一、单选

1.以下说法中正确的是(　C　)

A)　C 语言程序总是从第一个的函数开始执行

B)　在 C 语言程序中,要调用的函数必须在 main()函数中定义

C)　C 语言程序总是从 main()函数开始执行

D)　C 语言程序中的 main()函数必须放在程序的开始部分

main()函数的位置可以任意。

2.下选项中不是 C 语句的是(　C　)。

A) {int i; i++; printf("%d\n",i);}

B) ;

C) a=5,c=10

D) { ; }

任何 C 表达式加上分号后，就构成一条 C 语句。而只有一个分号的语句称为空语句。

3.以下选项中不能作为 C 语言合法常量的是

(　A　)

A) 'cd'

B) 0.1e+6

C) "\a"

D) '\011'

常量分为

1：整型常量①十进制整常量：带正负号的整数，第一个数不能为 0。②八进制整常量：以 0 开头，其中只能出现 0~7 八个数字，通常是无符号数。③十六进制整常量：以 0X 或 0x 开头，可用 0~9，a~f 或 A~F(表示十进制值 10~15)。十六进制数通常用来表示地址，通常也是无符号数。

2：浮点型常量①十进制数形式②指数形式，如 6.5E-2(等于 6.5×10^-2) .34e+6(等于 0.34×10^6 ＋可省)，阶码标志 e(E)之前必须有数字，且 e 后面的指数必须为整数。

3：字符型常量：用单引号括起来的一个字符，还有一种特殊字符常量是以"\"开头的字符序列，即为转义字符。如'a' 'A' '\n'等。单引号、双引号、反斜杠和换行字符本身单独不能构成一个字符常量的字符，但可以和反斜杠一起组成转义字符。①简单转义字符。②八进制转义字符：由反斜杠和 1~3 个八进制数字构成。如'\071'③十六进制转义字符：由反斜杠和字母 x 和 1~2 个十六进制数字构成。如'\xEF'。

4：字符串常量：用一对双引号括起来的零个或多个字符组成的序列。如"a" "hello" "$43.23"。系统在储存字符串常量时自动在其末尾加上'\0'为结束标志。

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

A)　1

B)　2

C)　3

D)　不确定

赋值运算符的结合性——自右至左结合，赋值表达式的值等于赋值运算符右边表达式的值。而结果的类型由赋值运算符左边的变量的类型决定。

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

A)　int a=b=c=1;
B)　int a,b,c=1;
C)　a=b=c=1;
D)　int a=1,b=1,c=1;
6.字符串"\t\065\xff\n"中的字符数(不算’\0’)为(　D　)
A)　5
B)　14
C)　8
D)　4
注意：‘b’和“b”是完全不同的。前者是字符常量，在内存中占用字节数为 1；而后者是字符串常量，在内存中占用字节数为 2(因为有\0 的结束标志)。
计算字符数跟所占字节数都要算\0(除非题目明确说不算\0)，但是用 strlen 函数计算字节数时不算\0。
7.对两个静态函数 A 和 B 进行如下初始化：
char A[]="ABCDEF";
char B[]={'A','B','C','D','E','F'};
则下列叙述正确的是(　D　)。
A)　A 和 B 完全相同
B)　A 和 B 只是长度相等
C)　A 和 B 不相同，A 是指针数组
D)　A 数组长度比 B 数组长
A 为字符串，结尾有\0 为结束标志，一共占 7 个字符。而 B 为字符数组，一共占 6 个字符。
8.若有以下定义：
char a;int b;
float c;double d;
则表达式 a*b+d-c 值的类型为(　D　)
A) float
B) int
C) char
D) double
char 型、short 型数据在运算前须先转换为 int 型。unsigned short 型先转换为 unsigned int 型。float 型先转换为 double 型。int 与 double 进行运算时先将 int 转换为 double。自动转换后所有数都按 double 型进行运算。
9.在 C 语言中，运算对象必须是整型数的运算符是(　A　)
A)　%
B)　/
C)　%和\
D)　**
%求余运算符的两个操作对象都必须是整数。 结果的符号与%左边的操作数的符号相同。如-45%8=-5　45%-8=5
10.有以下程序
main()

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

程序运行后的输出结果是

(　C　)

A) 2,3,3

B) 2,3,2

C) 2,3,1

D) 2,2,1

①i=1 j=++i 则 j=2 i=2 ++i=2

②i=1 j=i++ 则 j=1 i=2 i++=1 区分 i i++ ++i 的区别

赋值运算符的优先级高于逗号运算符(逗号运算符优先级最低)，所以先对 z 赋值，相当于是(z=x++),y++…

++和--的优先级高于基本算术运算符，与-(取负)的优先级相同。结合方向为自右至左。如-j+++-i 相当于-(j++)+(-i)

在调用函数时，多数系统对系统函数参数的求值顺序是自右至左。如

a=5 输出 a,a++为 5,6 输出 a,++a 为 6,6

特例 输出 a,(a++)+(a++)为 7,10。a 按正常顺序计算得 7，但是算第二个式子将(a++)看成两个相同部分即 5+5=10，同理如果是 3 个(a++)为 8,15。

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

A) x%2==1

B) x/2

C) x%2!=0

D) x%2==0

!=是不等于 ==是等于(区别=为赋值运算符)

在关系运算和逻辑运算中，表达式为真用 1 表示，假用 0。

12.以下程序的输出结果是(　C　)。

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

A)　0

B)　1

C)　3

D)　不确定的值

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

A) -3

B) 9

C) -12

D) 6

先进行“a-=a*a”相当于 a=a-a*a，此时等号右边的 a 为 3，运算后的 a=3-3*3=-6

再进行“a+=-6”相当于 a=a+(-6)，此时等号右边的 a 为-6，运算后的 a=-6-6=-12
14.已知字符 A 的 ASCII 码为十进制的 65 下面程序的输出是(A)
```
main()
{
char ch1,ch2;
ch1='A'+'5'-'3';
ch2='A'+'6'-'3';
printf("%d,%c\n",ch1,ch2);
}
```
A) 67,D
B) B,C
C) C,D
D) 不确定的值
ch1 是以有符号十进制型式输出，而 ch2 仍以字符输出，且 68 所对应的控制字符为 D
15.有以下程序
```
main()
{
   char    a='a',b;
   printf("%c,",++a);
   printf("%c\n",b=a++);
}
```
程序运行后的输出结果是
(　A　)
A) b,b
B) b,c
C) a,b
D) a,c
a 的 ASCII 码值加 1 就是 b 的 ASCII 码值。所以++a 为 b，且 a 的值为 b，而 b=a++是先将 a 的值赋值给 b，即 b=b，若继续输出 a 的值则为 c。
16.设 a、b、c、d、m、n 均为 int 型变量，且 a=5、b=6、c=7、d=8、m=2、n=2，则逻辑表达式(m=a>b)&&(n=c>d)运算后，n 的值为(　C　)
A) 0
B) 1
C) 2
D) 3
因为 a>b 为假，则 m=0。又因为逻辑运算中有短路求值，既然 m=0 那么整个表达式一定为 0，就不用继续算 n 的值了。所以 n 保持原来的 2 不变。
17.设 x、y、t 均为 int 型变量，则执行语句：x=y=3;t=++x||++y;后，y 的值为(　C　)
A) 不定值
B) 4
C) 3
D) 1
短路求值。

*18.有以下程序段
int　k=0;
while(k=1)k++;
while 循环执行的次数是(　A　)
A) 无限次
B) 有语法错，不能执行
C) 一次也不执行
D) 执行 1 次
while 处的语句是一个赋值语句而不是判断语句，无法判断什么时候结束。
19.能正确表示逻辑关系“a>=10 或 a<=0”的 C 语言表达式是(　D　)
A) a>=10 or a=10
B) a>=0|a<=10
C) a>=10&&a<=0
D) a>=10||a<=0
20.表示关系 x≤y≤z 的 c 语言表达式为(　A　)
A) (x<=y)&&(y<=z)
B) (x<=y)AND(y<=z)
C) (x<=y<=z)
D) (x<=y)&(y<=z)
21.设有：int a=1, b=2, c=3, d=4, m=3, n=3;　执行 (m=a>b)||(n=c>d)后 n 的值为(　A　)。
A)　0
B)　1
C)　2
D)　3
与 16 题类似。
22.以下关于逻辑运算符两侧运算对象的叙述中正确的是
(　D　)
A) 只能是整数 0 或 1
B) 只能是整数 0 或非 0 整数
C) 可以是结构体类型的数据
D) 可以是任意合法的表达式
23.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　)
A)　200
B)　100
C)　100 200
D)　输出格式符不够，输出不确定的值
此处为一个逗号表达式，求解过程为：先求表达式 1，再求表达式 2…最后求表达式 n，整个逗号表达式的值为表达式 n 即最后一个表达式的值。
24.以下程序的输出结果是(　D　)
main()
　{　int　　k=17；

```
    printf("%d，%o，%x \n"，k，k，k)；
}
```
A)　17，021，0x11
B)　17，17，17
C)　17，0x11，021
D)　17，21，11
1、int①d 以有符号十进制输出②o 以无符号八进制输出③x 或 X 以无符号十六进制输出④u 以无符号十进制输出。
注意：在 C 语言格式字符的输出中，不输出前导符号 0 和 0x。所以此处不选 A。
2、float①f 以小数输出(隐含输出 6 位)②e 或 E 以指数输出③g 或 G 按 f 和 e 中数值宽度最小的输出
3、char①c 输出一个字符②s 输出字符串
25.x、y、z 被定义为 int 型变量，若从键盘给 x、y、z 输入数据，正确的输入语句是(B)。
A)　INPUT x、y、z;
B)　scanf("%d%d%d",&x,&y,&z);
C)　scanf("%d%d%d",x,y,z);
D)　read("%d%d%d",&x,&y;&z);
由键盘输入时一定要取地址即加&。
26.有以下程序
```
main()
{
  int  i;
  for(i=0;i<3;i++)
    switch(i)
    {
    case 0: prinft("%d",i);
    case 2: prinft("%d",i);
    default: prinft("%d",i);
    }
}
```
程序运行后的输出结果是(　C　)
A) 022111
B) 021021
C) 000122
D) 012
“case 常量表达式”部分只起语句标号的作用，而不进行条件判断。所以当 switch 后面的值等于某个 case 后面的值时，开始输出，并依次往下进行，直到遇到 switch 语句的右花括号或“break”语句为止。
27.C 语言中用于结构化程序设计的三种基本结构是(　A　)
A) 顺序结构、选择结构、循环结构
B) if、switch、break
C) for、while、do-while
D) if、for、continue

28.若执行下面的程序时从键盘上输入 5 则输出是(B)

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

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

x=5 时，x++为 5 但 x=6，所以执行 else printf。又因为执行过 if 语句所以 x 的值已变成 6，所以 x--为 6，而最终的 x 为 5。

29.在嵌套使用 if 语句时，C 语言规定 else 总是(C)

A) 和之前与其具有相同缩进位置的 if 配对(可能位置相同的 if 已配有 else)
B) 和之前与其最近的 if 配对
C) 和之前与其最近的且不带 else 的 if 配对
D) 和之前的第一个 if 配对

30.设有以下程序片段:

```
switch(X)
{
    case 'A':   printf("A");
    case 'B':   printf("B");
    default:   printf("error");
}
```

假设 X='A',程序输出结果是(D)

A) A
B) B
C) error
D) ABerror

31.读程序：

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

上面程序的输出结果是(C)

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

32.下述语句执行后变量 k 的值是(B)。

```
int k=1;
while(k++<10);
```

A) 10

B) 11
C) 9
D) 此为无限循环值不定
33.以下程序段的执行结果是(A)。
int x=5;
do{printf("%2d\n",x--);}
while(!x);
A) 5
B) 无任何输出
C) 4
D) 陷入死循环
34.设变量已正确定义，则以下能正确计算 f=n!的程序段是
(D)
A) f=0;
for(i=1;i<=n;i++) f*=i;
B) f=1;
for(i=1;i<n;i++) f*=i;(若改成 i<=n 则 B 也对)
C) f=1;
for(i=n;i>1;i++) f*=i;
D) f=1;
for(i=n;i>=2;i--) f*=i;
35.以下程序的输出结果是(D)。

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

A) 101
B) 6
C) 5
D) 4
for 语句中使用 continue 语句，会转移到 for 语句中的语句 3，再执行语句 2。
36.下列合法的数组定义是(D)。
A) int a[]="string";
B) int a[5]={0,1,2,3,4,5};(错误的原因是因为初值个数大于数组长度)

C)　char a="string";
D)　char a[]={0,1,2,3,4,5};
37.以下定义语句中，错误的是(　D　)
A) int　a[]={1,2};
B) char　*a[3];
C) char　s[10]="test";
D) int　n=5,a[n];
方括号中的常量表达式中不允许包含变量。
38.以下正确的数组定义语句是(　D　)。
A)　int y[1][4]={1,2,3,4,5};
B)　float x[3][]={{1},{2},{3}};
C)　long s[2][3]={{1},{1,2},{1,2,3}};
D)　int m[1][4]={4};
二维数组的初始化时，初值个数可以小于数组长度但不能大于。部分赋初值时其余元素的值也被初始化为 0。而且可以省略第一维的长度，但不能省略第二维的长度。
39.以下程序的输出结果是(　C　)。

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

A)　258
B)　741
C)　852
D)　369
40.　下面的程序中(　C　)有错误(每行程序前面的数字是行号)。

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

A)　没有错误
B)　第 4 行错误
C)　第 6 行错误
D)　第 9 行错误
用 scanf 键盘输入时要取地址加&

41.若有定义语句：int　a[3][6]；，按在内存中的存放顺序，a 数组的第 10 个元素是(　B　)

A) a[0][4]
B) a[1][3]
C) a[0][3]
D) a[1][4]

42.以下数组定义中不正确的是(　D　)

A) int　a[2][3];
B) int　b[][3]={0,1,2,3};
C) int　c[100][100]={0};
D) int　d[3][]={{1,2},{1,2,3},{1,2,3,4}};

43.以下程序的输出结果是(　A　)。

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

A)　0650
B)　1470
C)　5430
D)　输出值不定

45.以下程序的输出结果是

(　B　)

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

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

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

A)　4 个字节
B)　5 个字节
C)　6 个字节
D)　7 个字节

47.以下选项中，不能正确赋值的是(　A　)。

A)　char s1[10];s1="Ctest";
B)　char s2[]={'C','t','e','s','t'};
C)　char s3[20]="Ctest";
D)　char *s4="Ctest\n";

char s1[10]是一个数组，s1 的数据类型是 const char * 也就是常量指针。无法对一个常量进行赋值，只能对它指向的内存进行赋值。

48.给出以下定义：

char x[]="abcdfeg";
char y[]={'a','b','c','d','e','f','g'};
则正确的叙述为(C　)。
A)　数组 x 和数组 y 等价
B)　数组 x 和数组 y 长度相同
C)　数组 x 的长度大于数组 y 的长度
D)　数组 x 的长度小于数组 y 的长度
49.设有 char str[]="Beijing";则执行
printf("%d\n",strlen(strcpy(str,"China")));
后的输出结果为(A)。
A) 5
B) 7
C) 12
D) 14
strlen 只算实际字符数不算\0。strcpy 将 china 复制给 str 字符数组。
strcpy(字符数组名 1，字符串 2，要复制的字符数);
50.若要求从键盘读入含有空格字符的字符串，应使用函数(　B　)
A) getc()从终端输入单个字符
B) gets()s 是指输入字符串
C) getchar()从终端输入单个字符。
D) scanf()从键盘输入数据，其中的空格是用来区别不同字符串。
51.有语句：
char str1[10],str2[10]={"books"};
则能将字符串 books 赋给数组 str1 的正确语句是(B　)。
A)　str1={"Books"};
B)　strcpy(str1,str2);
C)　str1=str2;
D)　strcpy(str2,str1);
52.以下语句的输出结果是(　A　)
　　printf("%d\n",strlen("\tc\065\xff\n"));
A)　5
B)　14
C)　8
D)　输出项不合法，无正常输出
53.若有语句 int *point, a=4; 和 point=&a; 下面均代表地址的一组选项是(　D　)。
A)　a, point, *&a
B)　&*a, &a, *point
C)　*&point, *point, &a
D)　&a, &*point, point
54.设已有定义：float x;，则以下对指针变量 p 进行定义且赋初值的语句中正确的是(　D　)
A) float *p = 1024;
B) int *p = (float)x;
C) float p = &x;

D) float *p = &x;
给指针变量赋值①使用地址运算符(&)将变量的地址取出赋给指针变量使指针变量变为一具体变量。如 D 选项。
②将一个已有具体指向的指针变量赋值给另一个指针变量。如 55 题 C 选项。
55.设有语句：int a=1,b=2,*p1=&a,*p2=&b; 以下可使指针 p1 指向变量 b 的赋值语句是(C)。
A) p1=*p2
B) *p1=p2
C) p1=p2
D) *p1=*p2
56.有以下程序

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

程序运行后的输出结果是(A)
A) 1,2,3,4,5,6,7,8,9,0,
B) 2,3,4,5,6,7,8,9,10,1,
C) 0,1,2,3,4,5,6,7,8,9,
D) 1,1,1,1,1,1,1,1,1,1,
数组变量名代表数组的首地址，即第 0 号元素的地址。a+10 表示 a[10]的地址。
57.若已定义 char s[10]; 则在下面表达式中不能表示 s[1]的地址的是(B)。
A) s+1
B) s++
C) &s[0]+1 相当于 A 选项 s+1
D) &s[1]
58.在以下选项中，操作不合法的一组是(B)。
A) int x[6], *p; p=&x[0];
B) int x[6], *p; *p=x;若是以数组变量名对指针变量进行赋值则不用加*，如 C 选项。
C) int x[6], *p; p=x;
D) int x[6],p; p=x[0];
59.有以下程序

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

程序运行后的输出结果是
(A)
A) 15

B) 16
C) 12
D) 59
60.以下程序运行后,如果从键盘上输入:
book <回车>
book <空格><回车>
则输出的结果是(B)
#include
main()
{char a1[80],a2[80],*s1=a1, *s2=a2;
gets(s1);
gets(s2);
if(!strcmp(s1,s2))
printf("*");
else
printf("#");
printf("%d\n",strlen(strcat(s1,s2)));
}
A) *8
B) #9
C) #6
D) *9
!strcmp(s1,s2)是判断较两个字符串是否相等，若相等则输出*，不等则输出#。strcat 是连接两个字符串，再用 strlen 算字符串长度(不算\0)，即“bookbook(空格)”的长度。s1 与 s2 明显是两个不同的字符串。
*61.以下语句或语句组中，能正确进行字符串赋值的是(D)
A) char *sp; *sp="right!";
B) char s[10]; s="right!";改为 char s[10]=“right!”;
C) char s[10]; *s="right!";
D) char *sp="right!";
62.下面程序段的运行结果是(C)。
char *p= "abcdefgh";
p+=2;
printf("%d\n", strlen(strcpy(p, "ABCD")));
A) 6
B) 12
C) 4
D) 7
63.下面程序段的运行结果是(B)。
char str[ ]= "ABC", *p=str;
printf("%d\n", *(p+3));
A) 67
B) 0
C) 字符'C'的地址

D)　字符'C'
64.以下程序运行后，输出结果是(　B　)
```
main()
{　char *s="abcde";
　s+=2;
　printf("%d\n"，s);}
```
A) cde
B) 字符 c 的 ASCII 码值
C) 字符 c 的地址
D) 出错
66.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　)
A) 2
B) 3
C) 4
D) 5
67.若有语句：char *line[5]；，以下叙述中正确的是(　A　)
A) 定义 line 是一个数组，每个数组元素是一个基类型为 char 的指针变量
B) 定义 line 是一个指针变量，该变量可以指向一个长度为 5 的字符型数组
C) 定义 line 是一个指针数组，语句中的*号称为间址运算符
D) 定义 line 是一个指向字符型函数的指针
68.设有如下定义: char *aa[2]={"abcd","ABCD"}；则以下说法中正确的是(　B　)。
A)　aa 数组成元素的值分别是"abcd"和 ABCD"
B)　aa 是指针变量,它指向含有两个数组元素的字符型一维数组
C)　aa 数组的两个元素分别存放的是含有 4 个字符的一维字符数组的首地址
D)　aa 数组的两个元素中各自存放了字符串"abcd"和"ABCD"的首地址
69.以下函数调用语句中含有(　B　)个实参。
```
fun((exp1,exp2),(exp3,exp4,exp5));
```
A)　1
B)　2
C)　4
D)　5
函数实参表中参数可以是常数、变量或其他构造类型数据及表达式。各实参之间用逗号隔开。
70.下列函数定义中会出现编译错误的是(B)
A) max(int x, int y, int* z)
{ *z = x>y?x:y; }
B) int max(int x, y)
{
int z;

```
z = x>y?x:y;
return z;
}
```
C) max(int x, int y)
```
{
int z;
z = x>y?x:y;
return(z);
}
```
D) int max(int x, int y)
```
{ return (x>y ? x : y); }
```
71.以下所列的各函数首部中，正确的是(　C　)

A) void play(var a :integer,var b:integer)

B) void play(int a,b)

C) void play(int a,int b)

D) Sub play(a as integer,b as integer)

72.C 语言中，函数值类型的定义可以缺省，此时函数值的隐含类型是(　B　)

A) void

B) int

C) float

D) double

73.下面的函数调用语句中 func 函数的实参个数是

(　A　)

```
func(f2(v1, v2), (v3, v4, v5), (v6, max(v7, v8)));
```
A) 3

B) 4

C) 5

D) 8

74.有以下程序

```
fun(int    x, int    y)
{ static int    m=0, i=2;
   i+=m+1;       m=i+x+y;       return m;
}
main()
{ int    j=1, m=1, k;
   k=fun(j,m);       printf("%d,",k);
   k=fun(j,m);       printf("%d\n",k);
}
```
执行后的输出结果是

(　B　)

A) 5, 5

B) 5, 11

C) 11, 11

D) 11, 5

第一轮输出，由题得 f(j,m)就是 fun(x,y)，即 x=1，y=1。而 k 就等于 fun 下面函数中的 m。所以 i+=m+1 就是 i=i+m+1=2+0+1=3，k=m=i+x+y=3+1+1=5。第二轮输出，i=i+m+1=3+5+1=9，k=m=i+x+y=9+1+1=11。

75.以下函数值的类型是(　A　)

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

A) int
B) 不确定
C) void
D) float

定义函数的时候没定义函数返回值的类型，所以默认 fun 函数的返回值类型为 int。即使在函数内定义的 y 的类型为 float 型，但是由于 y 是返回值，且返回值的类型是 int，所以系统会进行自动转换，所以函数值的类型是 int。若想为 float 则必须在 fun 前面定义 float。

76.以下程序的输出结果是

(　C　)

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

A) 0
B) 2
C) 31
D) 无定值

定义的函数 fun 没有返回值，而且参数不是引用型的，所以形参的改变不会影响实参，即 a 的值，输出的是 31。区分 74 题，有 return。

77.有以下程序

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

执行后输出的结果是(　D　)

A) 3,4,5

B) 5,3,4
C) 5,4,3
D) 4,3,5
78.在调用函数时，如果实参是简单变量，它与对应形参之间的数据传递方式是
(　C　)
A) 地址传递
B) 单向值传递
C) 由实参传给形参，再由形参传回实参
D) 传递方式由用户指定
79.有以下程序

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

程序运行后的输出结果是(　B　)
A) 1,2,3,4,5,6,7,8,9,10,
B) 1,2,6,8,10,12,7,8,9,10,
C) 1,2,3,4,10,12,14,16,9,10,
D) 1,2,6,8,10,12,14,16,9,10,
数组的下角标从 0 开始，所以代入 f(a)函数内，函数 for 循环 i=2，应从第三个数 a[2]开始执行，根据函数 for 循环条件是 i<6，就是到第六个数 a[5]结束。即 a[2]~a[5]以 2 倍输出，其他按原来的值输出。
80.在 C 语言中，函数返回值的类型最终取决于
(　A　)
A) 函数定义时在函数首部所说明的函数类型
B) return 语句中表达式值的类型
C) 调用函数时主调函数所传递的实参类型
D) 函数定义时形参的类型
81.调用一个函数，此函数中没有 return 语句，下列说法正确的是：该函数(　A　)。
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
82.请读程序

```
#include
f(char *s)
{ char *p=s;
```

```
while( *p!='\0') p++;
return(p-s);
}
main()
{ printf("%d\n",f("ABCDEF"));}
```

上面程序的输出结果是(B)

A) 3

B) 6

C) 8

D) 0

字符在内存中占一个字节的存储空间，按字符的存储顺序，其地址依次递增，在函数 f 中循环过后，p 指向字符串的结束位置，s 指向字符串首地址，p-s 即为字符串长度。

83.若有以下调用语句,则不正确的 fun 函数的首部是(　D　)

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

A)　void fun(int m, int x[])

B)　void fun(int s, int h[41])

C)　void fun(int p, int *s)

D)　void fun(int n, int A)

&a[9]表示对变量 a[9]的引用，与它对应的形参必须是表示地址的变量，而 D 的第 2 个形参是一个普通变量。

84.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(C)

A) 10

B) 9

C) 7

D) 8

本题中定义了一个全局变量 a 和局部变量 a。在主函数中定义一个整型变量 s 并赋初值为 0，接着定义了一个局部变量 a 并赋值为 5，由于该局部变量的和全局变量 a 同名，全局变量在复合语句中将不起作用，然后调用函数 f(&a)(该函数的作用是返回存储变量 a 中的值，并让该值加 1，分析程序可知返回值为 5，并让 a 值加 1 变为 6)，并将返回值加到 s 中。此时 s 的值为 5，再执行该复合语句外的其他语句，同样调用函数 f(&a)，传递的参数是全局变量 a，故其返回值为 2，加到 s 中后 s 的值变为 7，最后输出的 s 值为 7。

85.有以下程序

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

程序执行后的输出结果是

(　C　)

A) 123,abc

B) abc,123

C) 1bc,a23

D) 321,cba

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

86.以下程序的输出结果(　B　)

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

A)　5,2,3

B)　-5,-12,-7

C)　-5,-12,-17

D)　5,-2,-7

87.已定义以下函数

```
int fun( int   *p)
{   return   *p;   }
```
fun 函数返回值是
(B)
A) 不确定的值
B) 一个整数
C) 形参 p 中存放的值
D) 形参 p 的地址值
函数值的类型应当是在定义函数时指定的。在定义函数时对函数值说明的类型一般和 return 语句中的表达式类型一致，即函数类型决定返回值的类型。此题中定义函数类型为 int 型，故函数返回值也为整型。
88.已定义以下函数：
```
fun (char* p2, char* p1)
{ while ((*p2=*p1) != '\0') {pl++;p2++;} }
```
函数的功能是
(A)
A) 将 p1 所指字符串复制到 p2 所指内存空间
B) 将 pl 所指字符串的地址赋给指针 p2
C) 对 pl 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'
while 循环语句的功能是将 p1 所指存储单元的内容赋值给 p2 所指的存储单元，然后 p1++、p2++，分别指向下一个存储单元，直到 p1 指向符号串的结束字符'\0'为止。故函数的功能是将 p1 所指字符串复制到 p2 所指内存空间。
区别：(*p)++，是先取指针 P 的值，然后对其值进行++运算，
*（p++），是先对指针 P 进行++运算，然后再取取值，
p++，同(p++)，因为按优先级来看，*和++是同级，他们都是从右到左的顺序进行运算，所以先++，再*。
89.有以下程序
```
#include <stdio.h>
int fun(int a, int b)
{   if (b==0)     return a;
    else        return (fun(--a, --b));
}
main()
{   printf("%d\n", fun(4, 2)); }
```
程序的运行结果是(B)
A) 1
B) 2
C) 3
D) 4
90.在函数调用过程中，如果函数 funA 调用了函数 funB，函数 funB 又调用了函数 funA，则(B)
A) 称为函数的直接递归调用
B) 称为函数的间接递归调用

C) 称为函数的循环调用
D) C 语言中不允许这样的递归调用
如果在调用函数 A 时，函数 A 调用了本身，称为函数的直接递归调用；如果函数 A 调用了函数 B，函数 B 又调用了函数 A，称为函数的间接递归调用。
91.以下叙述中正确的是(B)
A) 全局变量的作用域一定比局部变量的作用域范围大
B) 静态(static)类别变量的生存期贯穿于整个程序的运行期间
C) 函数的形参都属于全局变量
D) 未在定义语句中赋初值的 auto 变量和 static 变量的初值都是随机值
若在函数中定义与全局变量名字相同的局部变量，则全局变量在该函数中将不起作用，因此全局变量的作用域并不一定比局部变量的作用域大，故选项 A 不正确；静态变量一旦定义，将在整个程序的运行期间都存布，故选项 B 正确；函数的形参只在函数调用的时候分配存储空间，在退出函数时收回存储空间，因此是局部的，故选项 C 不正确；没有赋值的 auto 型变量的初值是随机的，没有赋值的 static 型变量的初值是 0，故选项 D 不正确。
92.以下程序的输出结果是
(B)

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

A) 100200
B) 57
C) 200100
D) 75
93.以下程序的输出结果是
(D)

```
int f()
{  static int i=0;
   int  s=1;
   s+=i;  i++;
   return  s;
}
main()
{  int  i,a=0;
   for(i=0;i<5;i++) a+=f();
   printf("%d\n",a);
}
```

A) 20
B) 24

C) 25
D) 15
函数 f 中变量 i 为静态变量，函数 f 调用结束后变量 i 所占据的存储单元不会释放，而在主函数中 f 被调用 5 次，具体过程如下：
第 1 次调用 f：s=s+i=1+0=1，i=i+1=1，主函数中 a=a+f()=0+1=1
第 2 次调用 f：s=s+i=1+1=2，i=i+1=2，主函数中 a=a+f()=1+2=3
第 3 次调用 f：s=s+i=1+2=3，i=i+1=3，主函数中 a=a+f()=3+3=6
第 4 次调用 f：s=s+i=1+3=4，i=i+1=4，主函数中 a=a+f()=6+4=10
第 5 次调用 f：s=s+i=1+4=5，i=i+4=5，主函数中 a=a+f()=10+5=15
所以 printf 语句的输出结果为 15。
二、多选
1.下列定义变量的语句中正确的是(　A B C　)
A) int　_int;
B) double　int_;
C) char　For;
D) float　US$;
标识符只能是字母(A~Z，a~z)、数字(0~9)、下划线(_)组成的字符串，并且其第一个字符必须是字母或下划线。
2.设有定义：int k=1, m=2；float f=7;，则以下选项中符合 C 语言语法的表达式是(A B D)
A) k=k>=k
B) -k++
C) k%int(f)
D) k<>m
强制类型转换的一般格式为：(数据类型)表达式，所以 int(f)应该写成：(int)f，正确写法：k%(int)f。
3.若有定义语句：double x[5]={1.0,2.0,3.0,4.0,5.0}, *p=x;则正确引用 x 数组元素的是(A C D)
A) *p
B) x[5]
C) *(p+1)
D) *x
本题定义了一个有 5 个数组元素的一维数组 x 和指针变量 p，且将 x 的首地址(即 x[0])的地址)赋给了指针变量 p，即 p 指向了 x 数组的第 0 号元素，所以*p 的值为 x[0]的值，选项 A 的引用正确；由于 p+1 就是 x[1]的地址，所以*(p+1)所指向的数组元素为 x[1]，故选项 C 的引用正确；由于 x 就是数组 x 的首地址，所以*x 即对 x[0]的引用，所以选项 D 引用正确；由于数组最大下标是元素个数减一，所以选项 B 的引用越界。
4.若要求定义具有 10 个 int 型元素的一维数组 a，则以下定义语句中正确的是(A B C)
A) #define　N　10
int　a[N];
B) #define　n　5
int　a [2*n];

C) int a[5+5];
D) int n=10,a[n];
在选项 D 中， n 是一个整型的变量。 C 语言规定，在一维数组的定义中，其下标只能是常量表达式，不能包含变量。
5.按照 C 语言规定的用户标识符命名规则，能出现在标识符中的是(A C D)
A) 大写字母
B) 连接符
C) 数字字符
D) 下划线
6.以下不合法的字符型常量是(B D)
A) '\x13'
B) '\081'
C) '\065'
D) "\n"
7.以下合法的字符常量是(B C D)
A) '\018'
B) '\"'
C) '\\'
D) '\xcc'
8.以下叙述中正确的是(A B D)
A) C 语句必须以分号结束
B) 复合语句在语法上被看作一条语句
C) 空语句出现在任何位置都不会影响程序运行
D) 赋值表达式末尾加分号就构成赋值语句
9.以下合法的赋值语句是(A D)
A) n=(i=2,++i);
B) j++;
C) ++(i+1);
D) x=j>0;
10.以下能正确定义二维数组的选项是(A B)
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}};
11.若有定义：int aa[8]；则以下表达式中能代表数组元 aa[1]的地址的是
(A B D)
A) &aa[0]+1
B) &aa[1]
C) &aa[0]++
D) aa+1
C 中＆小于后置＋＋的优先级所以&aa[0]++等价于＆(aa[0]++)所以先对 aa[0]中的元素值加 1 后，在对 aa[0]取址，结果还是 aa[0]的地址。类似于 i++，i 的值变为 i+1，但 i++的值仍未 i。
12.已有定义：int i,a[10],*p; ，则不合法的赋值语句是(A B)

A) p=100;
B) p=a[5]
C) p=&a[2]+2
D) p=a+2;
因为 p 指向的是地址，只能同类型赋值，而四个选项中只有 CD 是地址。
13.以下能正确进行字符串赋初值的语句是(B C D)
A) char str[5]="good!";
B) char str[]="good!";
C) char *str="good!";
D) char str[5]={‘g',‘o',‘o',‘d'};
A 选项中 good!后面还有终止符\0，所以长度为 6，而数组长度只有 5。
14.已知大写字母 A 的 ASCⅡ码是 65，小写字母 a 的 ASCⅡ码是 97。以下能将变量 c 中的大写字母转换为对应小写字母的语句是(　A B C　)
A) c=(c-'A')%26+'a'
B) c=c+32
C) c=c-'A'+'a'
D) c=('A'+c)%26-'a'
这里等号右边的 c 都是 c 的大写字母，ASCII 码值为 67。
15.设有以下定义和语句
char　str[20]="Program", *p;
p=str;
则以下叙述中错误的是(D)
A) *p 与 str[0]中的值相等
B) 可以执行 p++操作
C) 可以执行 str++操作
D) str 数组长度和 p 所指向的字符串长度相等
p 是指向字符串首地址，和 str 相同。而*p 是指 p 指向地址所储存的字符。
16.对于下面①，②两个循环语句，正确的描述是(A B)。
① while(1);
② for(; ;);
A) ①是无限循环
B) ②是无限循环
C) ①循环一次
D) ②循环一次
17.以下错误的函数定义形式是(B C D)
A) double fun (int x , int y)
B) double fun (int x ; int y)
C) double fun (int x , int y) ;
D) double fun (int x , y) ;
函数原型说明：类型说明符 被调函数名 (类型 形参,类型 形参…)
或者为：类型说明符 被调函数名 (类型,类型…)
18.对于基本类型相同的两个指针变量之间，可以进行的运算是(A B D)
A) <
B) =

C)　+
D)　-
A 选项两个同类型的指针间可以比较大小，比较原则应该是按照实际内存的高低位比较的。B 选项赋值。D 选项两个相同指针变量相减可以获得在之间相隔的同类型元素个数（在某个类型的数组中的应用）。
19.能把字符串:Hello!赋给数组 b 的语句是(A C D)
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!";
B 选项中的 b 不是代表整个数组而是代表数组首地址所以错误。字符数组初始化有两种方法：一种是逐个字符赋值，要加花括号；另一种是用字符常量对整个数组赋值，不用加花括号。C 是字符串拷贝函数。函数格式：char *strcpy (char *s1, const char *s2);
20.以下能对二维数组 a 进行正确初始化的语句是(A D)
A)　int a[2][3]={0};
B)　int a[2][ ]={{1, 2}, {0}};
C)　int a[2][3]={{1, 2}, {3, 4}, {5, 6}};
D)　int a[ ][3]={1, 2, 3, 4, 5, 6};
第一维的长度可以省略。而 C 选项的长度超过了数组长度所以错误，如果改为 a[3][2]则正确。
21.sizeof(double)是(C D)
A) 一种函数调用
B) 一个双精度型表达式
C) 一个整型表达式
D) sizeof 是运算符
sizeof 是计算在内存中所占空间，最终输出的是一个整型数。
22.设 a 为整型变量，能正确表达数学关系：10<a<15 的 C 语言表达式是(B C)
A)　10<a<15
B)　a==11||a==12||a==13||a==14
C)　a>10&&a<15
D)　(a<10)||(a>15)
三、判断
1.C 程序的基本组成单位是函数（√）一个 C 语言源程序可以由一个或多个源文件组成，而每个源文件可由一个或多个函数组成。
2.每个 C 程序中都必须要有一个 main()函数（√）每个 C 程序必须有且只有一个 main()函数，它代表程序开始执行的起始位置。
3.C 程序中注释部分可以出现在程序中任意合适的地方（√）
4.C 程序的执行总是从 main 函数开始，在 main 函数结束（√）
5.++(i+1);是非法的赋值语句（√）++，--是单目运算符，其优先级高于基本的算术运算符，与单目运算符-(取负而不是减号)的优先级相同。结合方式是“自右至左”。而算术运算符的结合方式为“自左至右”。正确应为(++i)+1。
6.C 语言中有逻辑类型（X）在 C 语言中没有逻辑类型，逻辑类型用整型来表示，C 语言中没有集合类型。

7.可以用关系运算符对字符串的大小进行比较 (X) 字符串本身不存在大小之分，只能对其所占字节大小进行比较。
8.设有定义语句：char b= '\123';则变量 b 包括 4 个字符 (X) 变量 b 是字符型常量，为单个字符。
9.若有定义语句：char s[10]="1234567\0\0";，则 strlen(s)的值是 9 (X) “\0”是终止符，而 strlen 是不算终止符长度的，所以最后的值应该为 7。
10.a=b+c=1 是正确的赋值表达式 (X) 赋值运算符左边的量必须是变量，不能是常量或用上述运算符结合起来的表达式。
11.对于单目运算符++、--，它们的运算对象可以是任何变量和常量 (X) 只能用于变量，而不能用于常量或表达式。
12.表达式：10！=9 的值是 true (X) 表达式的值应该为 0 或 1，而此表达式为真，值应该为 1 而非 true。
13.表达式 1||2||3||4 的值是 1。(√)
14.sizeof(float)的值是 4 (√)①float 占 4 个字节(32 位) ②double 占 8 个字节(64 位) ③long double 占 16 个字节(128 位)
15. if(x<y) {x++;y++;} 是正确的 if 语句。(√)
16. 用 do-while 语句构成的循环,在 while 后的表达式为零时结束循环 (√)
17.对 for(表达式 1； ； 表达式 3)可理解为 for(表达式 1；0；表达式 3) 。(X) 空语句和表达式 0 不相同
18.break 语句只能用于 switch 语句体中 (X) break 语句的功能是跳出正在执行的条件语句或循环语句。它可以出现在 switch 语句中，也可以出现在循环语句中。continue 语句只是结束本次循环，即跳过本次循环体中余下尚未执行的语句，接着再一次进行循环的条件判断。
19.当程序执行中，数组元素的下标超出所定义的下标范围时，系统将给出“下标越界”的出错信息 (X)
20.假定 int 类型变量占用两个字节，其有定义：int x[10]={0,2,4};，则数组 x 在内存中所占字节数是 6 (X) 数组长度为 10，但给数组赋初值只有 3 个值，长度小于数组长度所以系统会自动补上 0，字节数应为 20。注意所赋初值的个数不能超过数组长度，不然就出错。字符型数组是在末尾补上\0。
21.若有定义：int a[2][3];对 a 数组元素正确引用的是 a[2][3] (X) 数组 a[2][3]包括元素 a[0][0]，a[0][1]，a[0][2]，a[1][0]，a[1][1]，a[1][2]，此处的引用是下标越界。
22.若有说明 int s[3][4]={0};则只有元素 s[0][0] 可得到初值 0。(X) 原因如题 20。数组 s 中每个元素均可得到初值 0。
23.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 (X)
24.函数中的形式参数是局部变量 (√) 在函数内定义的变量只在本函数范围内在效。
25.函数的定义和函数的调用均可以嵌套 (X) 定义不可以，但调用可以。
26.实参和与其对应的形参共同占用一个存储单元 (X) 函数调用时，实参和与其对应的形参各占独立的存储单元。而形参不被调用时不分配内存单位。
27.用户定义的函数中可以没有 return 语句 (√) 返回值是通过 return 语句给主调函数返回一个数值，这也是主调函数和被调函数之间的单向数据传递方式。函数的 return 语句中可以没有表达式，函数中也可以没有 return 语句，只需要将函数定义 void 类型即可。
28.当调用函数时实参是一个数组名则向函数传送的是数组的首地址 (√) 当调

用函数时，实参是一个数组名，则向函数传送的是数组的首地址，函数中的形参可定义成以下三种形式：①形参定义成数组；②形参定义成可变长数组：③形参定义为指针变量。

29.C 语言中形参的默认存储类别是自动(auto)。 (√)