] a[3] a[4]

|    |    |    |    |    |
|----|----|----|----|----|
|    |    |    |    |    |
| 22 | 33 | 44 | 55 | 66 |

- (1) 若 p 已指向存储单元 a[1]。通过指针 p，给 s 赋值，使 s 指向最后一个存储单元 a[4]的语句是 【                      】。
- (2) 若指针 s 指向存储单元 a[2]，p 指向存储单元 a[0]，表达式 s-p 的值是 【                      】。
23. 以下程序运行后，输出结果是  
 A) ABCDEFG      B) CDG      C) abcdefgh      D) abCDefGh
- ```
main ( )
{ static char a[ ]="ABCDEFGH", b[ ]="abCDefGh";
  char *p1, *p2;
  int k ;
  p1=a; p2=b;
  for (k=0; k<8; k++)
    if (* (p1+k) ==* (p2+k))
      printf ("%c", * (p1+k));
  printf ("\n"); }
```
24. 在对文件进行操作的过程中，若要求文件的位置回到文件的开头，应当调用的函数是 【 】函数。
25. 下面的程序执行后，文件 testt.t 中的内容是
- ```
#include <stdio.h>
void fun(char *fname, char *st)
{ FILE *myf; int i;
 myf=fopen(fname, "w");
 for(i=0; i<strlen(st); i++) fputc(st[i], myf);
 fclose(myf);
}
main()
{ fun("test", "new world"; fun("test", "hello, " 0);
 A) hello, B) new worldhello, C) new world D) hello, rld
```
26. 以下程序段打开文件后，先利用 fseek 函数将文件位置指针定位在文件末尾，然后调用 ftell 函数返回当前文件位置指针的具体位置，从而确定文件长度，请填空。

```
FILE *myf; long fl;
myf= 【 】 ("test.t", "rb");
fseek(myf, 0, SEEK_END); fl=ftell(myf);
fclose(myf);
printf("%d\n", fl);
```

27. 若要打开 A 盘上 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\abc.txt", "r")  
 B) fopen("A:\\user\\abc.txt", "r+")  
 C) fopen("A:\user\abc.txt", "rb")  
 D) fopen("A:\\user\\abc.txt", "w")

28. 以下程序用来统计文件中字符个数。请填空。

```
#include "stdio.h"
main()
{ FILE *fp; long num=0L;
 if((fp=fopen("fname.dat", "r"))==NULL)
 { printf("Open error\n"); exit(0);}
 while(【 】)
 { fgetc(fp); num++;}
 printf("num=%ld\n", num-1);
 fclose(fp);
}
```

29. 若 fp 已正确定义为一个文件指针，d1.dat 为二进制文件，请填空，以便为“读”而打开此文件：fp=fopen( \_\_\_\_\_ );。

30. 有以下程序

```
#include <stdio.h>
main()
{ FILE *fp; int i=20, j=30, k, n;
 fp=fopen("d1.dat" "w");
 fprintf(fp, "%d\n", i); fprintf(fp, "%d\n", j);
 fclose(fp);
 fp=fopen("d1.dat", "r");
 fp=fscanf(fp, "%d%d", &k, &n); printf("%d%d\n", k, n);
 fclose(fp);
}
```

程序运行后的输出结果是

- A) 20 30                      B) 20 50                      C) 30 50                      D) 30 20

31. 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) 在利用 `fread` 函数从二进制文件中读数据时，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

# 全国计算机等级考试二级笔试试卷





## 基础知识

(考试时间 120 分钟, 满分 100 分)

2003 年 4 月

### 一、 选择题 ((1—40) 每小题 1 分, (41—50) 每小题 2 分, 共 60 分)

下列各题 A)、B)、C)、D) 四个选项中, 只有一个选项是正确的, 请将正确选项涂写在答题卡相应位置上, 答在试卷上不得分。

- (1) 用 8 位无符号二进制数能表示的最大十进制数为  
A) 127                      B) 128                      C) 255                      D) 256
- (2) 在 64 位高档微机中, 一个字长所占的二进制位数为  
A) 8                          B) 16                          C) 32                          D) 64
- (3) 在 Windows 环境下, 为了复制一个对象, 在用鼠标拖动该对象时应同时按住  
A) Alt 键                      B) Esc 键                      C) Shift 键                      D) Ctrl 键
- (4) 在 Windows 菜单中, 暗淡 (灰色) 的命令项表示该命令  
A) 暂时不能用                      B) 正在执行                      C) 包含下一层菜单                      D) 包含对话框
- (5) 在 DOS 环境下, 为了得到 TYPE 命令的帮助信息, 正确的 DOS 命令为  
A) TYPE                      B) TYPE/H                      C) TYPE/\*                      D) TYPE/?
- (6) 下列选项中, 能作为合法 DOS 文件名的是  
A) ANP/A.DAT                      B) ABCCOM                      C) ECD.BASIC                      D) XY+Z.TXT
- (7) 下列叙述中正确的是  
A) 在 Windows 环境下, 最大化的窗口是不能移动的  
B) 在 Windows 环境下, 应用程序窗口最小化后, 该应用程序暂停执行  
C) 在 Windows 环境下, 只有最小化的窗口才能关闭  
D) 在 Windows 环境下, 不能关闭最大化的窗口
- (8) 在 Windows 环境下, 工具栏中的复制按钮是  
A)                       B)                       C)                       D) 
- (9) 在 Windows98 环境下, 若要将整个屏幕上显示的内容存入剪贴板, 可以按  
A) Ctrl+PrintScreen 键                      B) Alt+PrintScreen 键

- C) Shift+PrintScreen 键 D) PrintScreen 键
- (10) 目前, 计算机病毒扩散最快的途径是
- A) 通过软件复制 B) 通过网络传播  
C) 通过磁盘拷贝 D) 运行游戏软件
- (11) 以下叙述中正确的是
- A) C 语言比其他语言高级  
B) C 语言可以不用编译就能被计算机识别执行  
C) C 语言以接近英语国家的自然语言和数学语言作为语言的表达形式  
D) C 语言出现的最晚, 具有其他语言的一切优点
- (12) C 语言中用于结构化程序设计的三种基本结构是
- 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
- (13) 在一个 C 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
- (14) 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
- (15) 下列关于 C 语言用户标识符的叙述中正确的是
- A) 用户标识符中可以出现在下划线和中划线(减号)  
B) 用户标识符中不可以出现中划线, 但可以出现下划线  
C) 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  
D) 用户标识符中可以出现在下划线和数字, 它们都可以放在用户标识符的开头
- (16) 若有以下程序段 (n 所赋的是八进制数)
- ```
int m=32767, n=032767  
printf("%d, %o/n", m, n);
```
- 执行后输出结果是
- A) 32767, 32767 B) 32767, 032767
C) 32767, 77777 D) 32767, 077777
- (17) 下列关于单目运算符++、--的叙述中正确的是
- A) 它们的运算对象可以是任何变量和常量
B) 它们的运算对象可以是 char 型变量和 int 型变量, 但不能是 float 型变量
C) 它们的运算对象可以是 int 型变量, 但不能是 double 型变量和 float 型变量

D) 它们的运算对象可以是 char 型变量、int 型变量和 float 型变量

(18) 若有以下程序段

```
int m=0xabc, n=0xabc;
m-=n;
printf("%X\n", m);
```

执行后输出结果是

- A) 0X0 B) 0x0 C) 0 D) 0XABC

(19) 有以下程序段

```
int m=0, n=0;    char c= 'a' ;
scanf ("%d%c%d", &m, &c, &n);
printf ("%d, %c, %d\n", m, c, n);
```

若从键盘上输入: 10A10<回车>, 则输出结果是:

- A) 10, A, 10 B) 10, a, 10 C) 10, a, 0 D) 10, A, 0

(20) 有以下程序

```
main()
{    int i;
   for (i=0; i<3; i++)
       switch(i)
       {    case    1:    printf("%d", i);
           case    2:    printf("%d", i);
           default:    printf("%d", i);
       }
}
```

执行后输出结果是

- A) 011122 B) 012 C) 012020 D) 120

(21) 有以下程序

```
main()
{    int    i=1, j=1, k=2;
   if ((j++ || k++) && i++) printf ("%d, %d, %d\n", i, j, k);
}
```

执行后输出结果是

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

(22) 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)
}

```

执行后输出结果是

- A) 2 B) 3 C) 4 D) 编译时有错，无结果

(23) 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);
}

```

执行后输出结果是

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

(24) 有以下程序

```

main()
{
    int    i=0, s=0;
    do{
        if(i%2) {i++;continue;}
        i++;
        s +=i;
    }while(i<7);
    printf("%d\n", s);
}

```

执行后输出结果是

- A) 16 B) 12 C) 28 D) 21

(25) 有以下程序

```

main()
{
    int    i=10, j=1;
    printf("%d, %d\n", i--, ++j);
}

```

执行后输出结果是

- A) 9, 2 B) 10, 2 C) 9, 1 D) 10, 1

(26) 有以下程序

```

main()
{
    char  a,b,c,*d;
    a=' \' ;      b= '\xbc' ;
    c= '\0xab' ;   d="\017";
    print ("%c%c%c\n", a, b, c, *d);
}

```

编译时出现错误，以下叙述中正确的是

- A) 程序中只有 a=' \'；语句不正确
- B) b= '\xbc'；语句不正确
- C) d="\0127"；语句不正确
- D) a=' \'；和 c= '\0xab'；语句都不正确

(27) 有以下程序

```

int f1(int x, int y)
{
    return x>y?x:y;
}
int f2(int x, int y)
{
    return x>y?y:x;
}
main()
{
    int a=4, b=3, c=5, d, e, f;
    d=f1(a, b); d=f1(d, c);
    e=f2(a, b); e=f2(e, c);
    f=a+b+c-d-e;
    printf ("%d, %d, %d\n", d, f, e);
}

```

执行后输出结果是

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

(28) 有如下程序

```

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

```

执行后输出结果是

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

(29) 若有以下定义和语句

```
#include <stdio.h>
int a=4, b=3, *p, *q, *w;
p=&a; q=&b; w=q; q=NULL;
```

则以下选项中错误的语句是

- A) *q=0; B) w=p; C) *p=a; D) *p=*w;

(30) 有以下程序

```
int *f(int *x, int *y)
{   if(*x<*y)   return x;
    else       return y;
}
main()
{   int a=7, b=8, *p, *q, *r;
}
```

执行后输出结果是

- A) 7, 8, 8 B) 7, 8, 7 C) 8, 7, 7 D) 8, 7, 8

(31) 有以下程序

```
main()
{   char s[]={"one", "two", "three"}, *p;
    p=s[1];
    printf("%c, %s\n", *(p+1), s[0]);
}
```

执行后输出结果是

- A) n, two B) t, one C) w, one D) o, two

(32) 有以下程序

```
main()
{   int x[8]={8, 7, 6, 5, 0, 0}, *s;
    s=x+3
    printf("%d\n", s[2]);
}
```

执行后输出结果是

- A) 随机值 B) 0 C) 5 D) 6

(33) 以下能正确定义数组并正确赋初值的语句是

- A) int N=5, b[N][N]; B) int a[1][2]={ {1}, {3} };
C) int c[2][]={ {1, 2}, {3, 4} }; D) int d[3][2]={ {1, 2}, {34} };

(34) 有以下程序

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

执行后输出结果是

A) 4 5 6

B) 2 5 8

C) 3 6 9

D) 7 8 9

(35) 以下函数的功能是：通过键盘输入数据，为数组中的所有元素赋值。

```
#define    N    10
void    arrin(int    x[N])
{
    int    i=0;
    while(i<N)
        scanf("%d", _____);
}
```

在下划线处应填入的是

A) x+i

B) &x[i+1]

C) x+(i++)

D) &x[++i]

(36) 有以下程序

```
main()
{
    char    s[]="\n123\\";
    printf("%d, %d\\n", strlen(s), sizeof(s));
}
```

执行后输出结果是

A) 赋初值的字符串有错

B) 6, 7

C) 5, 6

D) 6, 6

(37) 阅读以下函数

```
fun(char    *s1, char    *s2)
{
    int    i=0;
    while(s1[i]==s2[i]&& s2[i]!='\0') i++;
    return(s1[i]== && s2[i]!='\0');
}
```

此函数的功能是

A) 将 s2 所指字符串赋给 s1

B) 比较 s1 和 s2 所指字符串的大小，若 s1 比 s2 的大，函数值为 1，否则函数值为 0

C) 比较 s1 和 s2 所指字符串是否相等，若相等，函数值为 1，否则函数值为 0

D) 比较 s1 和 s2 所指字符串的长度，若 s1 比 s2 的长，函数值为 1，否则函数

值为 0

(38) 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

(39) 设有如下说明

```
typedef struct
{   int    n;  char    c;  double x;}STD;
```

则以下选项中, 能正确定义结构体数组并赋初值的语句是

- A) STD tt[2]={ {1, 'A', 62}, {2, 'B', 75} };
- B) STD tt[2]={1, "A", 62}, 2, "B", 75};
- C) struct tt[2]={ {1, 'A'}, {2, 'B'} };
- D) structttt[2]={ {1, "A", 62.5}, {2, "B", 75.0} };

(40) 有以下程序

```
main()
{   union { unsigned int    n;
          unsigned char    c;
        }u1;
    u1.c= 'A ';
    printf("%c\n", u1.n);
}
```

执行后输出结果是

- A) 产生语法错
- B) 随机值
- C) A
- D) 65

(41) 有以下程序

```
main()
{   char    str []="xyz", *ps=str;
    while(*ps)ps++;
    for (ps--;ps-str>=0;ps--)    puts(ps);
}
```

执行后的输出结果是

- | | | | |
|-------|------|------|------|
| A) yz | B) z | C) z | D) x |
| xyz | yz | yz | xy |
| xyz | xyz | | |

(42) 有以下程序

```
main()
```

```

{   int a[][3]={1, 2, 3}, {4, 5, 0}}, {pa}[3], i;
pa=a;
for (i=0; i<3; i++)
if (i<2)   pa[1][i]=pa[1][i]-1;
else   pa[1][i]=1;
printf("%d\n", a[0][1]+a[1][1]+a[1][2]);
}

```

执行后输出结果是

- A) 7 B) 6 C) 8 D) 无确定值

(43) 有以下程序

```

void fun(int *a, int i, int j)
{   int t;
if (i<j)
    { t=a[i]; a[i]=a[j]; a[j]=t;
      fun(a, ++i, --j);
    }
}

main()
{   int a[]={1, 2, 3, 4, 5, 6}, i;
    fun(a, 0, 5)
    for (i=0; i<6; i++)
        printf("%d", a[i]);
}

```

执行后的输出结果是

- A) 6 5 4 3 2 1 B) 4 3 2 1 5 6 C) 4 5 6 1 2 3 D) 1 2 3 4 5
6

(44) 有以下程序

```

main(int arge, char *argv[])
{   int n, i=0;
while (argv[1][i]!='\0')
{   n=fun(); i++;}
    printf("%d\n", n*argc);
}

int fun()
{   static int s=0;
    s+=1;
    return s;
}

```

```
}
```

假设程序经编译、连接后生成可执行文件 exam.exe, 若键入以下命令行

```
exam 123<回车>
```

则运行结果为

- A) 6 B) 8 C) 3 D) 4

(45) 以下程序段中, 能够通过调用函数 fun, 使 main 函数中的指针变量 p 指向一个合法的整型单元的是

- A) main()
{ int *p;
 fun(p);
}
- B) main
{ int *p;
 fun(&p);
}

- C) #include<stdlib.h>
main()
{ int *p;
 fun(&p);
}
- D) #include<stdlib.h>
main()
{ int *p;
 fun(p);
}
- int fun(int **p)
{ int s; p=&s;}
- int fun(int **p)
{ int s;p=&s;}
- int fun(int **p)
{ *p=(int *)malloc(2);}
- int fun(int *p)
{ p=(int *)malloc(sizeof(int));}

(46) 若要说明一个类型名 STP, 使得定义语句 STP s 等价于 char *s, 以下选项中正确的是

- A) typedef STP char *s;
C) typedef stp *char;
- B) typedef *char STP;
D) typedef char* STP;

(47) 设有如下定义

```
struct ss  
{ char name[10];  
  int age;  
  char sex;  
} std[3], *p=std;
```

下面各输入语句中错误的是

- A) scanf("%d", &(*p). age);
C) scanf("%c", &std[0]. sex)
- B) scanf("%s", &std.name);
D) scanf("%c", &(p->sex));

- (48) 设 char 型变量 x 中的值为 10100111, 则表达式 $(2+x) \wedge (\sim 3)$ 的值是
A) 10101001 B) 10101000 C) 11111101 D) 01010101
- (49) 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 语言中对二进制文件的访问速度比文本文件快
C) C 语言中, 随机读写方式不适用于文本文件
D) C 语言中, 顺序读写方式不适用于二进制文件
- (50) 以下程序企图把从终端输入的字符输出到名为 abc.txt 的文件中, 直到从终端读入字符#号时结束输入和输出操作, 但程序有错。

```
#include <stdio.h>
main()
{ FILE *fout; char ch;
  fout=fopen('abc.txt', 'w');
  ch=fgetc(stdin);
  while(ch!='#')
  { fputc(ch, fout);
    ch=fgetc(stdin);
  }
  fclose(fout);
}
```


出错的原因是
A) 函数 fopen 调用形式错误 B) 输入文件没有关闭
C) 函数 fgetc 调用形式错误 D) 文件指针 stdin 没有定义

二、填空题(每空 2 分, 共 20 分)

请将每空的正确答案写在答题卡上【1】—【20】序号的横线上, 答在试卷上不得分。

- (1) 用十六进制绘存储器中的字节地址进行编号, 若地址编号从 000 到 FFFF, 则该存储器的容量为【1】KB。
- (2) 假设在当前盘当前目录下有两个文件 A.TXT 和 B.TXT, 现要将文件 B.TXT 合并连接到文件 A.TXT 的后面。若使用 COPY 命令, 则完整的命令为【2】。
- (3) E-mail 地址由用户和域名两部分组成, 这两部分的分隔符为【3】。
- (4) 假设在当前盘当前目录下有一个可执行程序 USER.EXE。现要执行该程序, 并要求在执行过程中将显示输出的结果信息存入当前盘当前目录的文件 OUT.DAT 中, 则完整的 DOS 命令为【4】。
- (5) 在 DOS 环境下, 表示打印机的设备文件名为【5】。

(6) 若有语句

```
int i=-19, j=i%4;
printf("%d\n", j);
```

则输出结果是 **【6】**。

(7) 若有程序

```
main()
{ int i, j;
  scanf("i=%d, j=%d",&i, &j);
  printf("i=%d, j=%d\n", i, j);
}
```

要求给 i 赋 10, 给 j 赋 20, 则应该从键盘输入 **【7】**。

(8) 若有以下程序

```
main()
{ int p, a=5;
  if(p=a!=0)
    printf("%d\n", p);
  else
    printf("%d\n", p+2);
}
```

执行后输出结果是 **【8】**。

(9) 若有以下程序

```
main()
{ int a=4, b=3, c=5, t=0;
  if(a<b) t=a; a=b; b=t;
  if(a<c) t=a; a=c; c=t;
  printf("%d %d %d\n", a, b, c);
}
```

执行后输出结果是 **【9】**。

(10) 若有以下程序

```
main()
{ int
  a[4][4]={ {1, 2, -3, -4}, {0, -12, -13, 14}, {-21, 23, 0, -24}, {-31, 32, -33,
0}};
  int i, j, s=0;
  for(i=0; i<4; i++)
  { for(j=0; j<4; j++)
```

```

        {   if(a[i][j]<0)continue;
            if(a[i][j]==0)break;
            s+=a[i][j];
        }
    }
    printf("%d\n", s);
}

```

执行后输出的结果是 **【10】**。

(11) 若有以下程序

```

main()
{   char    a;
    a='H'-'A'+'0';
    print("%c\n", a);
}

```

执行后的输出结果是 **【11】**。

(12) 若有以下程序

```

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

```

执行后输出的结果是 **【12】**。

(13) 函数 YangHui 的功能是把杨辉三角形的数据赋给二维数组的下半三角，形式如下

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1

```

其构成规律是：

第 0 列元素和主对角线元素均为 1

其余元素为其左上方和正上方元素之和

数据的个数每行递增 1

请将程序补充完整。

```

#define N    6

```

```

void YangHui(int *[N][N])
{
    int i, j;
    x[0][0]=1
    for (i=1; i<N; i++)
    {
        x[i][0]= 【13】 =1;
        for (j=1; j<i; j++)
            x[i][j]= 【14】 ;
    }
}

```

- (14) 以下函数的功能是删除字符串 s 中的所有数字字符。请填空。

```

void dele(char *s)
{
    int n=0, i;
    for (i=0, s[i]; i++)
        if( 【15】 )
            s[n++]=s[i];
    s[n]= 【16】 ;
}

```

- (15) 设函数 findbig 已定义为求 3 个数中的最大值。以下程序将利用函数指针调用 findbig 函数。请填空。

```

main()
{
    int findbig(int, int, int);
    int (*f)(), x, y, z, big;
    f= 【17】 ;
    scanf("%d%d%d", &x, &y, &z);
    big=(*f)(x, y, z);
    printf("bing=%d\n", big);
}

```

- (16) 以下程序的输出结果是 **【18】**。

```

#defineint      MCRA(m)      2*m
#define        MCRB(n, m)    2*MCRA(n)+m
main()
{
    int i=2, j=3;
    printf("%d\n", MCRB(j, MCRA(i)));
}

```

- (17) 设有以下定义

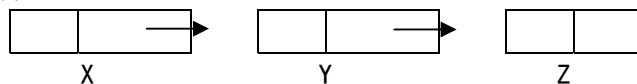
```

struct ss

```

```
{ int info;struct ss *link;}x,y,z;
```

且已建立如下图所示链表结构:



请写出删除点 y 的赋值语句【19】。

- (18) 已有文本文件 test.txt，其中的内容为：Hello, everyone!。以下程序中，文件 test.txt 已正确为“读”而打开，由文件指针 fr 指向该文件，则程序的输出结果是【20】。

```
#include <stdio.h>
main()
{ FILE *fr; char str[40];
.....
  fgets(str, 5, fr);
  printf("%s\n", str);
  fclose(fr);
}
```

评分标准及参考答案

一、选择题 ((1)~(40) 每小题 1 分, (41)~(50) 每小题 2 分, 共 60 分)

(1) C	(2) D	(3) D	(4) A	(5) D
(6) B	(7) A	(8) C	(9) D	(10) B
(11) C	(12) A	(13) B	(14) D	(15) B
(16) A	(17) D	(18) C	(19) A	(20) A
(21) C	(22) B	(23) D	(24) A	(25) B

(26) D	(27) C	(28) D	(29) A	(30) B
(31) C	(32) B	(33) D	(34) C	(35) C
(36) C	(37) C	(38) B	(39) A	(40) C
(41) C	(42) A	(43) A	(44) A	(45) C
(46) D	(47) B	(48) D	(49) D	(50) A

二、填空题(每空 2 分, 共 20 分)

(1)	【1】	64 (2^6)
(2)	【2】	COPY A. TXT+B. TXT A. TXT 或 COPY A. TXT+B. TXT
(3)	【3】	@
(4)	【4】	USER>OUT. DAT (USER>>OUT. DAT) 或 USER. EXE>OUT. DAT 或 USER. EXE>>OUT. DAT
(5)	【5】	PRN 或 LPT1 或 LPT1: 或 LPT2 或 LPT2:
(6)	【6】	-3
(7)	【7】	i=10, j=20
(8)	【8】	1
(9)	【9】	5 0 3
	【10】	58
	【11】	7
(10)	【12】	9
(11)	【13】	x[i][i]
(12)	【14】	x[i-1][j-1]+x[i-1][j] 或 [i-1][j]+x[i-1][j-1]
	【15】	s[i]<'0' s[i]>'9' 或 !(s[i]>='0' && s[i]<='9')
		或 !(isdigit(s[i]>'9' 或 !(s[i]>='0' && s[i]<='9'))
		或 s[i]>'9' s[i]<'0' 或 !(s[i]<='9' && s[i]>='0')
		或 *(s+i)<'0' *(s+i)>'9' 或 !(*(s+i)>='0' && *(s+i)<='9')
		或 !(isdigit(*(s+i))) 或 *(s+i)>'9' *(s+i)<'0'
		或 !(*(s+i)<='9' && *(s+i)>='0') 或 isdigit(*(s+i)) == 0
(13)	【16】	'\0' 或 0 或 NULL
	【17】	Findbig
	【18】	16
(14)	【19】	x. link=&z 或 x. link=y. ink
	【20】	Hell

全国计算机等级考试二级笔试试卷

基础知识

(考试时间 120 分钟, 满分 100 分)

2003 年 9 月

一、 选择题 ((1—40) 每小题 1 分, (41—50) 每小题 2 分, 共 60 分)

下列各题 A)、B)、C)、D) 四个选项中, 只有一个选项是正确的, 请将正确选项涂写在答题卡相应位置上, 答在试卷上不得分。

- (1) 在计算机中, 一个字节所包含二进制位的个数是
A) 2 B) 4 C) 8 D) 16
- (2) 在多媒体计算机中, CD-ROM 属于
A) 存储媒体 B) 传输媒体 C) 表现媒体 D) 表示媒体
- (3) 在 DOS 系统中, 带有通配符的文件名*. *表示
A) 硬盘上的全部文件 B) 当前盘当前目录中的全部文件
C) 当前盘上的全部文件 D) 根目录中的全部文件
- (4) 十六进制数 100 转换为十进制数为
A) 256 B) 512 C) 1024 D) 64
- (5) 能将高级语言编写的源程序转换为目标程序的软件是
A) 汇编程序 B) 编辑程序 C) 解释程序 D) 编译程序
- (6) 在 Internet 中, 用于在计算机之间传输文件的协议是
A) TELNET B) BBS C) FTP D) WWW
- (7) 在 Windows 环境下, 资源管理器左窗口中的某文件夹左边标有“+”标记表示
A) 该文件夹为空 B) 该文件夹中含子文件夹
C) 该文件夹中只包含有可执行文件 D) 该文件夹中包含系统文件
- (8) 在 Windows 环境下, 下列叙述中正确的是
A) 在“开始”菜单中可以增加项目, 也可以删除项目
B) 在“开始”菜单中不能增加项目, 也不能删除项目
C) 在“开始”菜单中可以增加项目, 但不能删除项目
D) 在“开始”菜单中不能增加项目, 但可以删除项目
- (9) 从 Windows 环境进入 DOS 方式后, 返回 Windows 环境的命令为
A) QUIT B) WIN C) EXIT D) ESC

(10) 下列叙述中正确的是

- A) 计算机病毒只感染可执行文件
- B) 计算机病毒只感染文本文件
- C) 计算机病毒只能通过软件复制的方式进行传播
- D) 计算机病毒可以通过读写磁盘或网络等方式进行传播

(11) 以下叙述中正确的是

- A) C 程序中注释部分可以出现在程序中任意合适的地方
- B) 花括号 “{” 和 “}” 只能作为函数体的定界符
- C) 构成 C 程序的基本单位是函数，所有函数名都可以由用户命名
- D) 分号是 C 语句之间的分隔符，不是语句的一部分

(12) 以下选项中可作为 C 语言合法整数的是

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

(13) 以下不能定义为用户标识符的是

- A) scanf B) Void C) _3com_ D) int

(14) 有以下程序

```
main()
{ int a;      char c=10;
  float f=100.0; double x;
  a=f/=c*=(x=6.5);
  printf("%d %d %3.1f %3.1f\n", a, c, f, x);
}
```

程序运行后的输出结果是

- A) 1 65 1 6.5 B) 1 65 1.5 6.5
- C) 1 65 1.0 6.5 D) 2 65 1.5 6.5

(15) 以下选项中非法的表达式是

- A) $0 \leq x < 100$ B) $i=j==0$ C) $(\text{char})(65+3)$ D) $x+1=x+1$

(16) 有以下程序

```
main()
{ int a=1, b=2, m=0, n=0, k;
  k=(n=b>a) || (m=a<b);
  printf("%d,%d\n", k, m);
}
```

程序运行后的输出结果是

- A) 0, 0 B) 0, 1 C) 1, 0 D) 1, 1

(17) 有定义语句：int x, y;; 若要通过 scanf("%d,%d",&x,&y); 语句使变量 x 得到

数值 11, 变量 y 得到数值 12, 下面四组输入形式中, 错误的是

- A) 11 12<回车>
- B) 11, 12<回车>
- C) 11, 12<回车>
- D) 11, <回车>
12<回车>

(18) 设有如下程序段:

```
int x=2002, y=2003;  
printf("%d\n", (x, y));
```

则以下叙述中正确的是

- A) 输出语句中格式说明符的个数少于输出项的个数, 不能正确输出
- B) 运行时产生出错信息
- C) 输出值为 2002
- D) 输出值为 2003

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

- A) $x=x*100+0.5/100.0$;
- B) $x=(x*100+0.5)/100.0$;
- C) $x=(int)(x*100+0.5)/100.0$;
- D) $x=(x/100+0.5)*100.0$;

(20) 有定义语句: int a=1, b=2, c=3, x; 则以下选项中各程序段执行后, x 的值不为 3 的是

- A) if (c<a) x=1;
else if (b<a) x=2;
else x=3;
- B) if (a<3) x=3;
else if (a<2) x=2;
else x=1;
- C) if (a<3) x=3;
if (a<2) x=2;
if (a<1) x=1;
- D) if (a<b) x=b;
if (b<c) x=c;
if (c<a) x=a;

(21) 有以下程序

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

若要使程序的输出值为 2, 则应该从键盘给 n 输入的值是

- A) -1
- B) -3
- C) -5
- D) 0

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

于 0)

```
s=a;
```

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

则与上述程序段功能等价的赋值语句是

A) $s=a+b$;

B) $s=a+c$;

C) $s=s+c$;

D) $s=b+c$;

(23) 有以下程序

```
main()
```

```
{ int k=4, n=4;
```

```
for ( ; n<k ;)
```

```
{ n++;
```

```
if (n%3!=0) continue;
```

```
k--; }
```

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

```
}
```

程序运行后的输出结果是

A) 1, 1

B) 2, 2

C) 3, 3

D) 4, 4

(24) 要求以下程序的功能是计算: $s = 1 + \frac{1}{2} + \frac{1}{3} \dots + \frac{1}{10}$

```
main ()
```

```
{ int n; float s;
```

```
s=1.0;
```

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

```
s=s+1/n;
```

```
print ("%6.4f\n", s);
```

```
}
```

程序运行后输出结果错误, 导致错误结果的程序行是

A) $s=1.0$;

B) $\text{for} (n=10;n>1;n--)$

C) $s=s+1/n$;

D) $\text{printf} ("%6.4f/n", s)$;

(25) 已定义 ch 为字符型变量, 以下赋值语句中错误的是

A) $\text{ch}=""$;

B) $\text{ch}=62+3$;

C) $\text{ch}=\text{NULL}$;

D) $\text{ch}="\text{xaa}"$;

(26) 若已定义的函数有返回值, 则以下关于该函数调用的叙述中错误的是

A) 调用可以作为独立的语句存在

B) 调用可以作为一个函数的实参

C) 调用可以出现在表达式中

D) 调用可以作为一个函数的形参

(27) 有以下函数定义:

```
void fun( int n, double x) { ..... }
```

若以下选项中的变量都已正确定义并赋值, 则对函数 fun 的正确调用语句是

- A) fun(int y, double m); B) k=fun(10, 12.5);
C) fun(x, n); D) void fun(n, x);

(28) 有以下程序

```
void fun(char *a, char *b)
{ a=b; (*a)++; }
main ()
{ char c1="A", c2="a", *p1, *p2;
  p1=&c1; p2=&c2; fun(p1, p2);
  printf( "&c&c\n", c1, c2);
}
```

程序运行后的输出结果是

- A) Ab B) aa C) Aa D) Bb

(29) 若程序中已包含头文件 stdio.h, 以下选项中, 正确运用指针变量的程序段是

- A) t *i=NULL; B) float *f=NULL;
scanf("%d", f); *f=10.5;
C) char t="m", *c=&t; D) long *L;
*c=&t; L= '\0' ;

(30) 有以下程序

```
#include <stdio.h>
main()
{ printf("%d\n", NULL); }
```

程序运行后的输出结果是

- A) 0 B) 1 C) -1 D) NULL 没定义, 出错

(31) 已定义 c 为字符型变量, 则下列语句中正确的是

- A) c= '97' ; B) c="97"; C) c=97; D) c="a";

(32) 以下不能正确定义二维数组的选项是

- 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} };

(33) 以下选项中不能正确把 c1 定义成结构体变量的是

- A) typedef struct B) struct color cl
{ int red; { int red;
 int green; int green;
 int blue; int blue;
} COLOR; };
COLOR cl;

```
C) struct color
{ int red;
  int green;
  int blue;
} c1;
```

```
D) struct
{ int red;
  int green;
  int blue;
} c1;
```

(34) 以下能正确定义一维数组的选项是

A) int num [];

B) #define N 100

int num [N];

C) int num[0..100];

D) int N=100;

int num[N];

(35) 下列选项中正确的语句组是

A) char s[8]; s={"Beijing"}; B) char *s; s={"Beijing"};

C) char s[8]; s="Beijing"; D) char *s; s="Beijing";

(36) 已定义以下函数

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

该函数的返回值是

A) 不确定的值

B) 形参 p 中存放的值

C) 形参 p 所指存储单元中的值

D) 形参 p 的地址值

(37) 下列函数定义中, 会出现编译错误的是

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

B) int max(int x, y)

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

```
int z ;
```

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

```
return z ;
```

```
}
```

C) max (int x, int y)

D) int max(int x, int y)

```
{ int z;
```

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

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

```
}
```

(38) 有以下程序

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

```
#define F(X,Y) (X)*(Y)
```

```
main ()
```

```
{ int a=3, b=4;
```

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

```
}
```

程序运行后的输出结果是

A) 12

B) 15

C) 16

D) 20

(39) 有以下程序

```
fun(int a, int b)
{ if(a>b) return(a);
  else return(b);
}
main()
{ int x=3, y=8, z=6, r;
  r=fun (fun(x,y), 2*z);
  printf("%d\n", r);
}
```

程序运行后的输出结果是

A) 3

B) 6

C) 8

D) 12

(40) 若有定义: int *p[3];, 则以下叙述中正确的是

A) 定义了一个基类型为 int 的指针变量 p, 该变量具有三个指针

B) 定义了一个指针数组 p, 该数组含有三个元素, 每个元素都是基类型为 int 的指针

C) 定义了一个名为 *p 的整型数组, 该数组含有三个 int 类型元素

D) 定义了一个可指向一维数组的指针变量 p, 所指一维数组应具有三个 int 类型元素

(41) 以下程序中函数 scmp 的功能是返回形参指针 s1 和 s2 所指字符串中较小字符串的首地址

```
#include <stdio.h>
#include <string.h>
char *scmp(char *s1, char *s2)
{ if( strcmp (s1, s2<0)
    return(s1);
  else return(s2);
}
```

main ()

```
{ int i; char string[20], str[3][20];
  for(i=0; i<3; i++) gets(str[i]);
```

strcpy(string, scmp(str[0], str[1])); /*库函数 strcpy 对字符串进行复制*/

```
strcpy(string, scmp(string, str[2]));
```

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

制*/

```
}
```

运行时依次输入: abcd、abba 和 abc 三个字符串, 则输出结果为

- A) abcd B) abba C) abc D) abca

(42) 有以下程序

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

程序运行后的输出结果是

- A) 10 B) 11 C) 20 D) 21

(43) 有以下程序段

```
main()
{ int a=5, *b, **c;
  c=&b; b=&a;
  .....
}
```

程序在执行了 c=&b:b=&a; 语句后, 表达式: **c 的值是

- A) 变量 a 的地址 B) 变量 b 中的值 C) 变量 a 中的值 D) 变量 b 的地址

(44) 有以下程序

```
#include <string.h>
main()
{ char str[][20]={"Hello", "Beijing"}, *p=str;
  printf("%d\n", strlen(p+20));
}
```

程序运行后的输出结果是

- A) 0 B) 5 C) 7 D) 20

(45) 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) 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'

(46) 有以下程序

```
main()
{ int  x=3, y=2, z=1;
  printf("%d\n", x/y&~z);
}
```

程序运行后的输出结果是

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

(47) 若 fp 已正确定义并指向某个文件，当未遇到该文件结束标志时函数 feof(fp) 的值为

- A) 0 B) 1 C) -1 D) 一个非 0 值

(48) 下列关于 C 语言数据文件的叙述中正确的是

- A) 文件由 ASCII 码字符序列组成，C 语言只能读写文本文件
B) 文件由二进制数据序列组成，C 语言只能读写二进制文件
C) 文件由记录序列组成，可按数据的存放形式分为二进制文件和文本文件
D) 文件由数据流形式组成，可按数据的存放形式分为二进制文件和文本文件

(49) 有以下程序

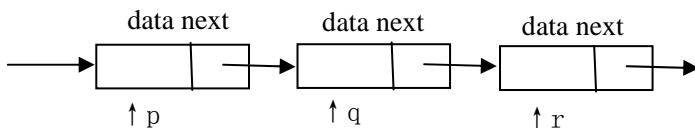
```
main()
{ int  a[3][3], *p, i;
  p=&a [0][0] ;
  for ( i=0 ; i<9 ; i++) p[i]=i+1;
  printf("%d\n", a[1][2]);
}
```

程序运行后的输出结果是

- A) 3 B) 6 C) 9 D) 2

(50) 有以下结构体说明和变量定义，如图所示，指针 p、q、r 分别指向一个链表中的三个连续结点。

```
struct  node
{ int  data
  struct  node  *next ;
} *p,  *q,  *r;
```



现要将 q 和 r 所指结点的先后位置交换，同时要保持链表的连续，以下错误的程

序段是

- A) $r \rightarrow next = q$; $q \rightarrow next = r \rightarrow next$; $p \rightarrow next = r$;
- B) $q \rightarrow next = r \rightarrow next$; $p \rightarrow next = r$; $r \rightarrow next = q$;
- C) $p \rightarrow next = r$; $q \rightarrow next = r \rightarrow next$; $r \rightarrow next = q$;
- D) $q \rightarrow next = r \rightarrow next$; $r \rightarrow next = q$; $p \rightarrow next = r$;

二、填空题（每空 2 分，共 40 分）

请将每一个空的正确答案写在答题卡【1】至【20】序号的横线上，答在试卷上不得分。

- (1) 计算机软件分为系统软件与应用软件，UNIX 属于【1】。
- (2) 假设在当前盘的当前目录下有两个文件 A. TXT 和 B. TXT，现要将文件 B. TXT 合并连接到文件 A. TXT 后面。若使用 TYPE 命令，则完整的命令为【2】。
- (3) Internet 的信息服务方式分为基本服务和扩展服务，电子邮件属于【3】。
- (4) 在 Windows 环境下，利用菜单命令删除固定硬盘上的文件与文件夹，实际上是将需要删除的文件与文件夹移动到【4】文件夹中。
- (5) 在 Windows 环境下，为了进行“添加/删除程序”操作，首先打开【5】，然后双击其中的相应图标。

- (6) 以下程序运行后的输出结果是【6】。

```
main()
{ int  p=30;
  printf ( "%d\n" , (p/3>0 ? p/10 : p%3));
}
```

- (7) 以下程序运行后的输出结果是【7】。

```
main()
{ char  m;
  m= 'B' +32;  printf ("%c\n",m);
}
```

- (8) 以下程序运行后的输出结果是【8】。

```
main()
{ int  a=1, b=3, c=5;
  if (c=a+b) printf("yes\n");
  else  printf("no\n");
}
```

- (9) 以下程序运行后的输出结果是【9】。


```

main()
{ int i,m=0, n=0, k=0;
  for (i=9; i<=11; i++)
    switch(i/10)
    { case 0 : m++; n++; break;
      case 10: n++;break;
      default: k++;n++;
    }
  printf("%d %d %d\n",m,n,k);
}

```

(10) 执行以下程序后，输出‘#’号的个数是 【10】。

```

#include <stdio.h>
main()
{ int i,j;
  for(i=1; i<5; i++)
    for(j=2; j<=i; j++) putchar('#');
}

```

(11) 以下程序的功能是调用函数 fun 计算： $m=1-2+3-4+\dots+9-10$ ，并输出结果。请填空。

```

int fun(int n)
{ int n=0,f=1, i;
  for (i=1; i<=n; i++)
  { m+=i*f;
    f= 【11】 ;
  }
  return m;
}
main()
{ printf("m=%d\n", 【12】 );}

```

(12) 以下程序运行后的输出结果是 【13】

```

main()
{ int i, n[]={0,0,0,0,0};
  for(i=1; i<=4; i++)
  { n[i]=n[i-1]*2+1;
    printf("%d",n[i]);
  }
}

```

(13) 以下程序运行后的输出结果是 【14】

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

(14) 以下程序运行后的输出结果是 【15】

```
int a=5;
fun(int b)
{ static int a=10;
  a+=b++;
  printf("%d", a);
}
main()
{ int c=20;
  fun(c);
  a+=c++;
  printf("%d\n", a);
}
```

(15) 请在以下程序第一行的下划线处填写适当内容, 使程序能正确运行。

```
【16】 (double, double);
main()
{ double x, y;
  scanf("%1f%1f", &x, &y);
  printf("%1f\n", max(x, y));
}
double max(double a, double b)
{ return (a>b ? a:b) ;}
```

(16) 以下程序运行后输入: 3, abcde<回车>, 则输出结果是 【17】

```
#include <string.h>
move(char *str, int n)
{ char temp; int i;
  temp=str[n-1];
```

```

        for (i=n-1; i>0; i--)    str[i]=str[i-1];
        str[0]=temp;
    }
    main()
    {   char    s[50];        int    n, i, z;
        scanf("%d,%s", &n, s);
        z=strlen(s);
        for (i=1; i<=n; i++)    move(s, z);
        printf("%s\n", s);
    }

```

(17) 以下程序运行后的输出结果是 **【18】**

```

fun(int    x)
{   if (x/2>0)    fun(x/2);
    printf("%d", x);
}
main()
{   fun (6); }

```

(18) 已有定义如下:

```

struct    node
{   int    data;
    struct    node    *next;
} *p;

```

以下语句调用 malloc 函数, 使指针 p 指向一个具有 struct node 类型的动态存储空间。请填写。

p=(struct node *) malloc(**【19】**);

(19) 以下程序的功能是将字符串 s 中的数字字符放入 d 数组中, 最后输出 d 中的字符串。例如, 输入字符串: abc123edf456gh, 执行程序后输出: 123456。请填写。

```

#include    <stdio.h>
#include    <ctype.h>
main()
{   char    s[80],    d[80];        int    i, j;
    gets(s);
    for (i=j=0; s[i]!='\0' ; i++)
        if( 【20】 )    { d[j]=s[i]; j++;}
    d[j]='\0' ;
    puts(d);
}

```

评分标准和参考答案

一、选择题（(1)～(40)每小题 1 分，(41)～(50)每小题 2 分，共 60 分）

(1) C	(2) A	(3) B	(4) A	(5) D
(6) C	(7) B	(8) A	(9) C	(10) D
(11) A	(12) C	(13) D	(14) B	(15) D
(16) C	(17) A	(18) D	(19) C	(20) C
(21) B	(22) B	(23) C	(24) C	(25) A
(26) D	(27) C	(28) A	(29) D	(30) A
(31) C	(32) D	(33) B	(34) B	(35) D
(36) C	(37) B	(38) A	(39) D	(40) B
(41) B	(42) B	(43) C	(44) C	(45) A
(46) D	(47) A	(48) D	(49) B	(50) A

二、填空题（每空 2 分，共 20 分）

(1)	【1】	系统软件
(2)	【2】	TYPE B. T XT >> A. TXT
(3)	【3】	基本服务
(4)	【4】	回收站
(5)	【5】	控制面板
(6)	【6】	3
(7)	【7】	b
(8)	【8】	yes
(9)	【9】	1 3 2
(10)	【10】	6
(11)	【11】	-f
	【12】	fun(10)
(12)	【13】	1 3 7 15
(13)	【14】	1 2 3 0 5 6 0 0 9
(14)	【15】	30 25
(15)	【16】	double max
(16)	【17】	cdeab
(17)	【18】	1 3 6
(18)	【19】	sizeof(struct node)
(19)	【20】	s[i]>='0' && s[i]<='9'

全国计算机等级考试二级笔试试卷

基础知识

(考试时间 120 分钟, 满分 100 分)

2004 年 4 月

一、 选择题 ((1—40) 每小题 1 分, (41—50) 每小题 2 分, 共 60 分)

下列各题 A)、B)、C)、D) 四个选项中, 只有一个选项是正确的, 请将正确选项涂写在答题卡相应位置上, 答在试卷上不得分。

- (1) 1MB 等于
 - A) 1000 字节
 - B) 1024 字节
 - C) 1000×1000 字节
 - D) 1024×1024 字节
- (2) 与十六进制数 200 等值的十进制数为
 - A) 256
 - B) 512
 - C) 1024
 - D) 2048
- (3) 所谓“裸机”是指
 - A) 单片机
 - B) 单板机
 - C) 不装备任何软件的计算机
 - D) 只装备操作系统的计算机
- (4) 能将高级语言编写的源程序转换成目标程序的是
 - A) 链接程序
 - B) 解释程序
 - C) 编译程序
 - D) 编辑程序
- (5) 在 64 位计算机中, 一个字长所占的字节数为
 - A) 64
 - B) 8
 - C) 4
 - D) 1
- (6) 在 Windows 环境下, 当一个应用程序窗口被最小化后, 该应用程序
 - A) 继续在后台运行
 - B) 继续在前台运行
 - C) 终止运行
 - D) 暂停运行
- (7) 在 Windows 环境下, 能实现窗口移动的操作是
 - A) 用鼠标拖动窗口中的任何部位
 - B) 用鼠标拖动窗口的边框
 - C) 用鼠标拖动窗口的控制按钮
 - D) 用鼠标拖动窗口的标题栏
- (8) 在 Windows 环境下, PrintScreen 键的作用是
 - A) 打印当前窗口的内容
 - B) 打印屏幕内容
 - C) 复制屏幕到剪贴板
 - D) 复制当前窗口到剪贴板
- (9) Internet 的通信协议是
 - A) TCP/IP
 - B) BBS
 - C) WWW
 - D) FTP

- (10) 下列叙述中正确的是
- A) 计算机病毒只感染可执行文件
 - B) 计算机病毒只感染文本文件
 - C) 计算机病毒只能通过软件复制的方式进行传播
 - D) 计算机病毒可以通过网络或读写磁盘等方式进行传播
- (11) 一个算法应该具有“确定性”等 5 个特性，下面对另外 4 个特性的描述中错误的是
- A) 有零个或多个输入
 - B) 有零个或多个输出
 - C) 有穷性
 - D) 可行性
- (12) 以下叙述中正确的是
- A) C 语言的源程序不必通过编译就可以直接运行
 - B) C 语言中的每条可执行语句最终都将被转换成二进制的机器指令
 - C) C 源程序经编译形成的二进制代码可以直接运行
 - D) C 语言中的函数不可以单独进行编译
- (13) 以下符合 C 语言语法的实型常量是
- A) 1.2E0.5
 - B) 3.14.159E
 - C) .5E-3
 - D) E15
- (14) 以下 4 组用户定义标识符中，全部合法的一组是
- | | | | |
|-----------------------|--------------------|---------------------|---------------------|
| A) <code>_main</code> | B) <code>lf</code> | C) <code>txt</code> | D) <code>int</code> |
| <code>enclude</code> | <code>-max</code> | <code>REAL</code> | <code>k_2</code> |
| <code>sin</code> | <code>turbo</code> | <code>3COM</code> | <code>_001</code> |
- (15) 若以下选项中的变量已正确定义，则正确的赋值语句是
- A) `x1=26.8%3`
 - B) `1+2=x2`
 - C) `x3=0x12`
 - D) `x4=1+2=3;`
- (16) 设有以下定义
- ```
int a=0;
double b=1.25;
char c='A' ;
#define d 2
```
- 则下面语句中错误的是
- A) `a++;`
  - B) `b++`
  - C) `c++;`
  - D) `d++;`
- (17) 设有定义：`float a=2, b=4, h=3;`，以下 C 语言表达式与代数式 计算结果不相符的是
- A) `(a+B) *h/2`
  - B) `(1/2)*(a+B) *h`
  - C) `(a+B) *h*1/2`
  - D) `h/2*(a+B)`
- (18) 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

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

(19) 以下 4 个选项中，不能看作一条语句的是

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

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

(21) 以下程序段中与语句 k=a>b?(b>c?1:0):0; 功能等价的是

- A) if((a>B) &&(b>C)) k=1; B) if((a>B) || (b>C)) k=1
else k=0;
C) if(a<=B) k=0; D) if(a>B) k=1;
else if(b<=C) k=1; else if(b>C) k=1;
else k=0;

(22) 有以下程序

```

main( )
{ char k; int i;
  for(i=1; i<3; i++)
  { scanf( "%c" , &k);
    switch(k)
    { case '0' : printf( "another\n");
      case '1' : printf( "number\n");
    }
  }
}

```

程序运行时，从键盘输入：01<回车>，程序执行后的输出结果是

- A) another B) another C) another D) number
number number number number
another number

(23) 有以下程序

```

main( )
{ int x=0, y=5, z=3;
  while(z-->0&&++x<5) y=y-1;
}

```



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

程序执行后的输出结果是

- A) 3, 2, 0 B) 3, 2, -1 C) 4, 3, -1 D) 5, -2, -5

(24) 有以下程序

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

程序执行后的输出结果是

- 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 中的奇数之和 D) 自然数 1~10 中的偶数之和

(25) 有以下程序

```
main( )
{   int i, n=0;
    for (i=2; i<5; i++)
        {   do
            {   if(i%3)   continue;
                n++;
            } while(!i);
        }
    n++;
    printf( "n=%d\n" , n);
}
```

程序执行后的输出结果是

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

(26) 若程序中定义了以下函数

```
double  myadd(double a, double B)
{ return (a+B) ;}
```

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

- A) double myadd(double a, B) ;
B) double myadd(double, double);
C) double myadd(double b, double A) ;
D) double myadd(double x, double y);

(27) 有以下程序

```
char fun(char x , char y)
{ if(x<y) return x;
  return y;
}
main( )
{ int a=' 9' ,b=' 8' ,c=' 7' ;
  printf( "%c\n" , fun(fun(a, B) , fun(b, C) ));
}
```

程序的执行结果是

- A) 函数调用出错 B) 8 C) 9 D) 7

(28) 设有定义: int n=0,*p=&n,**q=&p;则以下选项中, 正确的赋值语句是

- A) p=1; B) *q=2; C) q=p; D) *p=5;

(29) 有以下程序

```
void f(int v , int w)
{ int t;
  t=v;v=w;w=t;
}
main( )
{ int x=1,y=3,z=2;
  if(x>y) f(x,y);
  else if(y>z) f(y,z);
  else f(x,z);
  printf( "%d,%d,%d\n" , x,y,z);
}
```

执行后输出结果是

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

(30) 有以下程序段

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

b 中的值是

- A) 5 B) 6 C) 8 D) 9

(31) 有以下程序

```
main( )
{ char a[]=" abcdefg" , b[10]=" abcdefg" ;
  printf( "%d %d\n" , sizeof(A) , sizeof(B) );
}
```

```
}
```

执行后输出结果是

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

(32) 有以下程序

```
void swap1(int c[])
{ int t;
  t=c[0];c[0]=c[1];c[1]=t;
}
void swap2(int c0,int c1)
{ int t;
  t=c0;c0=c1;c1=t;
}
main( )
{ int a[2]={3,5},b[2]={3,5};
  swap1(A) ; swap2(b[0],b[1]);
  printf(“%d %d %d %d\n”,a[0],a[1],b[0],b[1]);
}
```

其输出结果是

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

(33) 有以下程序

```
void sum(int *A)
{ a[0]=a[1];}
main( )
{ int aa[10]={1,2,3,4,5,6,7,8,9,10},i;
  for(i=2;i>=0;i--) &nb
    sp; sum(&aa[i]);
  printf(“%d\n”,aa[0]);
}
```

执行后的输出结果是

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

(34) 34、有以下程序

```
int f(int b[][4])
{ int i,j,s=0;
  for(j=0;j<4;j++)
  { i=j;
    if(i>2) i=3-j;
    s+=b[i][j];
  }
```

```

    }
    return s;
}
main( )
{ int a[4][4]={ {1, 2, 3, 4}, {0, 2, 4, 5}, {3, 6, 9, 12}, {3, 2, 1, 0} };
  printf( "%d\n", f(A) );
}

```

执行后的输出结果是

- A) 12 B) 11 C) 18 D) 16

(35) 有以下定义

```

#include <stdio.h>
char a[10], *b=a;

```

不能给数组 a 输入字符串的语句是

- A) gets(A) B) gets(a[0]) C) gets(&a[0]); D) gets(B) ;

(36) 有以下程序

```

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

```

执行后输出结果是

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

(37) 有以下程序

```

int a=2;
int f(int *A)
{return (*A) ++;}
main( )
{ int s=0;
  { int a=5;
    s+=f(&a) ;
  }
  s+=f(&a) ;
  printf( "%d\n", s);
}

```

执行后输出结果是

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

(38) 有以下程序

```

#define f(x)    x*x
main( )
{   int i;
    i=f(4+4)/f(2+2);
    printf( "%d\n" , i);
}

```

执行后输出结果是

- A) 28 B) 22 C) 16 D) 4

(39) 设有以下语句

```

typedef struct S
{   int g; char h;} T;

```

则下面叙述中正确的是

- A) 可用 S 定义结构体变量 B) 可以用 T 定义结构体变量
C) S 是 struct 类型的变量 D) T 是 struct S 类型的变量

(40) 有以下程序

```

struct STU
{   char name[10];
    int num;
};
void f1(struct STU C)
{   struct STU b={ "LiSiGuo" , 2042};
    c=b;
}
void f2(struct STU *C)
{   struct STU b={ "SunDan" , 2044};
    *c=b;
}
main( )
{   struct STU a={ "YangSan" , 2041}, b={ "WangYin" , 2043};
    f1(A) ;f2(&B) ;
    printf( "%d %d\n" , a. num, b. num);
}

```

执行后的输出结果是

- A) 2041 2044 B) 2041 2043 C) 2042 2044 D) 2042 2043

(41) 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) 1 1 D) 43 0

(42) 下面程序的功能是输出以下形式的金字塔图案是:

```

*
***
*****
*****

```

```

main( )
{ int i, j;
  for(i=1; i<=4; i++)
  { for(j=1; j<=4-i; j++)      printf( " ");
    for(j=1; j<=_____; j++)    printf( "*" );
    printf( "\n" );
  }
}

```

在下划线处应填入的是

- A) i B) 2*i-1 C) 2*i+1 D) i+2

(43) 有以下程序

```

void sort(int a[], int n)
{ int i, j, t;
  for(i=0; i<n-1; i+=2)
    for(j=i+2; j<n; j+=2)
      if(a[i]<a[j]) { t=a[i]; a[i]=a[j]; a[j]=t; }
}
main()
{ int aa[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, i;
  sort(aa, 10);
  for(i=0; i<10; i++)    printf( "%d" , aa[i]);
  printf( "\n" );
}

```

其输出结果是

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

- (44) 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[]=" abcdefg" ,*s=t;
C) char s[10];s=" abcdefg" ; D) char s[10];strcpy(s, " abcdefg");

- (45) 有以下程序

```
#include <string.h>
main(int argc ,char *argv[ ])
{ int i, len=0;
  for(i=1;i<argc;i+=2) len+=strlen(argv[i]);
  printf("5d\n", len);
}
```

经编译链接后生成的可执行文件是 ex.exe，若运行时输入以下带参数的命令行

ex abcd efg h3 k44

执行后输出结果是

- A) 14 B) 12 C) 8 D) 6

- (46) 有以下程序

```
void f(int a[], int i, int j)
{ int t;
  if(i<j)
  { t=a[i]; a[i]=a[j]; a[j]=t;
    f(a, i+1, j-1);
  }
}
main( )
{ int i, aa[5]={1, 2, 3, 4, 5};
  f(aa, 0, 4);
  for(i=0; i<5; i++) printf("%d, ", aa[i]); printf("\n");
}
```

执行后输出结果是

- A) 5, 4, 3, 2, 1, B) 5, 2, 3, 4, 1, C) 1, 2, 3, 4, 5, D) 1, 2, 3, 4, 5,

- (47) 有以下程序

```
struct STU
{ char name[10];
  int num;
  int Score;
};
main( )
{ struct STU s[5]={{"YangSan", 20041, 703}, {"LiSiGuo", 20042, 580},
```

```

{ "wangYin" , 20043, 680}, { "SunDan" , 20044, 550},
  { "Penghua" , 20045, 537}}, *p[5], *t;
  int i, j;
  for (i=0; i<5; i++)   p[i]=&s[i];
  for (i=0; i<4; i++)
    for (j=i+1; j<5; j++)
      if (p[i]->Score>p[j]->Score)
        { t=p[i]; p[i]=p[j]; p[j]=t; }
  printf( "5d  %d\n" , s[1]. Score, p[1]->Score);
}

```

执行后输出结果是

A) 550 550 B) 680 680 C) 580 550 D) 580 680

(48) 有以下程序

```

#include <stdlib.h>
struct NODE{
  int num;
  struct NODE *next;
};
main( )
{
  struct NODE *p, *q, *r;
  int sum=0;
  p=(struct NODE *)malloc(sizeof(struct NODE));
  q=(struct NODE *)malloc(sizeof(struct NODE));
  r=(struct NODE *)malloc(sizeof(struct NODE));
  p->num=1; q->num=2; r->num=3;
  p->next=q; q->next=r; r->next=NULL;
  sum+=q->next->num; sum+=p->num;
  printf( "%d\n" , sum);
}

```

执行后的输出结果是

A) 3 B) 4 C) 5 D) 6

(49) 有以下程序

```

#include <stdio.h>
main( )
{
  FILE *fp;  int i, k=0, n=0;
  fp=fopen( "d1. dat" , " w" );
  for (i=1; i<4; i++)   fprintf(fp, " %d" , i);
}

```



```

    fclose(fp);
    fp=fopen("d1.dat", "r");
    fscanf(fp, "%d%d", &k, &n);    printf("%d %d\n", k, n);
    fclose(fp);
}

```

执行后输出结果是

- A) 1 2 B) 123 0 C) 1 23 D) 0 0

- (50) 有以下程序（提示：程序中 `fseek(fp, -2L*sizeof(int), SEEK_END)`；语句的作用是使位置指针从文件尾向前移 `2*sizeof(int)` 字节）

```

#include <stdio.h>
main( )
{ FILE *fp;  int i, a[4]={1, 2, 3, 4}, b;
  fp=fopen("data.dat", "wb");
  for(i=0; i<4; i++)  fwrite(&a[i], sizeof(int), 1, fp);
  fclose(fp);
  fp=fopen("data.dat", "rb");
  fseek(fp, -2L*sizeof(int), SEEK_END) ;
  fread(&b, sizeof(int), 1, fp); /*从文件中读取 sizeof(int) 字节的数据到变量 b 中*/
  fclose(fp);
  printf("%d\n", B) ;
}

```

执行后输出结果是

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

二、填空题（每空 2 分，共 40 分）

请将每一个空的正确答案写在答题卡【1】至【20】序号的横线上，答在试卷上不得分。

- (1) E-mail 地址由用户名和域名两部分组成，这两部分的分隔符为【1】。
- (2) 计算机网络分为广域网和局域网，因特网属于【2】。
- (3) 从 Windows 环境进入 MS-DOS 方式后，返回 Windows 环境的 DOS 命令为【3】。
- (4) 在 Windows 环境下，只要在“开始”菜单中单击【4】选项，就可以查阅最近打开过的文档清单。

(5) 计算机的运算速度可以用 MIPS 来描述, 1MIPS 是指每秒执行【5】条指令。

(6) 已知字符 A 的 ASCII 码值为 65, 以下语句的输出结果是【6】

```
char ch=' B' ;  
printf( "%c %d\n" , ch, ch);
```

(7) 有以下语句段

```
int  n1=10, n2=20;  
printf( " 【7】 ", n1, n2);
```

要求按以下格式输出 n1 和 n2 的值, 每个输出行从第一列开始, 请填空。

```
n1=10  
n2=20
```

(8) 有以下程序

```
main( )  
{ int  t=1, i=5;  
  for(; i>=0; i--)  t*=i;  
    printf( "%d\n" , t);  
}
```

执行后输出结果是【8】。

(9) 有以下程序

```
main( )  
{  int  n=0, m=1, x=2;  
    if(!n)    x-=1;  
    if(m)     x-=2;  
    if(x)     x-=3;  
    printf( "%d\n" , x);  
}
```

执行后输出结果是【9】。

(10) 有以下程序

```
#include <stdio.h>  
main( )  
{  char  ch1, ch2;  int  n1, n2;  
    ch1=getchar();  ch2=getchar();  
    n1=ch1- '0' ;  n2=n1*10+(ch2- '0' );  
    printf( "%d\n" , n2);  
}
```

程序运行时输入: 12<回车>, 执行后输出结果是【10】。

(11) 有以下程序

```
void f( int y, int *x)
{y=y+*x;  *x=*x+y;}
main( )
{  int x=2, y=4;
    f(y, &x);
    printf( "%d    %d\n", x, y);
}
```

执行后输出的结果是【11】。

(12) 函数 fun 的功能是计算 x^n

```
double fun(double x, int n)
{  int i; double y=1;
    for(i=1; i<=n; i++)  y=y*x;
    return y;
}
```

主函数中已经正确定义 m, a, b 变量并赋值, 并调用 fun 函数计算: $m=a^4+b^4-(a+b)^3$ 。
实现这一计算的函数调用语句为【12】。

(13) 下面 rotate 函数的功能是: 将 n 行 n 列的矩阵 A 转置为 A', 例如:

	1	2	3	4		1	5	9	13
	5	6	7	8		2	6	10	14
当 A=	9	10	11	12	则 A' =	3	7	11	15
	13	14	15	16		4	8	12	16

请填空

```
#define  N    4
void rotate(int a[][N])
{  int i, j, t;
    for(i=0; i<N; i++)
        for(j=0; 【13】; j++)
            {  t=a[i][j];
                【14】 ;
                a[j][i]=t;
            }
}
```

(14) 以下 strcpy() 函数实现字符串复制, 即将 t 所指字符串复制到 s 所指向内存空间中, 形成一个新的字符串 s。请填空。

```
void strcpy(char *s, char *t)
```

```

    { while(*s++= 【15】); }
main( )
{ char str1[100],str2[]=" abcdefgh" ;
  strcpy(str1, str2);
  printf( "%s\n" , str1);
}

```

(15) 下列程序的运行结果是: 【16】

```

#include <string.h>
char *ss(char *s)
{ return s+strlen(s)/2;}
main( )
{ char *p,*str=" abcdefgh" ;
  p=ss(str);   printf( "%s\n" , p);
}

```

(16) 下面程序的运行结果是: 【17】

```

int f( int a[], int n)
{ if(n>1) return a[0]+f(&a[1],n-1);
  else    return a[0];
}
main ( )
{ int aa[3]={1, 2, 3}, s;
  s=f(&aa[0], 3);   printf( "%d\n" , s);
}

```

(17) 以下程序中给指针 p 分配三个 double 型动态内存单元, 请填空。

```

# include <stdlib.h>
main ( )
{ double *p;
  p=(double *) malloc(【18】);
  p[0]=1.5;p[1]=2.5;p[2]=3.5;
  printf( "%f%f%f\n" , p[0], p[1], p[2]);
}

```

(18) 以下程序的运行结果是 【19】

```

# include <string.h>
typedef struct student{
    char name[10];
    long sno;
    float score;
}

```

```

    }STU;
main( )
{   STU
    a={ “zhangsan” , 2001, 95}, b={ “Shangxian” , 2002, 90}, c={ “Anhua” , 200
3, 95}, d, *p=&d;
    d=a;
    if(strcmp(a.name, b.name)>0)    d=b;
    if(strcmp(c.name, d.name)>0)    d=c;
    printf( “%ld%s\n” , d.sno, p->name);
    }

```

(18) 以下 sum 函数的功能是计算下列级数之和。

$$S=1+x+x^2/2!+x^3/3!+. +x^n/n!$$

请给函数中的各变量正确赋初值。

```

double sum( double x, int n )
{   int i; double a,b,s;
    【20】
    for( i=1; i<=n; i++)
        {   a=a*x;    b=b*i;    s=s+a/b; }
    return s;
}

```

评分标准及参考答案

一、选择题（(1)～(40)每小题 1 分，(41)～(50)每小题 2 分，共 60 分）

(1) D	(2) B	(3) C	(4) C	(5) B
(6) A	(7) D	(8) C	(9) A	(10) C
(11) B	(12) B	(13) C	(14) A	(15) C
(16) D	(17) B	(18) C	(19) D	(20) A
(21) A	(22) C	(23) B	(24) D	(25) D
(26) A	(27) D	(28) D	(29) C	(30) D
(31) C	(32) B	(33) A	(34) D	(35) B
(36) A	(37) C	(38) A	(39) D	(40) A
(41) A	(42) B	(43) C	(44) C	(45) D
(46) A	(47) C	(48) B	(49) B	(50) D

二、填空题：（每空 2 分，共 40 分）

1、@
2、广域网
3、EXIT 或 exit
4、文档
5、百万条或一百万条或 1000000 条
6、B 66
7、n1=%d\nn2=%d
8、0
9、-4
10、12
11、8 4
12、fun(a, 4)+fun(b, 4)-fun(a+b, 3);
13、j<=i
14、a[i][j]=a[j][i]
15、*t++
16、6
17、efgh
18、3*sizeof(double)
19、2002Shangxian
20、a=1.0;b=1.0;s=1.0;