

C语言复习题

12元

第1部分 程序设计基本概念

一、选择题

1、以下叙述不正确的是 (C)。

- A. 在C程序中，语句之间必须用分号“;”分隔
- B. 若a是实型变量，C程序中a=10是正确的，因为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，都能正确无误地表示
- D. 在C程序中，%是只能用于正数运算的运算符

2、以下叙述中不正确的是 (C)。

- A. 一个好的程序应该有详尽的注释
- B. C程序中的#include和#define均不是C语句
- C. 在C程序中，赋值运算符的优先级最低
- D. 在C程序中，“j++”是一条赋值语句

3、C语言中，当复合语句多于一条时，要有一对 (C) 括起来。

- A. 方括号
- B. 圆括号
- C. 花括号
- D. 尖括号

4、下列程序中格式错在 (C) 处。

- A. main ()
- B. int a, b, c;
- C. a=2: b=3;
- D. z=a+b;

5、C语言的程序一行写不下时，可以 (C)。

- A. 用逗号换行
- B. 用分号换行
- C. 在任意一空格处换行
- D. 用回车符换行

6、C语言中允许将一条语句写在两行上，下列语句不正确的是 (D)。

- A. int a, b;
- B. int a
- C. int a, b;
- D. int a, b

(C语言规定可以在语句中的任何一个分隔符或空格处换行)

7、下列程序中，(A) 是错误的注释方法。

- A. #in/*包含*/clude<stdio.h>
void main()
{
}
- B. #include<stdio.h>
void main() /* */ /* 主函数 */
{
}
- C. #include<stdio.h>
void main()
{ int x/* 初始化 */=10;
/* 打印 */ printf("%d",x);
}
- D. #include<stdio.h>
void main()
{ int x=10;
printf("%d",x);
/* 打印X的值 */*=10*/
}

(C语言规定：注释不能插在标识符的中间，在其它有分隔符的地方都可以插入注释)

8、(B) 不是C语言提供的合法关键字。

- A. switch
- B. print
- C. case
- D. default

9、下列说明正确的是 (D)。

- A. main函数必须放在C程序的开头
- B. main函数必须放在C程序的最后
- C. main函数可以放在C程序的中间，即在一些函数之前在另一些函数之后，但在执行C程序时是从程序开头执行的

D. main 函数可以放在 C 程序的中间部分, 即在一些函数之前在另一些函数之后, 但在执行 C 程序时是从 main 函数开始的

10、C 语言规定, 必须用 (C) 作为主函数名。

- A. function B. include C. main D. stdio

11、下列说法正确的是 (C)。

- A. 一个函数的函数体必须要有变量定义和执行部分, 二者缺一不可
B. 一个函数的函数体必须要有执行部分, 可以没有变量定义
C. 一个函数的函数体没有变量定义和执行部分, 函数可以是空函数
D. 以上都不对

12、在 C 语言中, 每个语句和数据定义是用 (C) 结束。

- A. 句号 B. 逗号 C. 分号 D. 括号

13、下列字符串是标识符的是 (A)。 (标识符只能以字母或下划线开头, C 语言的保留字不能作为标识符)

- A. HJ B. 9_student C. long D. LINE

14、以下说法中正确的是 (C)。 (用来标记常量、变量、数组、函数等名称的字符序列)

- A. C 语言程序总是从第一个定义的函数开始执行
B. 在 C 语言程序中, 要调用的函数必须在 main() 函数中定义
C. C 语言程序总是从 main() 函数开始执行
D. C 语言程序中的 main() 函数必须放在程序的开始部分

15、(A) 是 C 程序的基本构成单位。

- A. 函数 B. 函数和过程 C. 超文本过程 D. 子程序

16、C 语言提供的合法关键字是 (A)。

- A. continue B. procedure C. begin D. append

二、填空题

1、下列程序中的错误是 (程序中少了一个花括号 “}”)

```
void main()
{
    .....
    {
        .....
    }
}
```

2、在 VC++ 环境中用 Execute 命令运行一个 C 程序时, 所运行的程序扩展名是 (.EXE)

3、一个函数由两部分组成, 它们是 (函数的说明部分 (函数头部)) 和 (函数体)。

4、在 C 语言程序中, 凡在一个标识符后面紧跟着一对圆括弧, 就表明它是一个 (函数)。

5、主函数名后面的一对圆括号中间可以为空, 但一对圆括号不能 (省略)。

6、函数体一般包括 (变量定义部分、执行部分)。

7、C 语言的关键字都用 (小写)。(大写或小写)

第3部分 顺序结构

一、选择题

① 运行以下程序:

```
void main()
{
    int a; float b,c;
    scanf("%2d%3f%4f",&a,&b,&c);
    printf("a=%d,b=%f,c=%f\n",a,b,c);
}
```

在键盘上从第一列开始输入 9876543210<CR> (此处<CR>代表 Enter 键), 则程序的输出结果是 (C)。

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

② 若有以下程序段:

```
void main()
{
    int a=2,b=5;
    printf("a=%d,b=%d\n",a,b);
}
```

其输出结果是 (D)。

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

③ 若 short int 类型占两个字节, 则以下程序段:

short int a=-1;
printf("%d,%u\n", a,a);

其输出是 (D)。

- A. -1, -1
- B. -1, 32767
- C. -1, 32768
- D. -1, 65535

④ 以下程序段:

```
int x=496;
printf("%*-06d*\n",x);
```

其输出是 (A)。

- A. *496 *
- B. * 496*
- C. *000496*
- D. 输出格式符不合法

⑤ 以下程序段:

```
float a=3.1415;
printf("|%6.0f|\n",a);
```

其输出是 (C)。

- A. |3.1415|
- B. | 3.0|
- C. | 3|
- D. | 3.1|

6. printf("|%10.5f|\n",12345.678); 语句的输出是 (C)。

- A. |2345.67800|
- B. |12345.6780|
- C. |12345.67800|
- D. |12345.678|

7、以下程序段:

```
float a=57.666;
```

```
printf("010.2f\n",a);
```

其输出是 (C)。

A. *0000057.66*

B. * 57.66*

C. *0000057.67*

D. * 57.67*

8、若变量 c 定义为 float 类型, 当从终端输入 283.1900<CR>(<CR>代表回车键), 能给变量 c 赋以 283.19 的输入语句是 (D)。

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

B. scanf("%8.4f",&c);

C. scanf("%6.2f",&c);

D. scanf("%8f",&c);

9、若变量已正确说明, 要求用 scanf("a=%f,b=%f",&a,&b); 语句给 a 赋予 3.12、给 b 赋予 9.0, 则正确的输入形式是 (□代表空格, <CR>代表 Enter 键) (C)。

A. 3.12□□9.0<CR>

B. a=□□3.12b=□□□9<CR>

C. a=3.12, b=9<CR>

D. a=3.12□□, b=9.0□□<CR>

10、以下程序:

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

```
#include<math.h>
```

```
void main()
```

```
{ double a=-3.0,b=2;
```

```
printf("%3.0f%3.0f\n", pow(b,fabs(a)),pow(fabs(a),b));
```

```
}
```

其输出结果是 (B)。

A. 9 8

B. 8 9

C. 6 6

D. 以上三个都不对

11、下面程序的输出结果是 (C)。

```
void main()
```

```
{ int a=-1,b=4,k;
```

```
k=(++a<=0)&&!(b--<=0);
```

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

```
}
```

A. 1 0 4

B. 0 0 4

C. 1 0 3

D. 0 0 3

12、下面程序的输出结果是 (A)。

```
void main()
```

```
{ int x;
```

```
x=-3+4*5-6; printf("%d",x);
```

```
x=3+4%5-6; printf("%d",x);
```

```
x=-3*4%6/5; printf("%d",x);
```

```
x=(7+6)%5/2; printf("%d",x);
```

```
}
```

A. 11 1 0 1

B. 11 -3 2 1

C. 12 -3 2 1

D. 11 1 2 1

13、根据题目中已给出的数据输入和输出形式, 程序中输入输出语句的正确内容是 (B)。

```
void main()
```

```
{ int a; float x;
```

```
printf("input a,x: ");
```

输入语句

输出语句

```
}
```

输入形式: input a,x: 3 2.1

输出形式: a+x=5.10

A. scanf("%d,%f",&a,&x);

```
printf("\na+x=%4.2f",a+x);
```

C. scanf("%d%f",&a,&x);

```
printf("\na+x=%6.1f",a+x);
```

14、下面程序的结果为 (A)。

```
void main()
```

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

```
  x=y=1;
```

```
  z=x++-1;
```

```
  printf("%d,%d\t",x,z);
```

```
  z+=-x+++(++y);
```

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

```
}
```

A. 2,0 3,0

B. 2,1 3,0

C. 2,0 2,1

D. 2,1 0,1

15、以下 C 程序, 正确的运行结果是 (C)。(注□表示空格)

```
void main()
```

```
{ long y=-34567;
```

```
  printf("y=%-8ld\n",y);
```

```
  printf("y=%-08ld\n",y);
```

```
  printf("y=%08ld\n",y);
```

```
  printf("y=%+8ld\n",y);
```

```
}
```

A. y=□□-34567

```
y=-□□34567
```

```
y=-0034567
```

```
y=-34567
```

C. y=-34567

```
y=-34567
```

```
y=-34567
```

```
y=-0034567
```

```
y=□□-34567
```

B. y=-34567

```
y=-34567
```

```
y=-0034567
```

```
y=-□34567
```

D. y=□□-34567

```
y=-0034567
```

```
y=-34567
```

```
y=00034567
```

```
y=-34567
```

16、在 scanf 函数的格式控制中, 格式说明的类型与输入项的类型应该一一对应匹配, 如果类型不匹配, 系统将 (B)。

A. 不予接收

B. 并不给出出错信息, 但不可能得到正确数据

C. 能接收到正确输入

D. 给出出错信息, 不予接收输入

17、以下程序的输出结果是 (B)。

```
void main()
{   char c1='6',c2='0';
    printf("%c,%c,%d,%d\n",c1,c2,c1-c2,c1+c2);
}
```

- A. 因输出格式不合法, 输出出错信息 B. 6, 0, 6, 102
C. 6, 0, 7, 6 D. 6, 0, 5, 7

18、C语言中调用 printf 函数进行输出时需要注意, 在格式控制串中, 格式说明与输出项的个数必须相同。如果格式说明的个数小于输出项的个数, 多余的输出项将 (A); 如果格式说明的个数多于输出项的个数, 则对于多余的格式将输出不定值 (或 0)。

- A. 不予输出 B. 输出空格 C. 照样输出 D. 输出不定值或 0

19、以下程序的输出结果是 (D)。

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

- A. 10 10 B. 9 9 C. 9 10 D. 10 9

20、下列说法正确的是 (D)。

- A. 输入项可以是一个实型常量, 如 scanf("%f",4.8);
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("%5.3f",&f);
D. 当输入数据时, 必须指明变量的地址, 如 scanf("%f",&f);

21、若 a、b、c、d 都是 int 类型变量且初值为 0, 以下选项中不正确的赋值语句是 (C)。

- A. a=b=c=100; B. d++;
C. c+b; D. d=(c=22)-(b++);

22、选项中不是 C 语句的是 (C)。

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

23、合法的 C 语言赋值语句是 (B)。

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

24、下面的程序:

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

其输出结果是 (C)。

- A. 0 B. 1 C. 3 D. 不确定的值

25、若变量已正确说明为 int 类型, 要从键盘给 a、b、c 输入数据, 正确的输入语句是 (D)。

- A. read(a,b,c); B. scanf("%d%d%d",a,b,c);
C. scanf("%D%D%D",&a,&b,&c); D. scanf("%d%d%d",&a,&b,&c);

26、若变量已正确说明为 float 类型, 要通过输入语句: scanf("%f%f%f",&a,&b,&c); 给 a 赋予 10.0、b 赋予 22.0、c 赋予 33.0, 以下不正确的输入形式是 (B)。

- A. 10 B. 10.0, -22.0, 33.0

22

33

C. 10.0

D. 10 22

22.0 33.0

33

27、若变量已正确定义,要将 a 和 b 中的数进行交换,下面选项中不正确的语句组是 (C)。

A. a=a+b,b=a-b,a=a-b

B. t=a,a=b,b=t;

C. a=t;t=b;b=a;

D. t=b;b=a;a=t;

28、若变量已正确定义,以下程序段:

x=5.16894;

printf("%f\n",(int)(x*1000+0.5)/(float)1000);

的输出结果是 (D)。

A. 输出格式说明与输出项不匹配,输出无定值

B. 5.170000

C. 5.168000

D. 5.169000

29、若有以下程序段:

int c1=1,c2=2,c3;

c3=c1/c2;

c3 中的值是 (A)。

A. 0

B. 1/2

C. 0.5

D. 1

30、若有以下程序段:

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

c=(a--a-5),(a=b,b+3);

printf("%d,%d,%d\n",a,b,c);

其输出结果是 (B)。

A. 3, 0, -10

B. 0, 0, 5

C. -10, 3, -10

D. 3, 0, 3

31、按以下由 if 和 goto 所构成的循环程序段改写成 for 语句循环程序段是 (B)。

int i=1,s=0;

begin: if(i<=10) { s+=i; i++; goto begin; }

printf("s=%4d\n",s);

A. i=1;s=0;

B. int i=1;s=0;

for(i=1;i<=10;i++) s+=i;

for(i=1;i<=10;i++) s+=i;

printf("s=%4d\n",s);

printf("s=%4d\n",s);

C. int i=1;s=1;

D. i=1;s=1;

for(i=1;i<=10;i++) s+=i;

for(i=1;i<=10;i++) s+=i;

printf("s=%4d\n",s);

printf("s=%4d\n",s);

二、填空题

1、以下程序段:

int x=0177;

printf("x=%3d,x=%6d,x=%6o,x=%6x,x=%6u\n",x,x,x,x,x);

其输出结果是 x=127,x=000127,x=000177,x=00007f,x=000127。

2、以下程序段:

```
int x=0177;
```

```
printf("x=%-3d,x=%-6d,x=$%-06d,x=$%06d,x=%06d\n",x,x,x,x,x);
```

其输出结果是 x=127,x=127□□□,x=\$127□□□,x=\$000127,x=%06d。

3、以下程序段:

```
double a=513.789215;
```

```
printf("a=%8.6f,a=%8.2f,a=%14.8f,a=%14.8lf\n",a,a,a,a);
```

其输出结果是 a=513.789215,a=□□513.79,a=□□513.78921500,a=□□513.78921500。

4、对不同类型的数据有不同的格式字符,例如: %d 格式字符是用来输出十进制整数, %c 格式字符是用来输出一个字符, %s 格式字符是用来输出一个字符串。

5、符号 "&" 是 取地址 运算符, &a 是指 a 在内存中的地址。

6、putchar 函数的作用是 向终端输出一个字符。

7、阅读程序:

```
void main()
```

```
{ char str1[]="people and computer",str2[10];
```

```
scanf("%s",str2);
```

```
printf("%s",str2);
```

```
printf("%s\n",str1);
```

```
}
```

运行上面的程序,输入字符串 PEOPLE AND COMPUTER,则程序的输出结果是:

PEOPLEpeople and computer。

8、“%-ms”表示如果串长 小于 m,则在 m 列范围内,字符串向 左 靠, 右 补空格。

9、若有以下定义,请写出以下程序段输出语句执行后的输出结果 (1) -200, 2500 (2) i=-200, j=2500 (3) i=-200。

```
int i=-200,j=2500;
```

```
printf("(1)%d,%d\n",i,j);
```

```
printf("(2)i=%d,j=%d\n",i,j);
```

```
printf("(3)i=%d\nj=%d\n",i,j);
```

10、变量 i、j、k 已定义为 int 类型并有初值 0,用以下语句进行输入

```
scanf("%d",&i);scanf("%d",&j);scanf("%d",&k);
```

当执行以上输入语句,从键盘输入: 12.3<CR>

则变量 i、j、k 的值分别是 12、0、0。

11、复合语句在语法上被认为是 一条语句。空语句的形式是 ;。

12、以下程序段:

```
int k; float a; double x;
```

```
scanf("%d%f%lf",&k,&a,&x);
```

```
printf("k=%d,a=%f,x=%f\n",k,a,x);
```

要求通过 scanf 语句给变量赋值,然后输出变量的值。写出运行时给 k 输入 100,给 a 输入 25.82,给 x 输入 1.89234 时的三种可能的输入形式:

(1) 100 25.82 1.89234

(2) 100<回车符>

25.82<回车符>

1.89234<回车符>

(3) 100<制表符>25.82<制表符>1.89234

三、编程题

1、编写程序，读入3个双精度数，求它们的平均值并保留此平均值小数点后一位数，对小数点后第二位数进行四舍五入，最后输出结果。

答案：编程如下：

```
#include<stdio.h>
void main()
{
    double a,b,c,ave;
    printf("Enter a,b,c: ");
    scanf("%lf%lf%lf",&a,&b,&c);
    ave=(a+b+c)/3;
    printf("ave=%f\n",ave);
    ave=(int)((ave+0.05)*10)/10.0;
    printf("a=%f,b=%f,c=%f,ave=%f\n",a,b,c,ave);
}
```

2、编写程序，读入3个整数给a、b、c，然后把a中原来的值给b，把b中原来的值给c，把c中原来的值给a。

【程序】

```
#include<stdio.h>
void main()
{
    int a,b,c,t;
    printf("Enter a,b,c:\n");
    scanf("%d%d%d",&a,&b,&c);
    printf("(1) a=%d,b=%d,c=%d\n",a,b,c);
    t=c;c=b;b=a;a=t;
    printf("(2) a=%d,b=%d,c=%d\n",a,b,c);
}
```

第4部分 选择结构

一、选择题

1、以下程序段:

```
void main()
{   int k=-3;
    if(k<=0) printf("****\n")
    else printf("&&&&\n");
}
```

其输出结果为 (D)。

- A. ##### B. &&&&
C. #####&&&& D. 由语法错误不能通过编译

2、若 a 和 b 均是正整型变量, 正确的 switch 语句是 (D)。

A. switch(pow(a,2)+pow(b,2))
 { case 1: case 3: y=a+b; break;
 case 0: case 5: y=a-b;
 }

B. switch(a*a+b*b);
 { case 3:
 case 1: y=a+b; break;
 case 0: y=b-a; break;
 }

C. switch a
 { default: x=a+b;
 case 10: y=a-b; break;
 case 11: x=a*d; break;
 }

D. switch(a+b)
 { case 10: x=a+b; break;
 case 11: y=a-b; break;
 }

3、以下程序的输出为 (A)。

```
int a,b,c;
a=10,b=50,c=30;
if(a>b) a=b,b=c;
c=a;
printf("a=%d,b=%d,c=%d",a,b,c);
```

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

4、对下述程序, (A) 是正确的判断。

```
#include<stdio.h>
void main()
{   int x,y;
```

```
scanf("%d,%d",&x,&y);
if(x>y)
    x=y,y=x;
else
    x++;y++;
printf("%d,%d",x,y);
}
```

- A. 由语法错误, 不能通过编译
 B. 若输入数据 3 和 4, 则输出 4 和 5
 C. 若输入数据 4 和 3, 则输出 3 和 4
 D. 若输入数据 4 和 3, 则输出 4 和 4
- 5、下面程序的输出结果是 (B)。

```
#include<stdio.h>
void main()
{
    int x=100,a=20,b=10;
    int v1=5,v2=0;
    if(a<b)
        if(b!=15)
            x=1;
        else
            if(v2) x=10;
    x=-1;
    printf("%d",x);
}
```

- A. 100 B. -1 C. 1 D. 10

6、执行下面语句后的输出结果为 (D)。

```
int i=-1;
if(i<=0) printf("****\n");
else printf("%%%%%%%%\n")
```

- A. **** B. %%%%
 C. %%%%c D. 由语法错误, 不能正确执行

7、下述程序段中, 正确的是 (D)。

```
A. int x=0,y=10;
    switch(x)
    {
        case y: x++; break;
        case y+2: x+=10; break;
        case y-1: x=-7; break;
    }
```

```
B. int x=0,y;
    switch(x)
    {
        case x>0: y=1; break;
        case x==0: y=0; break;
        case x<0: y=-1; break;
    }
```

C. #define y 20

```
int x=10;
switch(x)
{ case 12: z=3; break;
  case y+1: x+=10; break;
  case y-8: y-=3; break;
}
```

D. int x=0,y;

```
switch(x)
{ case 3:
  case -1: y=2; break;
  case 2: break;
}
```

8. 执行程序时输入“ADescriptor”，以下程序的输出结果是（ D ）。

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

```
void main()
```

```
{ char c;
  int v0=0,v1=0,v2=0;
  do
  { switch(c=getchar())
    { case 'a': case 'A':
      case 'e': case 'E':
      case 'i': case 'I':
      case 'o': case 'O':
      case 'u': case 'U': v1++;
      default: v0++;v2++;
    }
  }
  while(c!='\n');
  printf("\nv0=%d,v1=%d,v2=%d\n",v0,v1,v2);
}
```

A. v0=7,v1=4,v2=7

B. v0=8,v1=4,v2=8

C. v0=11,v1=4,v2=11

D. v0=12,v1=4,v2=12

9. 以下程序的输出结果是（ B ）。

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

```
void main()
```

```
{ int x=1,y=0,a=0,b=0;
  switch(x)
  { case 1: switch(y)
    { case 0: a++; break;
      case 1: b++; break;
    }
    case 2: a++; b++; break;
    case 3: a++; b++;
  }
}
```

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

A. a=1,b=0

B. a=2,b=1

C. a=1,b=1

D. a=2,b=2

10、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（ A ）。

```
void 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

11、以下 if 语句书写正确的是（ B ）。

A. if(x=0;) printf("%f",x);
else printf("%f",-x);

B. if(x>0) { x=x+1; printf("%f",x); }
else printf("%f",-x);

C. if(x>0); { x=x+1; printf("%f",-x); }
else printf("%f",-x);

D. if(x>0) { x=x+1; printf("%f",x) }
else printf("%f",-x);

12、逻辑运算符运算对象的数据类型为（ D ）。

A. 只能是0或1

B. 只能是 .T. 或 .F.

C. 只能是整型或字符型

D. 任何类型的数据

13、下面程序的执行结果为（ A ）。

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

```
void main()
```

```
{ int x=1,y=1,z=0;
  if(z<0)
    if(y>0) x=3;
    else x=5;
  printf("%d\t",x);
  if(z=y<0) x=3;
  else if(y==0) x=5;
    else x=7;
  printf("%d\t",x);
  printf("%d\t",z);
}
```

A. 1 7 0

B. 3 7 0

C. 5 5 0

D. 1 5 1

14、下面程序的执行结果为（ D ）。

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

```
void main()
```

```
{ int x,y=1;
  if(y!=0) x=5;
```

```
printf("%d\t",x);
if(y==0) x=3;
else x=5;
printf("%d\t\n",x);
}
```

- A. 1 3 B. 1 5 C. 5 3 D. 5 5

15、若有以下变量定义: float x; int a,b; 则正确的 switch 语句是 (D)。

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 2*a: printf("**\n");
}
```

D. switch(a+b)

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

16、假定所有变量均已正确说明, 下列程序段运行后 x 的值是 (B)。

```
a=b=c=0; x=35;
```

```
if(!a) x=-1;
```

```
else if(b);
```

```
if(c) x=3;
```

```
else x=4;
```

- A. 34 B. 4 C. 35 D. 3

17、若有以下函数关系:

$$y = \begin{cases} 2x & (x < 0) \\ x & (x > 0) \\ x+1 & (x = 0) \end{cases}$$

下面程序段能正确表示以上关系的是 (C)。

A. y=2x;

```
if(x!=0)
```

```
if(x>0) y=x;
```

```
else y=x+1;
```

B. y=2x;

```
if(x<=0)
```

```
if(x==0) y=x+1;
```

```
else y=x;
```

C. `if(x>=0)`
 `if(x>0) y=x;`
 `else y=x+1;`
 `else y=2*x;`

D. `y=x+1;`
 `if(x<=0)`
 `if(x<0) y=2x;`
 `else y=x;`

18、以下程序的运行结果是 (A)。

```
void main()
{
    int c,x,y;
    x=y=c=0;
    c=x++&& y++;
    printf("\n%d %d %d\n",x,y,c);
}
```

A. 100 B. 211 C. 011 D. 001

19、下列运算符中优先级最高的是 (B)。

A. > B. + C. && D. !=

20、下列运算符中优先级最低的算符是 (A)。

A. || B. != C. <= D. +

21、为表示关系: $x \geq y \geq z$, 应使用的 C 语言表达式是 (A)。

A. `(x>=y)&&y>=z)` B. `(x>=y)AND(y>=z)`

C. `(x>=y>=z)` D. `(x>=y)&(y>=z)`

22、设 a、b 和 c 都是 int 型变量, 且 a=3, b=4, c=5, 则以下的表达式中, 值为 0 的表达式是 (D)。

A. `a&& b` B. `a<=b`
 C. `a||b+c&&b-c` D. `!((a<b)&&!c||1)`

23、以下程序输出结果是 (C)。

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

A. 0 B. 1 C. 2 D. 3

24、以下程序的输出结果是 (A)。

```
void main()
{
    int w=4,x=3,y=2,z=1;
    printf("%d\n",(w<x?w:z<y?z:x));
}
```

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

25、若执行以下程序:

```
void main()
```

```

{   int a,b,s;
    scanf("%d%d",&a,&b);
    s=a;
    if(a<b) s=b;
    s*=s;
    printf("%d\n",s);
}

```

从键盘上输入 3 和 4, 则输出结果是 (B)。

A. 14 B. 16 C. 18 D. 20

二、填空题

1、以下两条 if 语句合并成一条 if 语句为:

```

if(a<=b) { x=1; printf("###x=%d\n",x); }
else { y=2; printf("***y=%d\n",y); }

```

[代码]

```

if(a<=b) x=1;
else y=2;
if(a>b) printf("***y=%d\n",y);
else printf("###x=%d\n",x);

```

2、在 C 语言中, 对于 if 语句, else 子句与 if 子句的配对约定是 else 与它前面最近的未配对的一个 if 配对。

3、以下程序的功能是计算一元二次方程 $ax^2+bx+c=0$ 的根, 补足程序中空缺的语句。

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

```
#include<math.h>
```

```
void main()
```

```

{   float a,b,c,_abs,_derta,_doublea,part1,part2;
    printf("enter a,b,c:");
    scanf("%f%f%f",&a,&b,&c);
    if( a==0 )
        if( b==0 ) printf("no answer due to input error\n");
        else printf("the single root is %f\n",-c/b);
    else
    {   _derta=b*b-4*a*c;
        _doublea=2*a;
        part1=-b/(2*a);
        _abs=abs(_derta);
        part2=sqrt(_abs)/_doublea;
        if( _derta<0 )
            printf("complex root\n real part=%f image part=%f\n",part1,part2);
        else
            printf("real roots\n root1=%f root2=%f\n",part1+part2,part1-part2)
    }
}

```

4、以下程序段的运行结果是: x=1 y=0


```
int x=1,y=0;
switch(x)
{
    case 1: switch(y)
        {
            case 0: printf("x=1 y=0\n"); break;
            case 1: printf("y=1\n"); break;
        }
    case '2': printf("x=2\n");
}
```

5、设有程序段如下：

```
switch(class)
{
    case 'A': printf("GREAT!\n");
    case 'B': printf("GOOD!\n");
    case 'C': printf("OK!\n");
    case 'D': printf("NO!\n");
    default : printf("ERROR!\n");
}
```

若 class 的值为'C'，则输出结果是 OK!

NO!

ERROR!

第5部分 循环结构

一、选择题

1、以下程序:

```
void main()
{   int x,i;
    for(i=1;i<=100;i++)
    {   x=i;
        if(++x%2==0)
            if(++x%3==0)
                if(++x%7==0)
                    printf("%d ",x);
    }
    printf("\n");
}
```

其输出结果是 (D)。

A. 39 81 B. 42 84 C. 25 68 D. 28 70

2、设 i 和 x 都是 int 类型, 则 for 循环语句 (A)。

```
for(i=0,x=0;i<=9&&x!=876;i++) scanf("%d",&x);
```

A. 最多执行 10 次 B. 最多执行 9 次
C. 是无限循环 D. 循环体一次也不执行

3、下述 for 循环语句 (B)。

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

A. 判断循环结束的条件非法 B. 是无限循环
C. 只循环一次 D. 一次也不循环

4、下述循环的循环次数是 (B)

```
int k=2;
while(k=0) printf("%d",k);
k--; printf("\n");
```

A. 无限次 B. 0 次 C. 1 次 D. 2 次

5、对于 (1), (2) 两个循环语句, (A) 是正确的描述。

(1) while(1);

(2) for(;;);

A. (1) (2) 都是无限循环 B. (1) 是无限循环, (2) 错误
C. (1) 循环一次, (2) 错误 D. (1) (2) 皆错误

6、下述程序的输出结果是 (B)。

```
#include<stdio.h>
void main()
{   int i,j,x=0;
    for(i=0;i<2;i++)
    {   x++;
        for(j=0;j<=3;j++)
```

```

    {   if(j%2) continue;
        x++;
    }
    x++;
}
printf("x=%d\n",x);
}

```

A. x=4 B. x=8 C. x=6 D. x=12

7、在下述程序中，判断 $i > j$ 共执行了 (D)。

```

#include<stdio.h>
void main()
{   int i=0,j=10,k=2,s=0;
    for(;;)
    {   i+=k;
        if(i>j)
        {   printf("%d ",s);
            break;
        }
        s+=i;
    }
}

```

A. 4 B. 7 C. 5 D. 6

8、若下述程序运行时输入的数据是：“3.6, 2.4, 1.5”，则输出结果是 (B)。

```

#include<stdio.h>
#include<math.h>
void 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",y);
}

```

A. 1.500000 B. 1.600000 C. 2.000000 D. 2.400000

9、以下程序的输出结果是 (A)。

```

#include<stdio.h>
void main()
{   int i;
    for(i=1;i<=5;i++)
    {   if(i%2) putchar('<');

```

```

        else continue;
        putchar('>');
    }
    putchar('#');
}

```

- A. $\diamond\diamond\diamond\#$ B. $>\diamond\diamond<\#$
 C. $\diamond\diamond\#$ D. $>\diamond<\#$

10、下述程序的输出结果是 (B)。

```

#include<stdio.h>
void main()
{
    int k=0,m=0;
    int i,j;
    for(i=0;i<2;i++)
    {
        for(j=0;j<3;j++)
            k++;
        k=j;
    }
    m=i+j;
    printf("k=%d,m=%d",k,m);
}

```

- A. k=0,m=3 B. k=0,m=5
 C. k=1,m=3 D. k=1,m=5

11、while 循环语句中，while 后一对圆括号中的表达式的值决定了循环体是否进行，因此进入 while 循环后，一定要有能使此表达式的值变为 (A) 的操作，否则循环将会无限制地进行下去。

- A. 0 B. 1 C. 成立 D. 2

12、下面程序的输出结果是 (A)。

```

void main()
{
    int n=0;
    while(n++<=1)
        printf("%d\t",n);
    printf("%d\n",n);
}

```

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

13、下面程序的输出结果是 (C)。

```

void main()
{
    char s[5]="121";
    int k=0,a=0,b=0;
    do
    {
        k++;
        if(k%2==0)
        {
            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

14、当输入为“quert?”时，下面程序的执行结果是（ A ）。

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

```
void main()
```

```

{   char c;
    c=getchar();
    while(c!='?')
    {   putchar(c);
        c=getchar();
    }
}

```

A. quert

B. Rvfsu

C. quert?

D. rvfsu?

15、在 do...while 循环中，循环由 do 开始，用 while 结束。必须注意的是：在 while 表达式后面的（ C ）不能丢，它表示 do...while 循环的结束。

A. 0

B. 1

C. ;

D. ,

16、若 i, j 已定义为 int 型，则以下程序段中循环体总的执行次数是（ A ）。

```

for(i=5;i>0;i--)
    for(j=0;j<4;j++)
    {   .....   }

```

A. 20

B. 24

C. 25

D. 30

17、分析下列程序：

```

void 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]);
}

```

上述程序运行后，输出结果是（ D ）。

A. 2

B. 1

C. 0

D. 3

18、在 C 语言的循环语句 for、while、do...while 中，用于直接中断最内层循环的语句是（ C ）。

A. switch

B. continue

C. break

D. if

19、执行下面的程序后，a 的值为（ B ）。

```

void 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

20、以下程序段：

```

int n=10;
while(n>7)
{
    n--;
    printf("%d\n",n);
}

```

其输出结果是 (B)。

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

21、以下程序：

```

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

```

其输出结果是 (C)。

A. 15 B. 14 C. 不确定 D. 0

22、若 x 是 int 型变量，以下程序段：

```

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

```

其输出结果是 (D)。

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

23、以下叙述正确的是 (D)。

- 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. 用 do...while 语句构成循环时，只有在 while 后的表达式为非零时结束循环
- D. 用 do...while 语句构成循环时，只有在 while 后的表达式为零时结束循环

24、以下程序段：

```

int k,j,s;
for(k=2;k<6;k++,k++)

```

```
{ s=1;
  for(j=k;j<6;j++) s+=j;
}
```

printf("%d\n",s);

其输出结果是 (D)。

A. 9 B. 1 C. 11 D. 10

25、以下程序段:

```
int i,j,m=0;
for(i=1;i<=15;i+=4)
  for(j=3;j<=19;j+=4) m++;
printf("%d\n",m);
```

其输出结果是 (C)。

A. 12 B. 15 C. 20 D. 25

26、以下程序段:

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

其输出结果是 (C)。

A. 1 B. 3 0 C. 1 -2 D. 死循环

27、以下程序:

```
void main()
{ int y=10;
  for(;y>0;y--)
    if(y%3==0)
    { printf("%d",--y);
      continue;
    }
}
```

其输出结果是 (B)。

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

二、填空题

1、当执行以下程序段后, i 的值是 5, j 的值是 4, k 的值是 6。

```
int a,b,c,d,i,j,k;
```

```
a=10;b=c=d=5;i=j=k=0;
```

```
for(;a>b;++b) i++;
```

```
while(a>++c) j++;
```

```
do k++; while(a>d++);
```

2、以下程序的输出结果是 -1。

```
void main()
```

```
{ int x=2;
```

```
  while(x--);
```

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

}

3、有以下程序段：

s=1.0;

for(k=1;k<=n;k++) s=s+1.0/(k*(k+1));

printf("%f\n",s);

请填空，使下面的程序段的功能完全与之等同。

s=0.0;

____ d=1.0 ____;

k=0;

do

{ s=s+d;

____ k++ ____;

d=1.0/(k*(k+1));

}

while(____ k<=n ____);

printf("%f\n",s);

4、下列是计算 $n!$ 的值的程序，请填空：

void main()

{ int i,s,n;

s=1;

printf("Enter n:");

scanf("%d",&n);

for(i=1;i<=n;i++)

{ ____ s*=i ____ ; }

printf("s=%d",s);

}

5、以下程序段的输出结果是 无限循环，没有输出结果。

int k,n,m;

n=10;m=1;k=1;

while(k<=n) m*=2;

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

6、continue 语句的作用是 结束本次循环执行，即跳过循环体中下面尚未执行的语句，接着进行下一次是否执行循环的判断。7、while 语句的特点是 先判断条件是否成立（为真），若成立则继续执行循环，若不成立，则结束循环。do...while 语句的特点是 至少执行一次循环。8、在循环中，continue 语句与 break 语句的区别是：continue 语句只是 结束本次循环，后进行循环的条件判断，即不终止整个循环的执行过程。break 语句是 终止整个循环的执行，不再进行条件判断。

9、将 for（表达式 1；表达式 2；表达式 3）语句改为等价的 while 语句为：

表达式 1；

while（表达式 2）

表达式 3；

10、下面程序的输出结果是 1, 2, 3。

void main()


```

{   int n=0;
    while(n++<=1) printf("%d",n);
    printf("%d\n",n);
}

```

11、以下程序段的输出结果是 11。

```

int i=0,sum=1;
do { sum+=i++; } while(i<5);
printf("%d\n",sum);

```

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

```

#include<stdio.h>
void main()
{   float x,amax,amin;
    scanf("%f",&x);
    amax=x; amin=x;
    while( x>=0 )
    {   if(x>amax) amax=x;
        if( x<amin ) amin=x;
        scanf("%f",&x);
    }
    printf("namax=%f\namin=%f\n",amax,amin);
}

```

第6部分 字符型数据

一、选择题

1、对以下程序:

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

```
void main()
```

```
{ int c;
```

```
while((c=getchar())!='\n')
```

```
{ switch(c-'2')
```

```
{ case 0:
```

```
case 1: putchar(c+4);
```

```
case 2: putchar(c+4); break;
```

```
case 3: putchar(c+3);
```

```
default: putchar(c+2); break;
```

```
}
```

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

```
}
```

从第一列开始输入数据: 2473<CR>(<CR>代表回车), 程序的输出结果是 (A)。

A. 668977

B. 668966

C. 66778777

D. 6688766

2、C语言中不合法的字符串常量是 (B)。

A. "121"

B. 'y='

C. "\n\n"

D. "ABCD\x6d"

3、以下程序

```
void main()
```

```
{ char c1='A',c2='Y';
```

```
printf("%d,%d\n",c1,c2);
```

```
}
```

其输出结果是 (D)。

A. 因输出格式不合法看, 无正确输出

B. 65, 90

C. A, Y

D. 65, 89

4、以下程序

```
void main()
```

```
{ char ch1,ch2;
```

```
ch1='A'+5-'3';
```

```
ch2='A'+5-'3';
```

```
printf("%d,%c\n",ch1,ch2);
```

```
}
```

其输出结果是 (A)。

A. 67,C

B. B,C

C. C,D

D. 不确定的值

5、设有定义语句: char a='\72'; 则变量 a (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

6. C 语言中不合法的字符常量是 (A)。

- A. '\0xff' B. '\65' C. '&' D. '\028'

7. 判断 char 型变量 c 是否为大写字母的最简单且正确的表达式是 (D)。

- A. 'A'<=c<='Z' B. (c>='A')&(c<='Z')
C. ('A'<=c)AND('Z'>=c) D. (c>='A')&&(c<='Z')

8. 以下程序段:

```
void main()
{
    char x='A';
    x=(x>='A'&&x<='Z')?(x+32):x;
    printf("%c\n",x);
}
```

其输出结果是 (B)。

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

9. C 语言中, 字符(char)型数据在微机内存中的存储形式是 (D)。

- A. 反码 B. 补码
C. EBCDIC 码 D. ASCII 码

10. 已知字母 A 的 ASCII 码值是 65, 以下程序 (C)。

```
#include<stdio.h>
void main()
{
    char a='A';
    int b=20;
    printf("%d,%o",(a=a+a,a+b,b),a+'a'-'A',b);
}
```

- A. 表达式非法, 输出零或不确定值
B. 因输出项过多, 无输出或输出不确定值
C. 输出结果为 20, 141
D. 输出结果为 20, 141, 20

11. 若 ch 为 char 型变量, k 为 int 型变量 (已知字符 a 的 ASCII 码是 97), 则执行下列语句后的输出为 (C)。

```
ch='a';
k=12;
printf("%x,%o",ch,ch,k);
printf("k=%d\n",k);
```

- A. 因变量类型与格式描述符的类型不匹配, 输出无定值
B. 输出项与格式描述符个数不符, 输出为 0 值或不定值
C. 61, 141, k=%d
D. 62, 141, k=%12

12. 已知字母 a 的 ASCII 码的十进制代码为 97, 则执行下列语句后输出为 (C)。

```
ch='a';
a--;
printf("%d,%c\n",a+'2'-'0',a+'3'-'0');
```

- A. a, c B. a--运算不合法, 故有语法错误

C. 98, c D. 格式描述和输出项不匹配, 输出无定值

13、以下程序的输出结果是 (D)。

```
#include<stdio.h>
void main()
{   int x=02,y=3;
    printf("x=%d,y=%d",x,y);
}
```

- A. x=2,y=3 B. x=%2,y=%3
C. x=%d,y=%d D. x=%d,y=%d

14、下列程序的输出为 (A)。

```
#include<stdio.h>
void main()
{   int m=7,n=4;
    float a=38.4,b=6.4,x;
    x=m/2+n*a/b+1/2;
    printf("%f\n",x);
}
```

- A. 27.000000 B. 27.500000
C. 28.000000 D. 28.500000

15、在 C 语言中, 字符型数据在内存中以 (D) 形式存放。

- A. 原码 B. BCD 码
C. 反码 D. ASCII 码

16、(B) 是非法的 C 语言转义字符。

- A. '\b' B. '\0xf' C. '\037' D. '\w'

17、语句 char s='\092'; 的作用是 (B)。

- A. 使 s 包含一个字符 B. 说明不合法, s 的值不定
C. 使 s 包含 4 个字符 D. 使 s 包含 3 个字符

18、字符串 "\w"ABC\w" 的长度是 (B)。

- A. 11 B. 7 C. 5 D. 3

19、有关下述语句输出的判断中, 正确的是 (C)。

```
char x='\xe0';
printf("%d",x);
```

- A. 赋值非法 B. 输出值不确定
C. 输出值为-32 D. 输出值为 224

二、填空题

1、EOF 的十进制值是 -1。

2、调用 C 语言提供的对字符进行处理的库函数时, 在 #include 命令中应包含的头文件是 ctype.h。

3、若变量已正确定义, 要通过以下语句给 a、b、c、d 分别输入字符 A、B、C、D, 给 w、x、y、z 分别输入整数 10、20、30、40, 正确的输入形式是: (请用 <CR> 代表回车)

```
scanf("%d%c%d%c%d%c%d%c",&w,&a,&x,&b,&y,&c,&z,&d);
```

答案:

10A 20B 30C 40D<CR>

或答案:

10A<CR>

20B<CR>

30C<CR>

40D<CR>

4、若有以下定义和输入语句:

```
char c1,c2,c3;
scanf("%4c",&c1);
scanf("%4c",&c2);
scanf("%4c",&c3);
```

要求给 c1、c2、c3 输入字符 A、B 和 C, 则从键盘上正确输入数据的形式为:
从第一列开始, 输出形式是:

A000B000C000<CR>

5、C 语言中, 字符占内存 1 个字节。

6、下列语句的输出结果是 31, 37, 1f, 31。

```
char a=31; printf("%d,%o,%x,%u\n",a,a,a,a);
```

7、阅读下述程序, 说明其输出结果。请填空:

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

```
void main()
```

```
{ int a=3,b=4;
```

```
printf("%d\n",a=a+1,b+a,b+1); 输出结果 4
```

```
printf("%d\n",(a=a+1,b+a,b+1)); 输出结果 5
```

```
}
```

8、char 型常量在内存中存放的是 ASCII 码值。

9、'\72' 在内存中占 1 个字节, "72" 在内存中占 2 个字节。

10、假设所有的变量均为整型, 则以下两组语句的输出结果分别是 8、%d。

(1) a=3+5,a*4; printf("%d\n",a);

(2) x=11/3; printf("%d\n",x);

11、有以下定义和输入语句, 要求给 c1、c2 输入字符 A 和 B, 给 a1、a2 输入 7.29 和 101.298,
从键盘上正确输入数据的形式是: ?

```
char c1,c2;
```

```
float a1,a2;
```

```
scanf("%f%f",&a1,&a2);
```

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

答案: 7.29 101.298AB<CR>

或: 7.29<CR>

101.298AB<CR>

第7部分 函数

一、选择题

1、在C语言中,若对函数类型未加显式说明,则函数的隐含类型是(C)。

A. void B. double C. int D. char

2、下述函数段中,(C)含有错误。

A. int f1()

```
{ int x;
    scanf("%d",&x);
    return x++,1;
}
```

B. int f2()

```
{ return (x>1?printf("R"):putchar('r'));
}
```

C. main()

```
{ float x=3.5;
    float y=fmax(2.0,3.0)=x++;
    printf("%f",x+y);
}
```

D. void Rep(int *x,int *y)

```
{ int t=*x;
    *x=*y;
    *y=t;
}
```

3、下述程序段的输出结果是(A)。

```
int x=10;
int y=x++;
printf("%d,%d",(x++,y),y++);
```

A. 11,10 B. 11,11 C. 10,10 D. 10,11

4、下述程序段的输出结果是(B)。

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

```
void prt(int *x,int *y,int *z)
```

```
{ printf("%d,%d,%d\n",++*x,++*y,*z++); }
```

```
void main()
```

```
{ int a=10,b=40,c=20;
    prt(&a,&b,&c);
    prt(&a,&b,&c);
}
```

A. 11,42,31

B. 11,41,20

12,22,41

12,42,20

C. 11,21,40

D. 11,41,21

11,21,41

12,42,22

5、对于下述程序,判断(A)是正确的。

```
#include<stdio.h>
void f(int *p)
{ *p=10; }
void main()
{ int *p;
  f(p);
  printf("%d",(*p)++);
}
```

- A. 输出的值是随机值 B. 因输出语句错误而不能执行
C. 输出值为 0 D. 输出值为 11

6、下述程序的输出结果是 (C)。

```
#include<stdio.h>
long fun(int n)
{ long s;
  if(n==1||n==2)
    s=2;
  else
    s=n+fun(n-1);
  return s;
}
```

```
void main()
{ printf("\n%d",fun(4)); }
```

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

7、下述程序的输出结果是 (A)。

```
#include<stdio.h>
int fun(int x)
{ int p;
  if(x==0||x==1)
    return 3;
  else
    p=x-fun(x-2);
  return p;
}
```

```
void main()
{ printf("\n%d",fun(9)); }
```

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

8、以下函数

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

的类型是 (C)。

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

9、以下程序:

```
func(int a,int b)
```

```

{   int c;
    c=a+b;
    return c;
}

void main()
{   int x=6,y=7,z=8,r;
    r=func((x--,y++,x+y),z--);
    printf("%d\n",r);
}

```

其输出结果是 (C)。

- A. 11 B. 20 C. 21 D. 31

10、下述程序的运行结果为 (A)。

```

#include<stdio.h>
void abc(char *str)
{   int a,b;
    for(a=b=0;str[a]!='\0';a++)
        if(str[a]!='c')
            {   str[b++]=str[a];   }
    str[b]='\0';
}

void main()
{   char str[]="abcdef";
    abc(str);
    printf("str[]=%s",str);
}

```

- A. str[]=abdef B. str[]=abcdef
C. str[]=a D. str[]=ab

11、关于建立函数的目的，以下说法正确的是 (B)。

- 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

12、在 C 语言程序中，以下描述正确的是 (B)。

- 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

13、以下程序的正确运行结果是 (D)。

```

#include<stdio.h>
#define MAX 10
int a[MAX],i;
void main()
{   printf("\n");sub1();sub3(a);sub2();sub3(a);   }
sub2()
{   int a[MAX],i,max;
    max=5;
}

```



```

    for(i=0;i<max;i++) a[i]=i;
}
sub1()
{ for(i=0;i<MAX;i++) a[i]=i+i; }
sub3(int a[])
{ int i;
  for(i=0;i<MAX;i++) printf("%d ",a[i]);
  printf("\n");
}

```

- A. 0 2 4 6 8 10 12 14 16 18
0 1 2 3 4
- B. 0 1 2 3 4
0 2 4 6 8 10 12 14 16 18
- C. 0 1 2 3 4 5 6 7 8 9
0 1 2 3 4
- D. 0 2 4 6 8 10 12 14 16 18
0 2 4 6 8 10 12 14 16 18

14、以下说法正确的是 (C)。

- A. 定义函数时，形参的类型说明可以放在函数体内，也可以放在函数体外
- 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类型不一致，以函数值类型为准
- D. 如果形参与实参的类型不一致，以实参类型为准

15、下面函数调用语句含有实参的个数为 (B)。

fun((a,b,c,d),(e,f,g));

- A. 1 B. 2 C. 4 D. 以上都不对

16、如果在一个函数的复合语句中定义了一个变量，则该变量 (A)。

- A. 只在该复合语句中有效，在该复合语句外无效
- B. 在该函数中任何位置都有效
- C. 在本程序的源文件范围内均有效
- D. 此定义方法错误，其变量为非法变量

17、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 (B)。

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

18、以下说法正确的是 (B)。

- 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. 用户如需调用标准库函数，调用前不必使用预编译命令将该函数所在文件包含到用户源文件中，系统自动去调用

19、C语言规定，函数返回值的类型是由 (D)。

- A. return 语句中的表达式类型所决定
- 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决定

20、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 (B)。

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

21、以下程序:

```
fun(int a,int b,int c)
{   c=a*b; }
void main()
{   int c;
    fun(2,3,c);
    printf("%d\n",c);
}
```

其输出结果是 (D)。

- A. 0
- B. 1
- C. 6
- D. 无定值

22、以下叙述中正确的是 (C)。

- A. C 语言程序总是从第一个定义的函数开始执行
- B. 在 C 语言程序中,要调用的函数必须在 main () 函数中定义
- C. C 语言程序总是从 main () 函数开始执行
- D. C 语言程序中的 main () 函数必须放在程序的开始部分

23、以下程序:

```
void main()
{   int i=2,p;
    p=f(i,i+1);
    printf("%d\n",p);
}
int f(int a,int b)
{   int c;
    c=a;
    if(a>b) c=1;
    else if(a==b) c=0;
    else c=-1;
    return (c);
}
```

其输出结果是 (A)。

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

24、以下程序:

```
double f(int n)
{   int i; double s;
    s=1.0;
    for(i=1;i<n;i++) s+=1.0/i;
    return s;
}
void main()
{   int i,m=3; float a=0.0;
```

```

for(i=0;i<m;i++) a+=f(i);
printf("%f\n",a);
}

```

其输出结果是 (A)。

A. 5.500000

B. 3.000000

C. 4.000000

D. 8.25

二、填空题

1、以下程序的输出结果是 12。

```

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

```

```

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

```

2、以下程序的输出结果是 4。

```

fun1(int a,int b)
{
    int c;
    a+=a,b+=b;
    c=fun2(a,b);
    return c*c;
}

```

```

fun2(int a,int b)
{
    int c;
    c=a*b%3;
    return c;
}

```

```

void main()
{
    int x=11,y=19;
    printf("%d\n",fun1(x,y));
}

```

3、下面 pi 函数的功能是根据以下公式返回满足精度 ε 要求的 π 的值，请填空。

$$\frac{\pi}{2} = 1 + \frac{1}{3} + \frac{1}{3} \cdot \frac{2}{5} + \frac{1}{3} \cdot \frac{2}{5} \cdot \frac{3}{7} + \frac{1}{3} \cdot \frac{2}{5} \cdot \frac{3}{7} \cdot \frac{4}{9} + \dots$$

```

double pi(double eps)
{
    double s=0.0,t=1.0;
    int n;
    for( n=1 ; t>eps;n++)
    {
        s+=t;
    }
}

```

```

        t=n*t/(2*n+1);
    }
    return (2.0*s);
}

```

4、以下程序的输出结果是 function running result is:9.0。

```

double sub(double x,double y,double z)
{
    y=1.0;
    z=z+x;
    return z;
}

void main()
{
    double a=2.5,b=9.0;
    printf("function running result is:%6.1f\n",sub(b-a,a,a));
}

```

5、以下程序的运行结果是 the result is: 1

```

        a=3,b=5,x=1
int x;
void main()
{
    int a=3,b=5;
    void swap(int a,int b);
    swap(a,b);
    printf("a=%d,b=%d,x=%d\n",a,b,x);
}

void swap(int a,int b)
{
    int temp;
    temp=a; a=b; b=temp;
    x=a/b;
    printf("the result is:%d\n",x);
}

```

6、若输入的值是-72，以下程序的运行结果是 -72=-2*2*2*3*3。

```

#include<stdio.h>
#include<math.h>
void main()
{
    int n;
    void fun(int n);
    scanf("%d",&n);
    printf("%d=",n);
    if(n<0) printf("-");
    n=fabs(n);
    fun(n);
}

void fun(int n)
{
    int k,r;
    for(k=2;k<=sqrt(n);k++)

```

```

    {   r=n%k;
        while(r==0)
        {   printf("%d",k);
            n=n/k;
            if(n>1) printf("*");
            r=n%k;
        }
    }
    if(n!=1) printf("%d\n",n);
}

```

7、以下程序的运行结果是

1	13	5	7
2	4	26	8
10	1	3	12
the value is 31			

```

#include<stdio.h>
func(int array[][4],int m)
{   int i,j,k;
    k=0;
    for(i=0;i<3;i++)
        for(j=0;j<4;j++)
            if(array[i][j]<m) k=k+array[i][j];
    return k;
}
void main()
{   static int a[3][4]={{1,13,5,7},{2,4,26,8},{10,1,3,12}};
    int i,j,m;
    for(i=0;i<3;i++)
    {   for(j=0;j<4;j++) printf("%4d",a[i][j]);
        printf("\n");
    }
    m=10;
    printf("\nthe value is %d\n",func(a,m));
}

```

第8部分 指针

一、选择题

1、以下程序:

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

其输出结果是 (D)。

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

2、以下程序:

```
void main()
{ int **k,*a,b=100;
  a=&b; k=&a;
  printf("%d\n",**k);
}
```

其输出结果是 (B)。

- A. 运行出错 B. 100 C. a 的地址 D. b 的地址

3、若有定义: int x,*pb; 则正确的赋值表达式是 (A)。

- A. pb=&x B. pb=x
C. *pb=&x D. *pb=*x

4、以下程序:

```
#include "stdio.h"
void main()
{ printf("%d\n",NULL); }
```

其输出结果是 (B)。

- A. 因变量无定义输出不定值 B. 0
C. -1 D. 1

5、以下4个程序中不能对两个整型值进行交换的是 (C)。

A.

```
void main()
{ int a=10,b=20;
  void swap(int *,int *);
  swap(&a,&b);
  printf("%d %d\n",a,b);
}
```

void swap(int *p,int *q)

```
{ int *t,a;
  t=&a;
  *t=*p; *p=*q; *q=*t;
}
```

B.

```

void main()
{   int a=10,b=20;
    void swap(int *,int *);
    swap(&a,&b);
    printf("%d %d\n",a,b);
}

void swap(int *p,int *q)
{   int t;
    t=*p; *p=*q; *q=t;
}

```

C.

```

void main()
{   int *a=0,*b=0;
    void swap(int *,int *);
    *a=10; *b=20;
    swap(a,b);
    printf("%d %d\n",*a,*b);
}

void swap(int *p,int *q)
{   int t;
    t=*p; *p=*q; *q=t;
}

```

D.

```

void main()
{   int a=10,b=20,*x=0,*y=0;
    void swap(int *,int *);
    x=&a; y=&b;
    swap(x,y);
    printf("%d %d\n",a,b);
}

void swap(int *p,int *q)
{   int t;
    t=*p; *p=*q; *q=t;
}

```

6、以下程序:

```

void fun(float *a,float *b)
{   float w;
    *a=*a+*a;
    w=*a;
    *a=*b;
    *b=w;
}

void main()

```

```

{   float x=2.0,y=3.0;
    float *px=&x,*py=&y;
    fun(px,py);
    printf("%2.0f,%2.0f\n",x,y);
}

```

其输出结果是 (C)。

- A. 4, 3 B. 2, 3 C. 3, 4 D. 3, 2

7、以下程序:

```

void sub(float x,float *y,float *z)
{   *y=*y-1.0;
    *z=*z+x;
}

void main()
{   float a=2.5,b=9.0,*pa,*pb;
    pa=&a;   pb=&b;
    sub(b-a,pa,pa);
    printf("%f\n",a);
}

```

其输出结果是 (C)。

- A. 9.000000 B. 1.500000 C. 8.000000 D. 10.500000

8、以下程序:

```

void sub(int x,int y,int *z)
{   *z=y-x; }

void 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);
}

```

其输出结果是 (B)。

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

9、以下程序:

```

void main()
{   int k=2,m=4,n=6;
    int *pk=&k,*pm=&m,*p;
    *(p=&n)=*pk*(*pm);
    printf("%d\n",n);
}

```

其输出结果是 (C)。

- A. 4 B. 6 C. 8 D. 10

10、分析下面的函数

```

void swap(int *p1,int *p2)
{   int *p
    *p=*p1; *p1=*p2; *p2=*p;
}

```


}

则该函数功能为 (D)。

- A. 交换*p1 和*p2 的值
- B. 正确, 但无法改变*p1 和*p2 的值
- C. 交换*p1 和*p2 的地址
- D. 可能造成系统故障, 因为使用了野指针

11、变量 i 的值为 3, i 的地址为 1000, 若欲使 p 为指向 i 的指针变量, 则下列赋值正确的是 (D)。

- A. &i=3
- B. *p=3
- C. *p=1000
- D. p=&i

12、已知下列函数定义:

```
setw(int *b,int m,int n,int dat)
{
    int k;
    for(k=0;k<m*n;k++)
    {
        *b=dat; b++;
    }
}
```

则调用此函数的正确写法是 (假设 a 的说明为 int a[50];) (D)。

- A. setw(*a,5,8,1);
- B. setw(&a,5,8,1);
- C. setw((int *)a,5,8,1);
- D. setw(a,5,8,1);

13、分析下面程序:

```
#include<stdio.h>
void main()
{
    int *p1,*p2,*p;
    int a=5,b=8;
    p1=&a; p2=&b;
    if(a<b) { p=p1; p1=p2; p2=p; }
    printf("%d,%d ",*p1,*p2);
    printf("%d,%d",a,b);
}
```

程序的输出结果是 (A)。

- A. 8, 5 5, 8
- B. 5, 8 8, 5
- C. 5, 8 5, 8
- D. 8, 5 8, 5

14、若有定义和语句如下:

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

输出结果是 (D)。

- A. 10, 20
- B. 10, 10
- C. 20, 10
- D. 20, 20

15、指针 s 所指字符串的长度为 (D)。

```
char *s="f\\Name\\Address\\n";
```

- A. 说法不合法
- B. 19
- C. 18
- D. 15

16、分析下面程序:

```
#include<stdio.h>
void main()
{
    int i;
```

```

int *int_ptr;
int_ptr=&i;
*int_ptr=5;
printf("i=%d",i);
}

```

程序的执行结果是 (D)。

- A. i=0 B. i 为不定值
C. 程序有错误 D. i=5

17、下列说法中不正确的是 (D)。

- A. 指针是一个变量
B. 指针中存放的是地址值
C. 指针可以进行加、减等算术运算
D. 指针变量不占用存储空间

18、若指针 p 已正确定义,其指向如下图所示,则执行语句 *p++; 后, *p 的值是 (B)。

- A. 20 B. 30 C. 40 D. 50

a[0]	a[1]	a[2]	a[3]	a[4]
10	20	30	40	50

p ↑

19、若指针 p 已正确定义,其指向如下图所示,则表达式 *++p 的值是 (B)。

- A. 20 B. 30 C. 21 D. 31

二、填空题

1、以下程序段的输出结果是 110。

```

int *var, ab;
ab=100; var=&ab; ab=*var+10;
printf("%d\n", *var);

```

2、以下程序的输出结果是 7 1。

```

int ast(int x,int y,int *cp,int *dp)

```

```

{
    *cp=x+y;
    *dp=x-y;
}

```

```

void main()

```

```

{
    int a,b,c,d;
    a=4; b=3;
    ast(a,b,&c,&d);
    printf("%d %d\n",c,d);
}

```

3、写出下列程序的执行结果

```

void swap(int *p1,int *p2)

```

```

{
    int p;
    p=*p1;
    *p1=*p2;
    *p2=p;
}

```

```
}  
void main()  
{   in a,b;  
    int *p1,*p2;  
    scanf("%d%d",&a,&b);  
    p1=&a; p2=&b;  
    if(a<b) swap(p1,p2);  
    printf("\na=%d,b=%d\n",a,b);  
}
```

若输入 6 9<CR>, 则程序的运行结果为 a=9, b=6。

4、一个专门用来存放另一个变量地址的变量叫 指针变量。

第9部分 数组

一、选择题

- 1、若有定义语句: `int (*p)[M];`, 其中的标识符 `p` 是 (C)。
- A. `M` 个指向整型变量的指针
 - B. 指向 `M` 个整型变量的函数指针
 - C. 一个行指针, 它指向具有 `M` 个整型元素的一维数组
 - D. 具有 `M` 个指针元素的一维指针数组, 每个元素都只能指向整型量
- 2、以下程序段给数组所有元素输入数据:

```
void main()
{
    int a[10], i=0;
    while(i<10) scanf("%d",      );
    .....
}
```

应在下划线处填入的是 (A)。

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

3、以下程序:

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

其输出结果是 (C)。

- A. 17
- B. 18
- C. 19
- D. 20

4、以下程序:

```
void main()
{
    int a[]={2,4,6,8}, *p=a, i;
    for(i=0; i<4; i++) a[i]=*p++;
    printf("%d\n", a[2]);
}
```

其输出结果是 (A)。

- A. 6
- B. 8
- C. 4
- D. 2

5、以下程序:

```
void fun(int *s, int n1, int n2)
{
    int i, j, t;
    i=n1; j=n2;
    while(i<j)
    {
        t=*(s+i); *(s+i)=*(s+j); *(s+j)=t;
        i++; j--;
    }
}
```

```
void 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));
}
```

其输出结果是 (C)。

- A. 0987654321 B. 4321098765
C. 5678901234 D. 0987651234

6、运行下面程序：

```
#include "stdio.h"
#define N 6
void main()
{
    char c[N]; int i=0;
    for(i=0;i<N;i++) c[i]=getchar();
    for(i=0;i<N;i++) putchar(c[i]);
}
```

输入 3 行，每行都是在第一列上开始；<CR>代表回车符：

a<CR>

b<CR>

cdef<CR>

则程序的输出结果是 (C)。

- | | | | |
|-----------|------|------|------|
| A. abcdef | B. a | C. a | D. a |
| | b | b | b |
| | c | cd | cdef |
| | d | | |

7、若有定义和语句：

```
int a[4][3]={1,2,3,4,5,6,7,8,9,10,11,12},(*prt)[3]=a,*p[4],i;
for(i=0;i<4;i++) p[i]=a[i];
```

则不能够正确表示 a 数组元素的表达式是 (A)。

- A. a[4][3] B. p[0][0] C. prt[2][2] D. (*(p+1))[1]

8、以下程序：

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

其输出结果是 (A)。

- A. 23 B. 26 C. 33 D. 36

9、若有定义：

```
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 数组元素的引用不正确的是 (D)。

- A. `a[p-a]` B. `*(&a[i])` C. `p[i]` D. `*(*(a+i))`

10、以下程序:

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

其输出结果是 (A)。

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

11、下面程序的运行结果是 (C)

```
void main()
{
    int a[6][6],i,j;
    for(i=1;i<6;i++)
        for(j=1;j<6;j++)
            a[i][j]=(i/j)*(j/i);
    for(i=1;i<6;i++)
    {
        for(j=1;j<6;j++)
            printf("%2d",a[i][j]);
        printf("\n");
    }
}
```

A. 1 1 1 1 1

B. 0 0 0 0 1

1 1 1 1 1

0 0 0 1 0

1 1 1 1 1

0 0 1 0 0

1 1 1 1 1

0 1 0 0 0

1 1 1 1 1

1 0 0 0 0

C. 1 0 0 0 0

D. 1 0 0 0 1

0 1 0 0 0

0 1 0 1 0

0 0 1 0 0

0 0 1 0 0

0 0 0 1 0

0 1 0 1 0

0 0 0 0 1

1 0 0 0 1

12、下述对 C 语言字符数组的描述中正确的是 (B)。

- A. 任何一个一维数组的元素在引用时, 其下标可以是整型、实型或字符型常量
 B. 一维数组的元素实质上就是一个变量, 代表内存中的一个存储单元
 C. 如果要引用数组中的每一个元素, 则直接使用该数组名称即可
 D. 以上都不对

13、以下语句定义正确的是 (D)。

- A. `int a[1][4]={1,2,3,4,5};`
 B. `float x[3][]={{1},{2},{3}};`
 C. `long b[2][3]={{1},{1,2},{1,2,3}};`
 D. `double y[][3]={0};`

14、下述对 C 语言字符数组的描述正确的是 (A)。

- A. 任何一维数组的名称都是该数组存储单元的起始地址, 且其每个元素按照顺序连续占用存储空间

- B. 一维数组的元素在引用时其下标大小没有限制
 C. 任何一个一维数组的元素, 可以根据内存的情况按照其先后顺序以连续或非连续的方式占用存储空间
 D. 一维数组的第一个元素是其下标为 1 的元素

15. 定义如下变量和数组:

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

则下面语句的输出结果是 (C)。

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

```
    printf("%d ",x[i][2-i]);
```

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

16. 执行以下程序段后, y 的值为 (B)。

```
static int a[]={1,3,5,7,9};
```

```
int y,x,*ptr;
```

```
y=1;
```

```
ptr=&a[1];
```

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

```
    y*=(ptr+x);
```

- A. 105 B. 15 C. 945 D. 无确定值

17. 设有变量说明 `int (*ptr)[M]`; 其中的标识符 `ptr` 是 (C)。

A. M 个指向整型变量的指针

B. 指向 M 个整型变量的函数指针

C. 一个指向具有 M 个整型元素的一维数组的指针

D. 具有 M 个指针元素的一维指针数组, 每个元素都只能指向整型变量

18. 在 C 语言中, 引用数组元素时, 其数组下标的数据类型允许是 (C)。

A. 整型常量

B. 整型表达式

C. 整型常量或整型表达式

D. 任何类型的表达式

19. 下列说法正确的是 (D)。

A. 在 C 语言中, 可以使用动态内存分配技术, 定义元素个数可变的数组

B. 在 C 语言中, 数组元素的个数可以不确定, 允许随机变动

C. 在 C 语言中, 数组元素的数据类型可以不一致

D. 在 C 语言中, 定义一个数组后, 就确定了它所容纳的具有相同数据类型元素的个数

20. 若有以下定义:

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

则值为 3 的表达式是 (B)。

A. `p+=2,*(p++)`

B. `p+=2,*++p`

C. `p+=3,*p++`

D. `p+=2,++*p`

21. 以下程序:

```
void main()
```

```
{    int n[2]={0},i,j,k=2;
```

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

```
        for(j=0;j<k;j++)    n[j]=n[i]+1;
```

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

}其输出结果是 (A)。

A. 不确定的值 B. 3 C. 2 D. 1

22、以下程序:

```
f(int b[],int n)
{   int i,r=1;
    for(i=0;i<=n;i++)   r=r*b[i];
    return  r;
}
```

```
void main()
{   int x,a[]={2,3,4,5,6,7,8,9};
    x=f(a,3);
    printf("%d\n",x);
}
```

其输出结果是 (B)。

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

23、以下程序中,若第一个 printf 语句输出的是 194,则第二个 printf 语句的输出结果是 (D)。

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

A. 212 B. 204 C. 1a4 D. 1a6

24、若有定义和语句:

```
int c[4][5],(*cp)[5]; cp=c;
```

则对 c 数组元素的正确引用是 (D)。

A. cp+1 B. *(cp+3) C. *(cp+1)+3 D. *(*cp+2)

25、若已定义:

```
int a[4][3]={1,2,3,4,5,6,7,8,9,10,11,12},(*prt)[3]=a,*p=a[0];
```

则能够正确表示数组元素 a[1][2]的表达式是 (D)。

A. *((*prt+1)[2]) B. *((p+5))
C. (*prt+1)+2 D. *((a+1)+2)

26、以下程序:

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

其输出结果是 (C)。

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

27、以下程序:

```
void 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+=*(p+i+j);
printf("%d\n",k);
}

```

其输出结果是 (A)。

A. 60 B. 68 C. 99 D. 108

28、若有以下说明，且 $0 \leq i < 10$ ，则 (C) 是对数组元素的错误引用。

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

A. *(a+i) B. a[p-a+i] C. p+i D. *(&a[i])

29、若有定义和语句，且 $0 \leq i < 10$ ，则对数组元素的错误引用是 (C)。

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

p=a;

A. *(a+i) B. a[p-a] C. p+i D. *(&a[i])

30、下述程序的作用是计算数组中的最大元素值及其下标。则划线处应为 (B)。

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

```
void finmax(int *s, int t, int *k)
```

```
{
    int i;
    for(i=0, *k=i; i<t; i++)
        if(s[i]>s[*k])
            _____;
}
```

```
void main()
```

```
{
    int a[10], i, k;
    for(i=0; i<10; i++)
        scanf("%d", &a[i]);
    findmax(a, 10, &k);
    printf("%d, %d", k, a[k]);
}
```

A. k=i B. *k=i C. k=i-s D. *k=i-s

31、下述程序的运行结果是 (A)。

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

```
void f(char *str)
```

```
{
    int a, b;
    for(a=b=0; str[a]!='\0'; a++)
        if(str[a]!='c')
            str[b++] = str[a];
    str[b] = '\0';
}
```

```
void main()
```

```
{
    char str[] = "abcdef";
    f(str);
    printf("str[]=%s", str);
}
```

A. str[]=abdef

B. str[]=abcdef

B. str[]=a

D. str[]=ab

32、以下 main 函数调用 invert 函数将串 s 的元素按逆序重新存放, (B) 是不能正确实现功能的程序。

```
void main()
```

```
{ char str[]="String";
```

```
  invert(str);
```

```
  .....
```

```
}
```

A.

```
void invert(char s[])
```

```
{ int i,j,t;
```

```
  for(i=0,j=strlen(s)-1;i<j;i++,--j)
```

```
  { t=*(s+i);
```

```
    *(s+i)=*(s+j);
```

```
    *(s+j)=t;
```

```
  }
```

```
}
```

B.

```
void invert(char *s)
```

```
{ int i,j,t;
```

```
  for(i=0,j=strlen(s)-1;i<j;i++,--j)
```

```
  { t=s[i];
```

```
    s[j]=s[i];
```

```
    s[i]=t;
```

```
  }
```

```
}
```

C.

```
void invert(char *s)
```

```
{ int t,*h,*p;
```

```
  h=s;
```

```
  p=s+strlen(s)-1;
```

```
  while(h<p)
```

```
  { t=*h;
```

```
    *h=*p;
```

```
    *p=t;
```

```
    p--;
```

```
    h++;
```

```
  }
```

```
}
```

D.

```
void invert(char s[])
```

```
{ int t,*p;
```

```
  p=s+strlen(s)-1;
```

```

while(s<p)
{
    t=*s;
    *s++=*p;
    *p--=t;
}
}

```

二、填空题

1、下述程序的功能是计算数组中的最大元素：

```

#include<stdio.h>
#define N 10
void main()
{
    int a[N];
    int max, i;
    #if N<=10
        for(i=0; i<N; i++)
            a[i]=10+i;
    #else
        for(i=0; i<N; i++)
            scanf("%d", a+i);
    #endif
    max=a[0];
    for(i=1; i<N; i++)
        if(max<a[i])
            max=a[i];
    printf("Max=%d\n", max);
}

```

2、下述函数从一个数组 v 中删除值为 key 的元素。数组的元素个数由指针 n 指明。请填空。

```

void denode(double v[], int *n)
{
    int i, j, k=-1;
    for(i=*n-1; i>=0; i--)
        if(v[i]==key)
        {
            k++;
            for(j=i; j<*n-(1+k); j++)
                v[j]=v[j+1];
        }
    if(k>=0)
        *n=*n-(1+k);
}

```

3、下述函数统计一个字符串中的单词个数。单词是指处在空格之间的字符序列。请填空。

```

int word(char *s)
{
    int num=0, flag=0;
    while(*s)
    {
        if(*s++==' ')

```

```

        flag=0;
    else
        if( flag==0 )
        {
            flag=1;
            num++;
        }
    }
    return num;
}

```

4、子串删除。函数 delete(s,i,n)从字符串 s 中删除从第 i 个字符开始的 n 个字符，请填空。

```

void delete(char s[],int i,int n)
{
    int j,k,length=0;
    while(s[length]) length++ ;
    --i;
    j=i;
    if( i<length )
    {
        k=i+n;
        if(i+n<=length)
            while(k<length)
                s[j++]=s[k++];
        s[j]='\0';
    }
}

```

5、index(t,s)函数用于确定字符串 t 是否为 s 的子串，若不是，函数返回-1，否则返回子串 t 在 s 中第一次出现时第一个字符的下标。请填空。

```

int index(char *t,char *s)
{
    int i=0,j=0;
    while(t[i]!='\0' && s[j]!='\0')
        if(t[i++]!=s[j++])
        {
            j=i-1; //当 while 中发生一次不成功的匹配时，母串中不能滑过一个
            i=0; //与子串具有相同长度的串，只能滑过一个字符位置。
        }
    if(t[i]=='\0')
        return j-i ;
    return -1;
}

```

6、若有以下定义和语句，则通过指针 p 应用值为 98 的数组元素的表达式是 *(p+5) 或 p[5]。

```

int w[10]={23,54,10,33,47,98,72,80,6},*p;
p=w;

```

7、在 C 语言中，对于二维数组 a[i][j]的地址可表示为 &a[i][j] 或 a[i]+j 或 *(a+i)+j。其中，对于 a 来说，它代表 数组第 i 行元素的首地址（或是由数组第 i 行元素组成的一维数组的数组名）。它是一个 地址值（或第 i 行元素组成的一维数组的首地址）。

8、在 C 语言中，引用数组只能通过 逐个引用 数组元素来实现，而不能通过整体引用

整个数组 来实现。

9、在 C 语言中, 将字符串作为 字符数组 处理。

10、若有定义 `int a[10]`; 则 `a[i]` 的地址可表示为 `&a[i]` 或 `*(a+i)`, `a[i]` 的值可表示为 `*(a+i)`。

11、以下程序的输出结果是 12。

```
void main()
{
    int arr[10], i, k=0;
    for(i=0; i<10; i++) arr[i]=i;
    for(i=0; i<4; i++) k+=arr[i]+i;
    printf("%d\n", k);
}
```

12、以下程序的输出结果是 3。

```
#define N 5
fun(char *s, char a, int n)
{
    int j;
    *s=a; j=n;
    while(a<s[j]) j--;
    return j;
}
```

```
void main()
{
    char s[N+1];
    int k, p;
    for(k=1; k<=N; k++) s[k]='A'+k+1;
    printf("%d\n", fun(s, 'E', N));
}
```

13、若输入 3 个整数 3, 2, 1, 则以下程序的输出结果是 2721。

```
void sub(int n, int uu[])
{
    int t;
    t=uu[n--];
    t+=3*uu[n];
    n++;
    if(t>=10)
    {
        uu[n++]=t/10;
        uu[n]=t%10;
    }
    else
        uu[n]=t;
}
```

```
void main()
{
    int i, n, aa[10]={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");
}
```

}

14、若有以下定义:

double w[10];

则数组元素下标的上限是 9 , 下限是 0 。15、以下程序的输出结果是 6 。

void main()

```
{ int a[]={2,4,6},*prt=&a[0],x=8,y,z;
  for(y=0;y<3;y++) z=*(prt+y)<x)?*(prt+y):x;
  printf("%d\n",z);
}
```

16、以下程序的输出结果是 -850, 2, 0 。

void main()

```
{ int i,j,row,col,m;
  int arr[3][3]={{100,200,300},{28,72,-30},{-850,2,6}};
  m=arr[0][0];
  for(i=0;i<3;i++)
    for(j=0;j<3;j++)
      if(m>arr[i][j])
        { m=arr[i][j]; row=i; col=j; }
  printf("%d,%d,%d\n",m,row,col);
}
```

17、以下程序统计从终端输入的字符中大写字母的个数, num[0]中统计字母 A 的个数, 其它依次类推。用#号结束输入, 请填空。

#include "stdio.h"

#include "ctype.h"

void main()

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

18、以下 findmax 返回 s 所指数组中最大元素的下标, 数组中元素的个数由 t 传入, 请填空。

findmax(int s[],int t)

```
{ int k,p;
  for(p=0,k=p;p<t;p++)
    if(s[p]>s[k]) k=p;
  return k;
}
```

第 10 部分 字符串

一、选择题

1、能正确进行字符串赋值、赋初值的语句组是 (B)。

A. char s[5]={‘a’,‘e’,‘i’,‘o’,‘u’,‘\0’};

B. char *s; s="good!";

C. char s[5]="good!";

D. char s[5]; s="good";

2、语句 printf("%d\n",strlen("ATS\n012\IW")); 的输出结果是 (C)。

A. 11

B. 10

C. 9

D. 8

3、以下程序:

```
#include "stdio.h"
```

```
#include "string.h"
```

```
void fun(char *w,int m)
```

```
{ char s,*p1,*p2;
```

```
  p1=w; p2=w+m-1;
```

```
  while(p1<p2)
```

```
  { s=*p1++; *p1=*p2--; *p2=s; }
```

```
}
```

```
void main()
```

```
{ char a[]="ABCDEFGH"
```

```
  fun(a,strlen(a));
```

```
  puts(a);
```

```
}
```

其输出结果是 (C)。

A. GFEDCBA

B. AGADAGA

C. AGAAGAG

D. GAGGAGA

4、设有如下定义:

```
char *aa[2]={ "abcd", "ABCD"};
```

则以下说法中正确的是 (D)。

A. aa 数组元素中的值分别是字符串: "abcd" 和 "ABCD"

B. aa 是指针变量, 它指向含有两个数组元素的字符型一维数组

C. aa 数组的两个元素分别存放的是含有四个字符的一维数组的首地址

D. aa 数组的两个元素中各自存放了字符 'a' 和 'A' 的地址

5、以下程序:

```
void main()
```

```
{ char *alpha[6]={ "ABCD", "EFGH", "IJKL", "MNOP", "QRST", "UVWX"};
```

```
  char **p
```

```
  int i;
```

```
  p=alpha;
```

```
  for(i=0;i<4;i++) printf("%s",p[i]);
```

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

```
}
```

其输出结果是 (C)。

- A. ABCDEFGHIJKL B. ABCD
C. ABCDEFGHIJKLMNOP D. AEIM

6、以下程序段:

```
char str[]="ABCD",*p=str;
printf("%d\n",*(p+4));
```

其输出结果是 (B)。

- A. 68 B. 0 C. 字符 D 的地址 D. 不确定的值

7、当运行以下程序:

```
#include "stdio.h"
```

```
char fun(char *c)
```

```
{ if(*c<='Z'&&*c>='A') *c='A'-'a';
  return *c;
```

```
}
```

```
void main()
```

```
{ char s[81],*p=s;
  gets(s);
  while(*p)
  { *p=fun(p); putchar(*p); p++; }
  putchar('\n');
```

```
}
```

时, 输入 OPEN THE DOOR<CR>(<CR>代表回车), 则输出结果是 (B)。

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

8、以下程序:

```
void main()
```

```
{ char s[]="ABCD",*p;
  for(p=s;p<s+4;p++) printf("%s\n",p);
```

```
}
```

其输出结果是 (A)。

- | | | | |
|---------|------|------|---------|
| A. ABCD | B. A | C. D | D. ABCD |
| BCD | B | C | ABC |
| CD | C | B | AB |
| D | D | A | A |

9、以下程序:

```
void 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'&&*<=9;j+=2)
      s=10*s+p[i][j]-'0';
  printf("%d\n",s);
```

```
}
```


其输出结果是 (A)。

- A. 6385 B. 69825 C. 63825 D. 693825

10、库函数 strcpy 用以复制字符串。若有以下定义和语句：

```
char str1[]="string",str2[8],*str3,*str4="string";
```

则对库函数 strcpy 的不正确调用是 (C)。

- A. strcpy(str1,"HELLO1"); B. strcpy(str2,"HELLO2");
C. strcpy(str3,"HELLO3"); D. strcpy(str4,"HELLO4");

11、有下面的程序段：

```
char a[3],b[]="China";
```

```
a=b;
```

```
printf("%s",a);
```

则 (D)。

- A. 运行后将输出 China B. 运行后将输出 Ch
C. 运行后将输出 Chi D. 编译出错

12、执行以下程序段后的结果为 (B)。

```
static char a[]="language";
```

```
char *ptr;
```

```
ptr=a;
```

```
while(*ptr)
```

```
    printf("%c",*ptr-32);
```

```
ptr++;
```

- A. LANGUAGE B. 陷入死循环
C. 有语法错误 D. language

13、若有以下定义和语句：

```
char *s1="12345",*s2="1234";
```

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

其输出结果是 (A)。

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

14、以下程序段的输出为 (B)。

```
static char a[]="language",b[]="progratne";
```

```
char *ptr1,*ptr2;
```

```
int k;
```

```
ptr1=a; ptr2=b;
```

```
for(k=0;k<7;k++)
```

```
    if(*(ptr1+k)==*(ptr2+k))
```

```
        printf("%c",*(ptr1+k));
```

- A. gae B. ga
C. language D. 有语法错误

15、下列程序的输出结果是 (D)。

```
void main()
```

```
{ char *p1,*p2,str[50]="xyz";
```

```
    p1="abcd";
```

```
    p2="ABCD";
```

```
    strcpy(str+2,strcat(p1+2,p2+1));
```

```
printf("%s",str);
```

```
};
```

A. xyabcAB

B. abcABz

C. Ababcz

D. xycdBCD

16、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)。

A. ps+2

B. s[2]

C. ps[2]

D. ps+=2,*ps

17、以下程序段的输出结果是 (B)。

```
static char a[]="Program";
```

```
char *ptr;
```

```
ptr=a;
```

```
for(ptr=a;ptr<a+7;ptr+=2) putchar(*ptr);
```

A. Program

B. Porm

C. 有语法错误

D. Por

18、下面程序段的运行结果是 (D)。

```
char a[7]="abcdef";
```

```
char b[4]="ABC";
```

```
strcpy(a,b);
```

```
printf("%c",a[5]);
```

A. □(表示空格)

B. \0

C. e

D. f

19、以下程序的输出结果是 (D)。

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

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

```
void main()
```

```
{ char w[][10]={"ABCD","EFGH","IJKL","MNOP"},k;
```

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

```
printf("%s\n",&w[k][k]);
```

```
}
```

A. ABCD

B. ABC

C. EFG

D. FGH

FGH

EFG

JK

KL

KL

IJ

O

M

20、以下程序的输出结果是 (A)。

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

```
void sub1(char a,char b){ char c; c=a;a=b;b=c; }
```

```
void sub2(char *a,char b){ char c; c=*a; *a=b; b=c; }
```

```
void sub3(char *a,char *b){ char c; c=*a; *a=*b; *b=c; }
```

```
void main()
```

```
{ char a,b;
```

```
a='A';b='B';sub3(&a,&b); putchar(a); putchar(b);
```

```
a='A';b='B';sub2(&a,b); putchar(a); putchar(b);
```

```

a='A';b='B';sub1(a,b); putchar(a); putchar(b);
}

```

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

二、填空题

1、以下程序的输出结果是 GFEDCB。

```

#include "stdio.h"
void main()
{
    char b[]="ABCDEFGH",*chp=&b[7];
    while(--chp>&b[0]) putchar(*chp);
    putchar('\n');
}

```

2、以下程序的输出结果是 SO。

```

void main()
{
    char *p[]={"BOOL","OPK","H","SP"};
    int i;
    for(i=3;i>=0;i--) printf("%c",*p[i]);
    printf("\n");
}

```

3、以下程序的输出结果是 It is。

```

#include "stdio.h"
#include "string.h"
#include "ctype.h"
void fun(char str[])
{
    int i,j;
    for(i=0,j=0;str[i];i++)
        if(isalpha(str[i])) str[j++]=str[i];
    str[j]='\0';
}
void main()
{
    char ss[80]="It is!";
    fun(ss);
    printf("%s\n",ss);
}

```

4、以下语句的输出结果是 3。

```
printf("%d\n",strlen("s\n\016\0end"));
```

5、以下程序的输出结果是 XYZA。

```

#include "stdio.h"
void fun(char *a1,char *a2,int n)
{
    int k;
    for(k=0;k<n;k++)
        a2[k]=(a1[k]-'A'-3+26)%26+'A';
    a2[n]='\0';
}

```

```

}
void main()
{
    char s1[5]="ABCD",s2[5];
    fun(s1,s2,4);
    puts(s2);
}

```

6、下面程序的功能是将已按升序排好序的两个字符串 a 和 b 中的字符按升序归并到字符串 c 中，补足所缺语句：

```

#include<stdio.h>
void main()
{
    char a[]="acegikm";
    char b[]="bdfhjlnpq";
    char c[80],*p;
    int i=0,j=0,k=0;
    while(a[i]!='\0'&&b[j]!='\0')
    {
        if(a[i]<b[j])
        {
            c[k]=a[i];
            i++;
        }
        else
        {
            c[k]=b[j];
            j++;
        }
        k++;
    }
    c[k]='\0';
    if( a[i]!='\0' ) p=b+j;
    else p=a+i;
    strcat(c,p);
    puts(c);
}

```

7、以下程序的输出结果是 SO。

```

void main()
{
    char *p[]={"BOOL","OPK","H","SP"};
    int i;
    for(i=3;i>=0;i--,i--) printf("%c",*p[i]);
    printf("\n");
}

```

8、下面程序的功能是从键盘输入一行字符，统计其中有多少个单词，单词之间用空格分隔，补足所缺语句：

```

#include<stdio.h>
void main()
{
    char s[80],c1,c2=' ';
    int i=0,num=0;
}

```

```

    gets(s);
    while(s[i]!='\0')
    {
        c1=s[i];
        if(i==0) c2=' ';
        else c2=s[i-1];
        if( c1!=' ' && c2!=' ' ) num++;
        i++;
    }
    printf("There are %d words.\n",num);
}

```

- 9、当运行以下程序时从键盘输入字符串 qwerty 和 abcd，则程序的输出结果是

10. str1 is: qwertyabcd 。

```

#include<stdio.h>
#include<string.h>
strle(char a[],char b[])
{
    int num=0,n=0;
    while(*(a+num)!='\0') num++;
    while(b[n]) { *(a+num)=b[n]; num++; n++; }
    return (num);
}
void main()
{
    char str1[81],str2[81],*p1=str1,*p2=str2;
    printf("Input first string \n");
    gets(p1);
    printf("Input second string \n");
    gets(p2);
    printf("%d,str1 is: %s\n",strle(p1,p2),p1);
}

```

- 10、下面程序的功能是将字符串 s 中的所有的字符 c 删除，补足所缺语句：

```

#include<stdio.h>
void main()
{
    char s[80];
    int i,j;
    gets(s);
    for(i=j=0;s[i]!='\0';i++)
    if(s[i]!='c') s[j++]=s[i];
    s[j]='\0';
    puts(s);
}

```

- 11、当运行以下程序时从键盘输入字符串 qwerty 和 abcd，则程序的输出结果是 10 。

```

#include<stdio.h>
#include<string.h>
strle(char a[],char b[])
{
    int num=0,n=0;

```

```

while(*(a+num)!='\0') num++;
while(b[n]) { num++; n++; }
return (num);
}
void main()
{
    char str1[81],str2[81],*p1=str1,*p2=str2;
    gets(p1); gets(p2);
    printf("%d\n",strle(p1,p2));
}

```

12、以下 fun 函数的功能是将一个字符串的内容颠倒过来，请填空：

```

#include<stdio.h>
#include<string.h>
void fun(char str[])
{
    int i,j,k;
    for(i=0,j=____strlen(str)-1____;i<j;i++, ____j--____ )
    {
        k=str[i]; str[i]=str[j]; str[j]=k; }
}

```

13、以下程序段的输出结果是 goodgood!。

```

char s[20]="goodgood!"; *sp=s;
sp=sp+2;
sp="to";
puts(s);

```

第 11 部分 对函数的进一步讨论

一、选择题

1、以下叙述正确的是 (A)。

- A. C 程序中各函数之间既允许直接递归调用, 也允许间接递归调用
- B. C 程序中各函数之间不允许直接递归调用, 也不允许间接递归调用
- C. C 程序中各函数之间允许直接递归调用, 不允许间接递归调用
- D. C 程序中各函数之间不允许直接递归调用, 允许间接递归调用

2、以下程序:

```
long fib(int n)
{
    if(n>2) return (fib(n-1)+fib(n-2));
    else return (2);
}

void main()
{
    printf("%ld\n",fib(6));
}
```

其输出结果是 (C)。

- A. 8
- B. 30
- C. 16
- D. 2

3、以下叙述不正确的是 (D)。

- A. C 程序的 main 函数可以没有参数
- B. C 程序的 main 函数可以有参数
- C. C 程序的 main 函数若有参数, 运行此程序时第一个参数的值最少是 1
- D. main 函数的第一个参数必须是整型, 其名字必须是 argc; 第二个参数名字必须是 argv, 可以定义成: char *argv[]

4、若有以下说明和定义:

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

```
fun(int *c) { ..... }
```

则对 fun 函数的正确调用语句是 (B)。

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

5、在 C 语言中, main 函数的参数的正确表达式是 (ABC)。

- A. main(int argc, char *argv[]);
- B. main(int c, char *v[]);
- C. main(int x, char **y);
- D. main(int argc, char argv[]);

6、以下程序的输出结果是 (B)。

- A. 11, 42, 31
- B. 11, 41, 20
- 12, 22, 41
- 12, 42, 20
- C. 11, 21, 40
- D. 11, 41, 21
- 11, 21, 41
- 12, 42, 22

```
#include<stdio.h>
void prt(int *x,int *y,int *z)
{ printf("%d,%d,%d\n",++*x,++*y,*(z++)); }
int a=10,c=20,b=40;
void main()
{ prt(&a,&b,&c); prt(&a,&b,&c); }
```

7、在下面的程序中调用 gettwo 函数分配两个双精度数的存储单元，分别由 p1 个 p2 指向。能正确执行以上功能的程序是 (B)。

A.

```
#include<stdlib.h>
void gettwo(double *q1,double *q2)
{ q1=(double *)malloc(sizeof(double));
  q2=(double *)malloc(sizeof(double));
}
void main(){ double *p1=NULL,*p2=NULL; gettwo(p1,p2); ... }
```

B.

```
#include<stdlib.h>
void gettwo(double **q1,double **q2)
{ *q1=(double *)malloc(sizeof(double));
  *q2=(double *)malloc(sizeof(double));
}
void main(){ double *p1=NULL,*p2=NULL; gettwo(&p1,&p2); ... }
```

C.

```
#include<stdlib.h>
void gettwo(double *q1,double *q2)
{ *q1=(double *)malloc(sizeof(double));
  *q2=(double *)malloc(sizeof(double));
}
void main(){ double *p1=NULL,*p2=NULL; gettwo(&p1,&p2); ..... }
```

D.

```
#include<stdlib.h>
void gettwo(double **q1,double **q2)
{ q1=(double *)malloc(sizeof(double));
  q2=(double *)malloc(sizeof(double));
}
void main(){ double *p1=NULL,*p2=NULL; gettwo(p1,p2); ..... }
```

8、C 语言规定，程序中各函数之间 (A)。

- 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

9、下述程序的作用是计算数组中的最大元素值及其下标。在划线处应填入 (B)。

```
#include<stdio.h>
void findmax(int *s,int t,int *k)
```



```

{   int p;
    for(p=0,*k=p;p<t;p++)
        if(s[p]>s[*k])
            _____;
}

```

```

void main()
{   int a[10],k,i;
    for(i=0;i<10;i++)
        scanf("%d",&a[i]);
    findmax(a,10,&k);
    printf("%d,%d\n",k,a[k]);
}

```

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

10、下述程序的输出结果是 (C)。

```

#include<stdio.h>
long fun(int n)
{   long s;
    if(n==1||n==2) s=2;
    else s=n+fun(n-1);
    return s;
}

```

```

void main()
{   printf("n%d\n",fun(4)); }

```

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

11、下述程序的输出结果是 (A)。

```

#include<stdio.h>
int fun(int x)
{   int p;
    if(x==0||x==1) return 3;
    else
    {   p=x-fun(x-2);
        return p;
    }
}

```

```

void main()
{   printf("%d\n",fun(9)); }

```

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

12、以下选项中能正确进行字符串比较的函数是 (CD)。

A.

```

strcmp1(char *s,char *t)
{   for(*s++=*t++;) if(*s=='\0') return (0);
    return (*s-*t);
}

```

B.

```
strcmp2(char *s, char *t)
{
    for(;; *s++ == *t++;) if(!*s) return (0);
    return (*s - *t);
}
```

C.

```
strcmp3(char *s, char *t)
{
    for(;; *s == *t;) { if(!*s) return (0); s++; t++; }
    return (*s - *t);
}
```

D.

```
strcmp4(char *s, char *t)
{
    for(;; *s == *t; s++, t++) if(!*s) return (0);
    return (*s - *t);
}
```

13、以下程序调用 getsum 函数把指针 p1 和 p2 所指存储单元中的数相加，和数放入指针 s 所指存储单元中。能正确执行以上功能的程序是 (C D)。

A.

```
#include<stdio.h>
void getsum(int *p1, int *p2, int *s) { int t; t = *p1 + *p2; s = &t; }
void main()
{
    int a, b, *p1 = &a, *p2 = &b, *s;
    scanf("%d%d", p1, p2);
    getsum(p1, p2, s);
    printf("%d\n", *s);
}
```

B.

```
#include<stdio.h>
void getsum(int *p1, int *p2, int *s) { s = *p1 + *p2; }
void main()
{
    int a, b, *p1 = &a, *p2 = &b, *s;
    scanf("%d%d", p1, p2);
    getsum(p1, p2, s);
    printf("%d\n", *s);
}
```

C.

```
#include<stdio.h>
#include<stdlib.h>
void getsum(int *p1, int *p2, int **s)
{
    *s = (int *)malloc(sizeof(int)); **s = *p1 + *p2; }
void main()
{
    int a, b, *p1 = &a, *p2 = &b, *s;
    scanf("%d%d", p1, p2);
    getsum(p1, p2, &s);
    printf("%d\n", *s);
}
```

```

}
D.
#include<stdio.h>
#include<stdlib.h>
void getsum(int *p1,int *p2,int *s) { *s=*p1+*p2; }
void main()
{   int a,b,*p1=&a,*p2=&b,*s;
    scanf("%d%d",&p1,&p2);
    s=(int *)malloc(sizeof(int));
    getsum(p1,p2,s);
    printf("%d\n",*s);
}

```

14、以下程序删除字符串 s 中的所有空格（包括 Tab 符、回车符、换行符）。请在对应的一组选择项中选择正确的内容填入：(1) 为（ AD ），(2) 为（ BC ），(3) 为（ A C ）。

```

#include<stdio.h>
#include<string.h>
#include<ctype.h>
void main()
{   void delspace(char *str);
    char s[81]="abcd  efg ";
    delspace(s);
    puts(s);
}
void delspace(char *str)
{   int i,t;
    char ts[81];
    for(i=0,t=0; (1) ; (2) )
        if(!isspace( (3) )) ts[t++]=str[i];
    ts[t]='\0';
    strcpy(str,ts);
}

```

- (1) A. str[i] B. !str[i] C. str[i]=='\0' D. str[i]!=NULL
 (2) A. t++ B. ++i C. i++ D. ++t
 (3) A. str[i] B. *ts[i] C. *(str+i) D. *(ts+i)

15、下面程序的输出结果是（ A ）。

```

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

```

```

    if(a>b) c=1;
    else if(a==b) c=0;
    else c=-1;
    return (c);
}

```

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

二、填空题

1、以下程序把字符串 s 中的内容按逆序输出，但不改变串中的内容。在下划线处填入正确的内容：

```

#include<stdio.h>
#include<string.h>
void main()
{   char s[10]="Hello!";
    inverp(s);
    printf("%s\n",s);
}
inverp(char *a)
{   if( !__a__ ) return 0;
    inverp(a+1);
    printf("%c",__a__);
}

```

2、以下程序实现将 3 个数中的大数输出在前，小数输出在后。在下划线处填入正确的内容：

```

#include<stdio.h>
void swap(int *p1,int *p2)
{   int *t;
    __t=(int *)malloc(sizeof(int)) 或  t=(int *)calloc(1,sizeof(int)) __;
    *t=*p1; *p1=*p2; *p2=*t;
}
void change(int *q1,int *q2,int *q3)
{   if(*q1>*q2) swap(q1,q2);
    if(*q1>*q3) swap(q1,q3);
    if(*q2>*q3) swap(q2,q3);
}
void main()
{   int a=5,b=4,c=2;
    change(__&c, &b, &a__);
    printf("%d,%d,%d\n",a,b,c);
}

```

3、以下程序调用 func 函数，实现函数 fs(x,y,z)和 fc(x,y,z)的计算。在下划线处填入正确的内容：

$$fs(x,y,z) = \frac{\sin(x)}{\sin(x-y) * \sin(x-z)} + \frac{\sin(y)}{\sin(y-z) * \sin(y-x)} + \frac{\sin(z)}{\sin(z-x) * \sin(z-y)}$$

$$fc(x, y, z) = \frac{\cos(x)}{\cos(x-y) * \cos(x-z)} + \frac{\cos(y)}{\cos(y-z) * \cos(y-x)} + \frac{\cos(z)}{\cos(z-x) * \cos(z-y)}$$

```
#include<stdio.h>
#include<math.h>
double mysin(double x){ return (sin(x)); }
double mycos(double x){ return (cos(x)); }
double func(double (*fnp)(),double a,double b,double c)
{ double x;
  x=(*fnp)(a)/(((*fnp)(b))*(*fnp)(c));
  return (x);
}
void main()
{ double x,y,z,(*fp)();
  double ms,mc;
  scanf("%lf%lf%lf",&x,&y,&z);
  fp=mysin;
  ms=func(fp, x, x-y, x-z)+func(fp, y, y-z, y-x)+func(fp, z, z-x, z-y);
  fp=mycos;
  mc=func(fp, x, x-y, x-z)+func(fp, y, y-z, y-x)+func(fp, z, z-x, z-y);
  printf("ms=%f,mc=%f\n",ms,mc);
}
```

4、下列程序实现用递归的方法求 $m!$ 请填空。

```
#include<stdio.h>
long fac(int n)
{ long t;
  if(n==1||n==0) return 1;
  else
  { t= n*fac(n-1);
    return t;
  }
}
```

5、以下程序的输出结果是 7。

```
int fun(int x)
{ int p;
  if(x==0||x==1) return (3);
  p=x-fun(x-2);
  return p;
}
```

```
void main()
{ printf("%d\n",fun(9)); }
```

6、以下程序的输出结果是 8。

```
#include<stdio.h>
void fun(int n,int *s)
```

```

{   int f1,f2;
    if(n==1||n==2) *s=1;
    else
    {   fun(n-1,&f1);
        fun(n-2,&f2);
        *s=f1+f2;
    }
}

```

```

void main()
{   int x;
    fun(6,&x);
    printf("%d\n",x);
}

```

7、以下程序调用 invert 函数按逆序重新放置 a 数组中元素的值，a 数组中的值在 main 函数中读入。请填空。

```

#include<stdio.h>
#define N 10
void invert(int *s,int i,int j)
{   int t;
    fi(i<j)
    {   t=*(s+i); *(s+i)=____*(s+j)____; *(s+j)=t;
        invert(s,____i+1____,j-1);
    }
}
void main()
{   int a[N],i;
    for(i=0;i<N;i++) scanf("%d",&a+____i____);
    invert(a,1,N-1);
    for(i=0;i<N;i++) printf("%d ",a[i]);
    printf("\n");
}

```

8、下述函数通过递归方法将字符串倒置，使用时需要指定字符数组的首地址、起始下标和终止下标。请填空。

```

#include<stdio.h>
void fun(char *s,int low,int high)
{   if( ____low>=high____ ) return;
    else
    {   char t;
        fun( ____s,low+1,high-1____ );
        t=s[low];
        s[low]=s[high];
        s[high]=t;
    }
}

```

9、以下程序调用函数 swap_p 将指针 s 和 t 所指单元 (a 和 b) 中的内容交换。请填空。

```
#include<stdio.h>

void main()
{   int a=10,b=20,*s,*t;
    void swap_p(int **,int **);
    s=&a; t=&b;
    swap_p(&s,&t);
    printf("%d %d\n",a,b);
}

void swap_p(  int  **  ss,  int  **  tt)
{   int temp;
    temp=  **ss  ;
    **ss=  **tt  ;
    **tt=temp;
}
```

10、下述程序的输出结果是 9 。

```
#include<stdio.h>
long fun(int n)
{   long s;
    if(x==0||x==1) s=2;
    else s=n+fun(n-1);
    return s;
}

void main()
{   printf("%ld\n",fun(4)); }
```

11、以下程序的输出结果是 8 。

```
#include<stdio.h>
int fun(int n,int *s)
{   int f1,f2;
    if(n==1||n==2) *s=1;
    else
    {   fun(n-1,&f1);
        fun(n-2,&f2);
        *s=f1+f2;
    }
}
```

```
void main()
{   int x;
    fun(6,&x);
    printf("%d\n",x);
}
```

12、以下程序的输出结果是 17 。

```
#include<stdio.h>
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); }
void main()
{ int x, (*p)(int,int);
  p=funa;
  x=sub(p,9,3);
  x+=sub(funb,8,3);
  printf("%d\n",x);
}

```

13、以下程序中的 trap 函数是一个用梯形法求定积分的通用函数，请填空。
梯形法求定积分 s 的公式为：

$$s = ((f(a) + f(b))/2 + \sum_{i=1}^{n-1} f(a+i*h)) * h, \quad h = \left| \frac{a-b}{n} \right|$$

其中， n 为积分小区间个数，以下程序调用 trap 函数求定积分，被积函数是：

$f(x)=x*x+3*x+2$ ，且 $n=1000$ ， $a=0$ ， $b=4$

```

#include<stdio.h>
#include<math.h>
double trap(double _____ (*fun)(double) _____, double a, double b)
{ double t,h;
  int i,n=1000;
  t=0.5*((*fun)(a)+(*fun)(b));
  h=fabs(a-b)/(double)(n);
  for(i=1;i<=n-1;i++) t=t+ _____ (*fun)(a+i*h) _____ ;
  t=t*h;
  return (t);
}
double mypoly(double x)
{ return (x*x+3.0*x+2.0); }
void main()
{ double y, (*pf)(double);
  pf= _____ mypoly _____ ;
  y=trap(pf,0.0,4.0);
  printf("%f\n",y);
}

```


第 12 部分 用户标识符的作用域和存储类别

一、选择题

1、C 语言中，形参的缺省的存储类型说明是 (A)。

- A. auto (自动)
- B. static (静态)
- C. register (寄存器)
- D. extern (外部)

2、以下叙述中不正确的是 (CD)。

- A. 一个变量的作用域的开始位置完全取决于变量定义语句的位置
- B. 全局变量可以在函数以外的任何部位进行定义
- C. 局部变量的“生存期”只限于本次函数调用，因此不可能将局部变量的运算结果保存至函数下一次调用时引用
- D. 一个变量说明为 static 存储类型是为了限制其它编译单位的引用

3、下述程序的输出结果是 (A)。

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

- A. 2,1,1
- B. 1,2,1
- C. 2,2,2
- D. 2,2,1

4、C 语言中形参的缺省存储类别是 (A)。(但不能将 auto 直接加在形参之前，唯一允许加在形参之前的存储类型关键字是 register)

- A. 自动 (auto)
- B. 静态 (static)
- C. 寄存器 (register)
- D. 外部 (extern)

5、在 C 语言中，存储类型为 (B) 的变量只在使用它们时才占用存储空间。

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

6、若一个外部变量的定义形式为 static int x; 那么，其中 static 的作用应该是 (C)。

- A. 将变量存放在静态存储区
- B. 使变量 x 可以由系统自动初始化
- C. 使 x 只能在本文件内使用
- D. 使 x 的值可以永久保留

7、以下程序：

```
#include<stdio.h>
void main()
{   int i=1,j=3;
    printf("%d,", i++);
    {   int i=0;
        i+=j*2;
        printf("%d,%d," i,j);
    }
    printf("%d,%d\n",i,j);
}
```

其输出结果是 (B)。

A. 1, 6, 3, 1, 3

B. 1, 6, 3, 2, 3

C. 1, 6, 3, 6, 3

D. 1, 7, 3, 2, 3

8、以下程序:

```
#include<stdio.h>
void main()
{   int a[]={1,2,3,4},i,x=0;
    for(i=0;i<4;i++) {   sub(a,&x);   printf("%d ",x);   }
    printf("\n");
}
sub(int *x,int *y)
{   static int t=3;
    *y=s[t];
    t--;
}
```

其输出结果是 (B)。

A. 1 2 3 4

B. 4 3 2 1

C. 0 0 0 0

D. 4 4 4 4

9、以下程序:

```
void main()
{   int k=4,m=1,p;
    p=func(k,m);
    printf("%d,",p);
    p=func(k,m);
    printf("%d\n",p);
}
func(int a,int b)
{   static int m,i=2;
    i+=m+1;
    m=i+a+b;
    return (m);
}
```

其输出结果是 (A)。

A. 8, 17

B. 8, 16

C. 8, 20

D. 8, 8

10、以下程序:

```
void fun(int *s)
{
    static int j=0;
    do
        s[j]+=s[j+1];
    while(++j<2);
}

void 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]);
}
```

其输出结果是 (C)。

A. 34756

B. 23445

C. 35745

D. 12345

11、以下程序:

```
f(int a)
{
    int b=0;
    static int c=3;
    a=c++,b=3;
    return (a);
}

void main()
{
    int a=2,i,k;
    for(i=0;i<2;i++) k=f(a++);
    printf("%d\n",k);
}
```

其输出结果是 (D)。

A. 3

B. 6

C. 5

D. 4

12、以下程序:

```
int m=13;

int fun2(int x,int y)
{
    int m=3;
    return (x*y-m);
}

void main()
{
    int a=7,b=5;
    printf("%d\n",fun2(a,b)/m);
}
```

其输出结果是 (B)。

A. 1

B. 2

C. 7

D. 10

13、以下程序的输出结果是 (D)。

```
f(int a)
{
    int b=0;
    static int c=3;
```

```

    a=c++,b++;
    return (a);
}
void main()
{   int a=2,i,k;
    for(i=0;i<2;i++) k=f(a++);
    printf("%d\n",k);
}

```

A. 3 B. 6 C. 5 D. 4

二、填空题

1、以下程序的输出结果是 2, 5, 1, 2, 3, -2。

```

void 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\n",a,b,c);
}

```

2、以下程序的输出结果是 2468。

```

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

```

第 13 部分 编译预处理

一、选择题

1、以下叙述中正确的是 (D)。

- A. 用#include 包含的头文件的后缀不可以是“.a”
- B. 若一些源程序中包含某个头文件, 当该头文件有错时, 只需对该头文件进行修改, 包含此头文件的所有源程序不必重新进行编译
- C. 宏命令行可以看成是一行 C 语句
- D. C 编译中的预处理是在编译之前进行的

2、下面是对宏定义的描述, 不正确的是 (C)。

- A. 宏不存在类型问题, 宏名无类型, 它的参数也无类型
- B. 宏替换不占用运行时间
- C. 宏替换时先求出实参表达式的值, 然后代入形参运算求值
- D. 其实, 宏替换只不过是字符替代而已

3、下述程序段中, (D) 不含有错误。

A.

```
#define F(n) ((n)==1?1:(n*F(n)-1))
```

B.

```
#define swap(x,y) int t=x;x=y;y=t;
```

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

```
void main()
```

```
{ int a,b;
```

```
  a=10,b=20;
```

```
  swap(a,b);
```

```
  printf(“%d,%d”,a,b);
```

```
}
```

C.

```
#define M(x,y) (x/y)
```

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

```
void main()
```

```
{ int a=2,b=3,c=0,d=5;
```

```
  int x=M(a+b,c+d);
```

```
  printf(“%d”,x);
```

```
}
```

D.

```
#define PLUS +
```

```
void main()
```

```
{ int x,y;
```

```
  #define OK 1
```

```
  scanf(“%d”,&x);
```

```
  y=10;
```

```

    if(x==OK)
        y=y PLUL 1;
    printf("%d",y);
}

```

4、执行下述程序后，输出结果是（ B ）。

```

#include<stdio.h>
#define SQR(x) x*x
void main()
{
    int a=10,k=2,m=1;
    a=SQR(k+m)/SQR(k+m);
    printf("%d",a);
}

```

A. 10 B. 1 C. 9 D. 0

5、下述叙述中，（ C ）是错误的。

- A. #include 命令可以包含一个含有函数定义的 C 语言源程序文件
- B. 使用#include<文件名>的形式比使用#include"文件名"格式省编译时间
- C. #include"C:\USER\F1.H"是正确的包含命令，它表示文件 F1.H 存放在 D 盘的 USER 目录下
- D. #include<...>格式中的文件名之前是不能包含路径的

6、以下程序（程序左边的数字为附加的行号）（ B ）。

```

1 #include<str.h>
2 #include<stdio.h>
3 void main()
4 {
5     char *s="string";
6     puts(s);
7     strcpy(s,"hello");
8     printf("%3s\n",s);
9 }

```

- A. 没有错
- B. 第 1 行有错
- C. 第 6 行有错
- D. 第 7 行有错

7、以下程序：

```

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

```

其输出结果是（ A ）。

- A. 15
- B. 100
- C. 10
- D. 150

8、以下程序：

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

for 循环执行的次数是 (C)。

A. 5 B. 6 C. 8 D. 9

9、以下程序：

```
#include "stdio.h"
#define FUDGF(y) 2.84+y
#define PR(a) printf("%d", (int)(a))
#define PRINT1(a) PR(a); putchar('\n')
void main()
{ int x=2;
  PRINT1(FUDGF(5)*x);
}
```

其输出结果是 (B)。

A. 11 B. 12 C. 13 D. 15

10、将大写字母变为小写字母的宏定义为 (B)。

- A. #define TOLOWER(ch) (ch>='A')&&(ch<='Z')?ch+26:ch
- B. #define TOLOWER(ch) ((ch)>='A')&&((ch)<='Z')?(ch)+'a'-'A':(ch)
- C. #define TOLOWER(ch) ((ch)>='A')||((ch)<='Z')?(ch)+'a'-'A':(ch)
- D. #define TOLOWER(ch) ((ch)>='A')||((ch)<='Z')?(ch)+'26:(ch)

11、下面的说法不正确的是 (B)。

- A. 函数调用时，先求出实参表达式，然后带入形参。而使用带参的宏只是进行简单的字符替换
- B. 函数调用是在程序运行时处理的，分配临时的内存单元。而宏展开则是在编译时进行的，在展开时也要分配内存单元，进行值传递
- C. 对于函数中的实参和形参都要定义类型，二者的类型要求一致，而宏不存在类型的问题，宏没有类型
- D. 调用函数只可得到一个返回值，而用宏可以设法得到几个结果

12、下面 (D) 不是 C 语言所提供的预处理功能。

- A. 宏定义 B. 文件包含 C. 条件编译 D. 字符预处理

13、下面的说法不正确的是 (D)。

- A. 使用宏的次数较多时，宏展开后源程序长度增长。而函数调用不会使源程序变长

- B. 函数调用是在程序运行时处理的, 分配临时的内存单元。而宏展开则是在编译时进行的, 在展开时不分配内存单元, 不进行值传递
- C. 宏替换占用编译时间
- D. 函数调用占用编译时间

14. 程序:

```
#define NUM 30+4
```

```
void main()
```

```
{ printf("NUM*20=%d", NUM*20; )
```

的执行结果为 (A)。

- A. NUM*20=110 B. NUM*20=680
- C. NUM*20=604 D. 以上都不对

15. 以下说法正确的是 (B)。

- A. 宏定义是 C 语句, 所以要在行末加分号
- B. 可以使用 #undef 命令来终止宏定义的作用域
- C. 在进行宏定义时, 宏定义不能层层置换
- D. 对程序中用双引号括起来的字符串内的字符, 与宏名相同的要进行置换

16. 在宏定义 #define MAX 30 中, 用宏名代替一个 (B)。

- A. 常量 B. 字符串 C. 整数 D. 长整数

二、填空题

1. 以下程序的输出结果是 9。

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

```
#define SQR(x) x*x
```

```
void main()
```

```
{ int a,k=3;
```

```
  a=++SQR(k+1);
```

```
  printf("%d\n",a);
```

```
}
```

2. 以下程序的输出结果是 ar=9 ar=9 ar=11。

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

```
#define PR(ar) printf("ar=%d ",ar)
```

```
void main()
```

```
{ int j,a[]={1,3,5,7,9,11,13,15},*p=a+5;
```

```
  for(j=3;j--)
```

```
    switch(j)
```

```
    { case 1:
```

```
      case 2: PR(*p++); break;
```

```
      case 3: PR(*(--p));
```

```
    }
```

```
}
```


3、程序：

```
#define FUN This is a C program
#include<stdio.h>
void main()
{   printf("First: This is FUN!\n");
    printf("Second: %s!\n", "FUN");
}
```

的执行结果是 First: This is FUN!
Second: FUN!

4、#define 命令出现在程序中函数的外面，宏名的有效范围为 从定义命令开始到本源程序文件结束。

5、可以使用 #undef 命令来终止宏定义的作用域。

第 14 部分 结构体、共用体和用户定义类型

一、选择题

1、若有如下说明，则（ D ）的叙述是正确的（已知 int 类型占两个字节）。

```
struct st
{
    int a;
    int b[2];
}a;
```

- A. 结构体变量 a 与结构体成员 a 同名，定义是非法的
- B. 程序只在执行到该定义时才为结构体 st 分配存储单元
- C. 程序运行时为结构体 st 分配 6 字节存储单元
- D. 类型名 struct st 可以通过 extern 关键字提前引用（即引用在前，说明在后）

2、若有以下结构体定义：

```
struct example
{
    int x;
    int y;
}v1;
```

则（ D ）是正确的引用或定义。

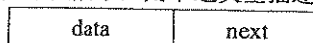
- A. example.x=10;
- B. example.v2.x=10;
- C. struct v2; v2.x=10;
- D. struct example v2={10};

3、对于如下的结构体定义，若对变量 person 的出生年份进行赋值，（ C ）是正确的赋值语句。

```
struct date
{
    int year, month, day;
};
struct worklist
{
    char name[20];
    char sex;
    struct date birthday;
}person;
```

- A. year=1976;
- B. birthday.year=1976;
- C. person.birthday.year=1976;
- D. person.year=1976;

4、已知形成链表的存储结构如下图所示，则下述类型描述中的空白处应填（ D ）。



```
struct link
{
    char data;
    _____;
}node;
```

- A. struct link next
- B. link *next
- C. struct next link
- D. struct link *next

5、下述程序的运行结果是 (A)。

```
#include<stdio.h>
struct st
{   int n;
    int *m;
}*p;
void main()
{   int d[5]={10,20,30,40,50};
    struct st arr[5]={100,d,200,d+1,300,d+2,400,d+3,500,d+4};
    p=arr;
    printf("%d\t",++p->n);
    printf("%d\t",(++p)->n);
    printf("%d\n",++(*p->m));
}
```

A. 101 200 21

B. 101 20 30

C. 200 101 21

D. 101 101 10

6、下述程序的执行结果是 (A)。

```
#include<stdio.h>
union un
{   short int   i;
    char c[2];
};
void main()
{   union un x;
    x.c[0]=10;
    x.c[1]=1;
    printf("%d\n",x.i);
}
```

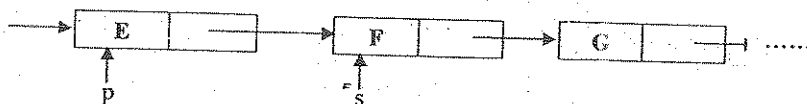
A. 266

B. 11

C. 265

D. 138

7、若已建立如下链表结构，指针 p 和 s 分别指向下图所示结点，则能够将指针 s 指向的结点从链表中删除并释放该结点的语句组是 (C)。



A. free(s);
p->next=s->next;

B. s=s->next;
p->next=s;
free(s);

C. p->next=s->next;
free(s);

D. s=s->next;
p->next=s;
p=p->next=s;
free(p);

12、设有以下定义：

```
struct stru
{   int a;
    long b;
    char c[6];
}sampl;
```

则以下赋值正确的是 (B)。

- A. sampl.c="abcd"; B. strcpy(sampl.c,"abcd");
C. sampl->c="abcd"; D. 以上都不对

13、以下程序的输出结果是 (C)。

```
struct st
{   int x;
    int *y;
}*p;
int dt[4]={10,20,30,40};
struct st aa[4]={50,&dt[0],60,&dt[0],70,&dt[0], 80,&dt[0]};
void main()
{   p=aa;
    printf("%d,",++p->x);
    printf("%d,",(++p->x));
    printf("%d\n",++(*p->y));
}
```

- A. 10, 20, 20 B. 50, 60, 21
C. 51, 60, 11 D. 60, 70, 31

14、以下程序的输出结果是 (D)。

```
#include<stdio.h>
void main()
{   union un
    {   long k;
        char c;
    };
    struct byte
    {   long b;
        union un c;
    }r;
    printf("%d\n",sizeof(r));
}
```

- A. 10 B. 13 C. 7 D. 8

15、下列说法不正确的是 (A)。

- A. 下列结构体定义时，占据了 5 个字节的空间

```
struct s { int num, age; char sex; };
```

- B. 结构体的成员名可以与程序中的变量名相同
 C. 对结构体中的成员可以单独使用，它的作用相当于普通变量
 D. 结构体的成员可以是一个结构体变量

16、设有如下定义：

```
struct sample
```

```
{   int a;
```

```
    int b;
```

```
};
```

```
struct st
```

```
{   int a;
```

```
    float b;
```

```
    struct sample *p;
```

```
}; st1, *pst;
```

若有 `pst=&st1`；则以下引用正确的是（ C ）。

A. `(*pst).p.a`

B. `(*pst)->p.a`

C. `pst->p->a`

D. `pst.p->a`

17、以下程序：

```
void 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);
```

```
}
```

其输出是（ D ）。

A. 0

B. 1

C. 3

D. 6

18、设有如下说明和语句：

```
typedef struct st
```

```
{   int n;
```

```
    struct st *next;
```

```
}; ST;
```

```
ST a[3], *p;
```

```
a[0].n=5; a[0].next=&a[1];
```

```
a[1].n=7; a[1].next=&a[2];
```

```
a[2].n=9; a[2].next='\0';
```

```
p=&a[0];
```

则值为 6 的表达式是（ D ）。

A. `p++->n`

B. `p->n++`

C. `(*p).n++`

D. `++p->n`

19、已知字符 0 的 ASCII 代码值的十进制数是 48，且数组的第 0 个元素在低位，以下程序：

```
void main()
```

```
{   union
```

```

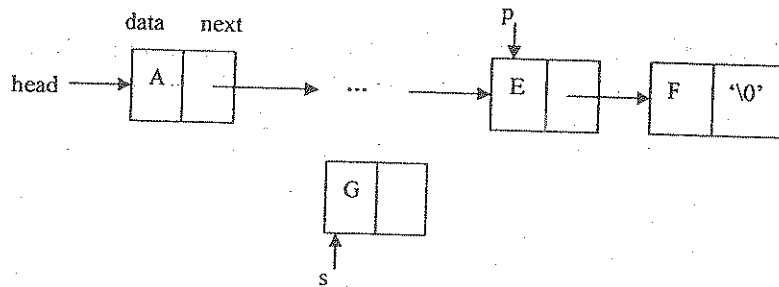
{
    int i[2];
    long k;
    char c[4];
}r,*s=&r;
s->i[0]=0x39;
s->i[1]=0x38;
printf("%x\n",s->c[0]);
}

```

其输出是 (A)。

- A. 39 B. 9 C. 38 D. 8

20、若已建立下面的链表结构，指针 p、s 分别指向图中所示结点，则不能将 s 所指结点插入到链表末尾的语句组是 (C)。



- A. $s \rightarrow \text{next} = \text{NULL}; p = p \rightarrow \text{next}; p \rightarrow \text{next} = s;$
 B. $p = p \rightarrow \text{next}; s \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} = s;$
 C. $p = p \rightarrow \text{next}; s \rightarrow \text{next} = p; p \rightarrow \text{next} = s;$
 D. $p = (*p). \text{next}; (*s). \text{next} = (*p). \text{next}; (*p). \text{next} = s;$

二、填空题

1、以下函数 creat 用来建立一个带头结点的单向链表，新产生的结点总是插在链表的末尾，结点数据域中的数值从键盘输入，以字符？作为输入结束标志。单向链表的头指针作为函数值返回。请填空。

```

#include<stdio.h>
struct list
{
    char data;
    struct list *next;
};
struct list *creat()
{
    struct list *h,*p,*q;
    char ch;
    h = (struct list *) malloc(sizeof( struct list ));
    p=q=h;
    ch=getchar();

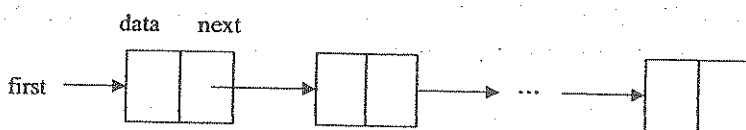
```

```

while(ch!='?')
{
    p= (struct list *) malloc(sizeof( struct list ));
    p->data==ch;
    q->next=p;
    q=p;
    ch=getchar();
}
p->next='\0';
return h ;
}

```

2、以下 min 函数的功能是：在带有头结点的单向链表中，查找结点数据域的最小值作为函数返回。请填空。



```

struct node
{
    int data;
    struct node *next;
};
int min(struct node *first)
{
    struct node *p;
    int m;
    p=first;
    m=p->data;
    for(p=p->next;p!='\0';p= p->next )
        if( m>p->data ) m=p->data;
    return m;
}

```

3、在 C 语言中，要定义一个结构体类型的变量，可采用三种方法，即 先定义结构体类型再定义结构体变量、在定义结构体类型的同时定义结构体变量 和 直接定义结构体类型变量。

4、在 C 语言中，使几个不同的变量共占同一段内存的结构称为 共用体。

5、以下程序的运行结果是 LBing。

```

typedef struct sampl
{
    char name[10];
    int number;
}SAM;

```

```

SAM test[3]={{"WangBing",10}, {"LiYun",20}, {"HuangHua",30}};

```



```
void main()
```

```
{ printf("%c%s\n", test[1].name[0], test[0].name+4); }
```

第 15 部分 文件

一、选择题

- 1、C 语言可以处理的文件类型是 (B)。
- A. 文本文件和数据文件 B. 文本文件和二进制文件
C. 数据文件和二进制文件 D. 数据代码文件

- 2、下述程序向文件输出的结果是 (C)。

```
#include<stdio.h>

void main()
{
    FILE *fp=fopen("TEST","wb");
    fprintf(fp,"%d%5.0f%c%d",58,76273.0,'-',2278);
    fclose(fp);
}
```

- A. 58.75273-22278 B. 5876273.000000-2278
C. 5876273-2278 D. 因文件为二进制文件而不可读

- 3、如果需要打开一个已经存在的非空文件“FILE”并对其进行修改，正确的打开语句是 (D)。

- A. fp=fopen("FILE","r");
B. fp=fopen("FILE","ab+");
C. fp=fopen("FILE","w+");
D. fp=fopen("FILE","r+");

- 4、已知一个文件中存放若干个学生档案记录，其数据结构如下：

```
struct st
{
    char num[10];
    int age;
    float s[5];
};
```

定义一个数组：struct st a[10];

假定文件已正确打开，(B) 不能正确地文件中读入 10 名学生数据到数组中。

- A. fread(a,sizeof(struct st),10,fp);
B. for(i=0;i<10;i++)
 fread(a[i],sizeof(struct st),1,fp);
C. for(i=0;i<10;i++)
 fread(a+i,sizeof(struct st),1,fp);
D. for(i=0;i<5;i+=2)
 fread(a+i,sizeof(struct st),2,fp);

- 5、标准库函数 fgetc(s,n,f) 的功能是 (B)。

- 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
6. 若 *fp* 是指向某文件的指针, 且已读到文件末尾, 库函数 *fgetc(fp)* 的返回值是 (C)。
A. EOF B. -1 C. 非零值 D. NULL
7. C 语言中系统的标准输入文件 (设备) 是指 (A)。
A. 键盘 B. 显示器 C. 软盘 D. 硬盘
8. 下述关于 C 语言文件操作的结论中, (B) 是正确的。
A. 对文件操作必须是先关闭文件
B. 对文件操作必须是先打开文件
C. 对文件操作顺序无要求
D. 对文件操作前必须先测试文件是否存在, 然后再打开文件
9. 利用 *fseek* 函数可以实现的操作是 (D)。
A. 改变文件的位置
B. 辅助实现文件的顺序读写
C. 辅助实现文件的随机读写
D. 以上答案都正确
10. *fscanf* 函数的正确调用形式是 (D)。
A. *fscanf*(文件指针, 格式字符串, 输出表列);
B. *fscanf*(格式字符串, 输出表列);
C. *fscanf*(格式字符串, 文件指针, 输出表列);
D. *fscanf*(文件指针, 格式字符串, 输入表列);
11. *fseek* 函数的正确调用形式是 (B)。
A. *fseek*(文件指针, 起始点, 位移量);
B. *fseek*(文件指针, 位移量, 起始点);
C. *fseek*(位移量, 起始点);
D. *fseek*(起始点, 位移量, 文件指针);
12. 缺省状态下, 系统的标准输出文件 (设备) 是指 (B)。
A. 键盘 B. 显示器 C. 软盘 D. 硬盘
13. 函数调用语句: *fseek(fp, -20L, SEEK_END)*; 的含义是 (C)。
A. 将文件位置指针移到距离文件头 20 个字节处
B. 将文件位置指针从当前位置向后移动 20 个字节
C. 将文件位置指针从文件末尾处向后退 20 个字节
D. 将文件位置指针移到离当前位置 20 个字节处
14. 函数 *ftell(fp)* 的作用是 (A)。
A. 得到文件当前位置指针的位置

- B. 移动流式文件的位置指针
- C. 初始化流式文件的位置指针
- D. 以上答案均正确

15. fwrite 函数的一般调用形式是 (D)。

- A. fwrite(buffer, count, fp, size);
- B. fwrite(fp, size, count, buffer);
- C. fwrite(fp, count, size, buffer);
- D. fwrite(buffer, size, count, fp);

16. 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, 则文件打开时的方式字符串应是 (B)。

- A. "ab+" B. "wb+" C. "rb+" D. "ab"

17. 在执行 fopen 函数时, 若执行不成功, 则该函数的返回值是 (D)。

- A. TRUE B. -1 C. 1 D. NULL

18. 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 (C)。

- A. 只写 B. 追加 C. 读或读写 D. 答案 B 和 C 都正确

19. 以下 fread 函数的调用形式中, 参数类型正确的是 (A)。

- A. fread(char *buf, int size, int count, FILE *fp)
- B. fread(FILE *fp, char *buf, int size, int count)
- C. fread(FILE *fp, int size, int count, char *buf)
- D. fread(int count, char *buf, int size, FILE *fp)

20. 函数 rewind 的作用是 (A)。

- A. 使位置指针重新返回到文件的开头
- B. 将位置指针指向文件中所要求的特定位置
- C. 使位置指针指向文件的末尾
- D. 使位置指针自动移至下一个字符位置

二、填空题

1. 以下程序用来统计文件中字符的个数。请填空。

```
#include "stdio.h"

void main()
{ FILE *fp; long num=0;
  if((fp=fopen("fname.dat", "r"))==NULL)
  { printf("Open error\n"); exit(0); }
  while( !feof(fp) )
  { if( fgetc(fp) ) num++; }
  printf("num=%ld\n", num);
}
```

2、以下程序实现文件的复制，请填写。

```
#include<stdio.h>
void fcopy(FILE *fout, FILE *fin)
{   char k;
    do
    {   k=fgetc( fin );
        if(!feof(fin)) break;
        fputc( k, fout );
    }while(1);
}
void main()
{   FILE *fin, *fout;
    if((fin=fopen("fin.dat","r"))==NULL)
    {   printf("Cannot open file fin.dat.\n");   exit(0);   }
    if((fout=fopen("fout.dat","w"))==NULL)
    {   printf("Cannot open file fout.dat.\n");   exit(0);   }
    fcopy(fout,fin);
    fclose(fin);
    fclose(fout);
}
```

3、feof(fp)函数用来判断文件是否结束，如果遇到文件结束，函数返回值为 1，否则为 0。

4、在 C 语言中，打开文件使用的函数是 fopen()。

5、定义 FILE *fp，这里 fp 是一个 指向 FILE 类型结构体的指针变量（简称文件指针）。

6、以下程序由键盘输入文件名，然后将输入的字符依次存放到该文件中，用#作为结束输入的标记，补足所缺语句。

```
#include<stdio.h>
#include<string.h>
void main()
{   FILE *fp;
    char ch, filename[10];
    printf("Input the name of file:\n");
    scanf("%s",filename);
    if(( fp=fopen(filename,"w") )==NULL)
    {   printf("Cannot open file.\n");   exit(0);   }
    printf("Enter string:\n");
    ch=getchar();
    while(ch!='#')
    {   fputc( ch,fp );
```

```

        ch=getchar();
    }
    fclose(fp);
}

```

7、函数调用语句: `fgets(buf,n,fp)`; 从 `fp` 指向的文件中读入 最多不超过 $n-1$ 个字符放到 `buf` 字符数组中, 函数返回值为 `buf` 的地址。

8、在 C 语言中, 文件的存取是以 字符 为单位的, 这种文件被称为 流式 文件。

9、设有以下结构体类型:

```

struct st
{
    char name[8];
    int num;
    float s[4];
}student[50];

```

并且结构体数组 `student` 中的元素都已有值, 若要将这些元素写到硬盘文件 `fp` 中, 请将以下 `fwrite` 语句补充完整:

```

fwrite(student, 50*sizeof(struct st), 1, fp);

```

10、在 C 程序中, 文件可以用 顺序 方式存取, 也可以用 随机 方式存取。

11、下面程序由终端键盘输入字符, 存放到文件中, 用 “!” 结束输入。在空白处填入适当内容:

```

#include<stdio.h>
void main()
{
    FILE *fp;
    char ch, fname[10];
    printf("Input name of file.\n");
    gets(fname);
    if((fp=fopen(fname,"w"))==NULL)
    {
        printf("Cannot open.\n"); exit(0);
    }
    printf("Enter data.\n");
    while( (ch=getchar()) != '!' ) fputc(ch,fp);
    fclose( fp );
}

```

12、在 C 语言中, 对文件的存取是以 字符 为单位, 即以 字节 为单位。

13、有一磁盘文件, 第一次使它显示在屏幕上, 第二次把它复制到另一文件中, 把该程序补充完整。

```

#include<stdio.h>
void main()

```

```
{ FILE *fp1,*fp2;
  fp1=fopen("file1.c","r");
  if(!fp1)
  { printf("Cannot open file file1.c"); exit(0); }
  fp2=fopen("file2.c","w");
  if(!fp2)
  { printf("Cannot open file file2.c"); exit(0); }
  while(!feof(fp1)) putchar(getc(fp1));
  rewind(fp1);
  while(!feof(fp1)) fputc(       fgetc(fp1)       , fp2);
  fclose(fp1); fclose(fp2);
}
```

