2009年9月全国计算机等级考试二级笔试试卷

C语言程序设计

（考试时间90分钟，满分100分）

参考答案及解析仅供参考

一、选择题（（1-10）、（21-40）每题2分，（11）-（20）每题1分，共70分）

下列各题A、B、C、D四个选项中，只有一个选项是正确的，请将正确填涂在答案卡相应位置上，答在试卷上不得分。

1、下列数据结构中，属于非线性结构的是

A)循环结构　B）带链队列　C)二叉树　D）带链栈

2、下列数据结构中，能够按照"先进后出"原则存取数据的是

A)循环队列　B)栈　C）队列　D)二叉树

3、对于循环队列，下列叙述中正确的

A）队头指针是固定不变的

B）队头指针一定不大于队尾指针

C）队头指针一定小于队尾指针

D）队头指针可以大于队尾指针，也可以小于队尾指针

4、 算法的空间复杂度是指

A) 算法在执行过程中所需要的计算机存储空间

B）算法所处理的数据量

C) 算法程序中的语句或指令条数

D）算法在执行过程中所需要的临时工作单元数

5、 软件设计中划分模块的一个准则是

A）低内聚低耦合　　B）高内聚低耦合

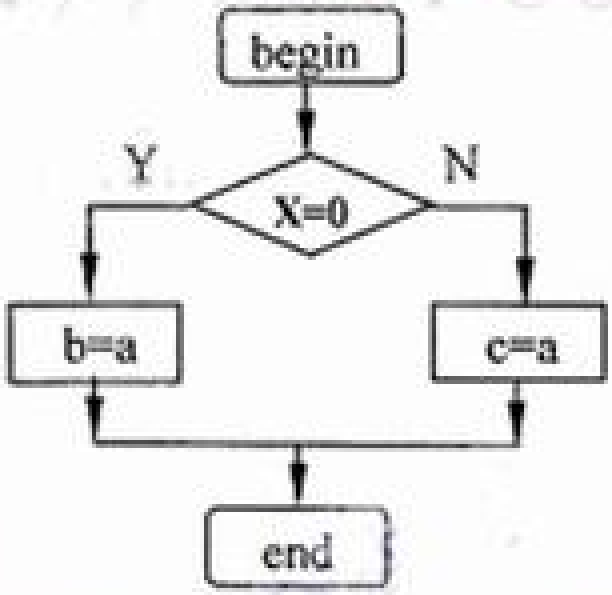
C）低内聚高耦合　　D) 高内聚高耦合

6、下列选项中不属于结构化程序设计原则的是

A）可封装　　B)自顶向下　　C）模块化　　D）逐步求精

7、软件详细设计的图如下，该图是：

A) N-S图　　B) PAD 图　　C) 程序流程图　　D) E-R图



8、数据库管理系统是：

A）操作系统的一部分　　B）在操作系统支持下的系统软件

C）一种编译系统　　D）一种操作系统

9、在E-R图型中，用来表示两个实体联系的图型的是：

A) 椭圆形　　B) 矩形　　C）菱形　　D）三角形

10、有三个关系R、S和T如下，

其中关系T由关系R和关系S通过某种操作得到：该操作为：

A) 选择　　B) 投影　　C）交　　D）并

R

A	B	C
a	1	2
b	2	1
c	3	1

S

A	B	C
d	3	2

T

A	B	C
a	1	2
b	2	1
c	3	1
d	3	2

11、一下叙述中正确的是：

A) 程序设计的任务是编写程序代码并上机调试。

B) 程序设计的任务是确定所用的数据结构。

C）程序设计的任务是确定所用的算法。

D）以上三种说法都不完整。

12、以下选项中，能用作用户标识符的是：

A) void　　B) 8_8　　C）_0_　　D) unsigned

解析：

11、选D。原文见高教版二级教程P2，程序设计的任务包括A、B、C及相关文档。

12、选C。标识符以字母或下划线开头，关键字不能用作标识符。

A、D为关键字，B以数字开头，所以都是错误的。

13、 阅读以下程序

```
#include <stdio.h>
main ( )
{   int case ;   float printF;
    printf("请输入2个数：");
    scanf("%d %f",&case,&printF);
    printf("%d %f\n",case,printF);
}
```

该程序在编译时产生错误，其出错原因是

A)定义语句出错，case是关键字，不能用作用户自定义标识符

B)定义语句出错，printF不能用作用户自定义标识符

C)定义语句无错，scanf不能作为输入函数使用

D)定义语句无错，printf不能输出case的值

解析：

13、选A。case是关键字，关键字不能用作标识符。C语言关键字见教材附录 。

（注：标识符区分大小写，printf不是关键字，可用作标识符，当然printF也可以 ）

14、 表达式： (int) ((double)9/2)-(9)%2的值是

A） 0　　B） 3　　C） 4　　D） 5

15、 若有定义语句：int x=10; , 则表达式 x -= x+x 的值为

A）-20　　B） -10　　C） 0　　D） 10

解析：

14、选B。考点为运算符的优先级。

括号>强制类型转换int >乘除 > 加减

(int)((double)9/2)-(9)%2 = (int)(9.0/2)-(9)%2= (int)(4.5)- 1=3

15、选B。考点为复合的赋值运算符。（注意x+x作为一个整体）

x -= x+x　→　x = x – (x+x)　→ x = – x = -10

16、有以下程序

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

程序运行后的输出结果是

A）0，0　　B）1，0　　C）3，2　　D) 1,2

解析：

16、选D。考点为赋值表达式的使用，赋值表达式的值和变量的值是一样的。

printf在输出赋值表达式的值时，先赋值再输出。

b=a+b=1+0=1　　a=2*b=2

17、 设有定义：int a=1,b=2,c=3;，以下语句中执行效果与其他三个不同的是

A）if(a>b)　c=a,a=b,b=c;

B) if(a>b)　{ c=a,a=b,b=c; }

C）if(a>b)　c=a;a=b;b=c;

D) if(a>b)　{ c=a;a=b;b=c; }

解析：

17、选C。考点为if语句和逗号表达式的使用。

逗号运算符也称为顺序求值运算符，依次执行逗号表达式中的每个表达式。逗号表达式是一个整体，复合语句也是一个整体，当a>b时，A、B、D中 if 后的语句可以作为整体被执行，把a、b的值交换。C的等价语句形式为：

```
if(a>b)　c=a;
a=b;b=c;
```

18、有以下程序

```
#include <sdtio.h>
main ()
{ int c=0,k ;
  for( k=1; k<3; k++ )
     switch ( k )
     {   default: c+=k ;
         case 2: c++; break ;
         case 4: c+=2; break ;
     }
  printf("%d\n",c);
}
```

程序运行后的输出结果是

A) 3　　B) 5　　C) 7　　D) 9

解析：

18、选A。考点为switch语句的使用。

switch语句中case和default的顺序可以任意，不影响程序结果。

switch语句在循环中被执行2次。

k=1时，　c+=k　c=0+1=1

无break，继续执行

c++　c=2

有break，终止switch

k=2时，　c++　c=3

有break，终止switch

循环结束，输出c。

19、与语句：k= a>b? (b>c?1:0) : 0; 功能相同的是

A) if((a>b)&&(b>c)) k=1;
   else k=0;

B) if((a>b)||(b>c)) k=1;
   else k=0;

C) if(a<=b) k=0;
   else if(b<=c) k=1;

D) if(a>b) k=1;
   else if(b>c) k=1;
   else k=0;

解析：

19、选A。考点为条件表达式的使用。

语句的功能为：

a>b且b>c时，k=1

a>b且b<=c时，k=0

a<=b时，k=0

符合语句功能的只有A。

20、 有以下程序

```
#include <stdio.h>
main( )
{ char s[ ]={"012xy"};  int i , n=0;
   for(i=0; s[i]!=0; i++)
       if(s[i]>='a'&&s[i]<='z')   n++;
   printf("%d\n",n);
}
```

程序运行后的输出结果是

A)0　　B）2　　C）3　　D）5

解析：

20、选B。考点为字符数组的使用。C语言用字符数组存放字符串，用 \0 作为结束标志。（\0 为ASCII码值为0的字符，也即数值0）

程序的功能为统计字符数组s中小写字符的个数，n为计数器。

21、有以下程序

```
#include <stdio.h>
main ( )
{  int  n=2 , k=0 ;
   while( k++&&n++>2 ) ;
   printf("%d  %d\n", k , n );
}
```

程序运行后的输出结果是

A）0 2　　B) 1 3　　C) 5 7　　D) 1 2

附加解析：

计算表达式 a&&b的值时，

如果a为真，才计算表达式b，

如果a为假， a&&b必定为假，

不计算表达式b。

解析：

21、选D。考点为while语句的使用。注意：while(k++&&n++>2)；的循环体为空语句，所以程序是输出退出while循环后k、n的值。

k++为先使用k的值再增1。先使用k的值，k=0，逻辑与结果为0，第一次执行while循环时条件就不成立，直接退出循环， k的值增1，n的值没有任何改变。

22、有以下定义语句，编译时会出现编译错误的是

A) char a='a';　　B) char a='\n';

C) char a='aa';　　D) char a='\x2d';

解析：

22、选C。考点为字符型变量的赋值和基本概念。

字符型为单引号括起的一个字符。

A为标准的字符型赋值方法，

B为把一个转义字符赋值给字符变量，也是正确的。

C为单引号括起的两个字符，不符合字符型定义。

D表面看上去是错误的，其实是正确的，也是一个转义字符。

'\x2d'表示ASCII码值为16进制数2d的字符，即'-'号。

23、有以下程序

```
#include <stdio.h>
main()
{  char  c1 , c2 ;
   c1='A'+'8'-'4';
   c2='A'+'8'-'5';
   printf("%c,%d\n"c1,c2);
}
```

已知字母A的ASCII码为65，程序运行输出后的结果是：

A）E,68　　B）D,69　　C）E,D　　D）输出无定值

解析：

23、选A。考点为字符型数据的使用和基本知识。字符型数据在内存中存放的是字符的ASCII码值，可以作为整型数据来处理。英文字符和数字在ASCII码表中是按顺序排列的。　c1='A'+'8'-'4'= 'A'+4 = 'E'　　c2='A'+'8'-'5'= 'A'+3 = 'D'

24、有以下程序

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

程序运行后的输出结果是

A）32　　B)12　　C)21　　D)22

解析：

24、选C。考点为函数参数的传递。C语言中函数参数的传递是值传递，是把实参的值传给形参，是单向传递，形参的改变不会影响到实参的值。

程序中，把实参a的值传给形参p，p=1，然后p=d++，再次赋值后p=2，输出p的值2。返回到主程序中，输出a的值1。（形参p的改变不会影响到实参a的值，a的值仍为1）

25、以下函数findmax拟实现在数组中查找最大值并作为函数值返回，但程序中有错导致不能实现预定功能。

```
#define  MIN  -2147483647
int   findmax( int x[ ] , int n )
{  int  i , max;
   for( i=0; i<n; i++ )
  {   max =MIN;
      if(max<x[i])   max=x[i]; }
   return  max;
}
```

造成错误的原因是

A）定义语句 int i,max；中max 未赋初值

B）赋值语句max=MIN; 中，不应给max 赋MIN值

C）语句 if(max<x[i]) max=x[i]; 中判断条件设置错误

D）赋值语句max=MIN; 放错了位置

解析：25、选D。考点为求最大值的算法。 max=MIN; 不应该放在循环内，而应该放到for循环的前面。先让max取最小的整数，这样第1次循环时max就可以取得第1个数组元素的值，然后在循环中把后面的数组元素依次和max比较，让max取大值。

26、有以下程序

```
#include <stdio.h>
main()
{   int  m=1 , n=2 , *p=&m , *q=&n , *r ;
    r=p; p=q; q=r;
    printf("%d,%d,%d\n", m, n, *p, *q ) ;
}
```

程序运行后的输出结果为

A） 1，2，1，2　　B） 1，2，2，1

C） 2，1，2，1　　D） 2，1，1，2

解析：

26、选B。考点为指针的基本概念。

p、q为指针，初始化时p指向m，q指向n。

执行r=p; p=q; q=r; 后，p和q的值交换，从而p指向n，q指向m。

指针的改变不会影响m、n的值，最后*p和*q的值为n、m的值。

27、若有定义语句：

int a[4][10]，*p，*q[4]；且0≤i<4，则错误的赋值是：

A）p=a　　B）q[i]=a[i]　　C）p=a[i]　　D）p=&a[2][1]

解析：

27、选A。考点为指向二维数组的指针的用法。

p为基类型为int的指针，指向一个整型数据，也就可以指向一个数组元素，所以D正确。

a是二维数组名，存放二维数组的首地址，但二维数组名是一个行指针，其基类型为具有10个元素的一维数组。所以A错误，二者基类型不一致（p+1指向下一个元素，而a+1指向二维数组的下一行）。如果p定义为int (*p)[10]，才可以赋值p=a。

在C语言中，二维数组a[4][10]可以看做是由4个元素组成的一维数组，这4个元素为a[0]、a[1]、a[2]、a[3]，而其中每一个元素又是由10个元素组成的一维数组。

在这里，a[i]也是一个数组名，可以表示一维数组的首地址，但a[i]是一个列指针，基类型为int，指向一维数组的第1个元素。

同时，指针数组q的每个数组元素q[i]的基类型也为int，所以p、a[i]、q[i]的基类型一致，选项B、C是正确的。

28、有以下程序：

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

程序运行后的输出结果是：

A） 9，One*World　　B） 9，One*Dream！

C） 10，One*Dream！　　D） 10，One*World

解析：

28、选C。考点为二维字符数组的使用和strlen()函数的使用。

初始化p=str[1]后， p指向第二个字符串"One*Dream！ "。

strlen()函数计算字符串的长度时不包括结束标志，所以strlen(p)=10。

29、有以下程序：

```
#include <stdio.h>
main()
{   int  a[ ]={ 2,3,5,4 } , i ;
    for( i=0; i<4; i++ )
     switch ( i%2 )
     {   case 0:  switch ( a[i]%2 )
                  { case0:  a[i]++ ;   break;
                    case1:  a[i]-- ;
                  }break;
         case 1:   a[i]=0;
     }
    for(i=0; i<4; i++)     printf("%d",a[i]) ;
    printf("\n");
}
```

程序运行的输出结果是：

A）3 3 4 4　　B）2 0 5 0

C）3 0 4 0　　D）0 3 0 4

解析：

29、选C。考点为switch语句。

外部switch语句在循环中被执行4次。

i=0时，执行case 0,内部switch语句也执行case 0,a[i]++, a[0]=3

i=1时，执行case 1， a[1]=0

排除法，只有C正确。

i=2时，执行case 0,内部switch语句执行case 1,a[i]--, a[2]=4

i=3时，执行case 1， a[3]=0

最后依次输出为：3 0 4 0

30、有以下程序：

```
#include <stdio.h>
#include <string.h>
main()
{   char a[10]=”abcd”;
    printf(“%d, %d\n”,strlen(a),sizeof(a));
}
```

程序运行后的输出结果为：

A）7，4　　B）4，10　　C）8，8　　D）10，10

解析：

30、选B。考点为strlen()函数和sizeof()运算符的使用。

strlen()函数计算字符串的长度时，遇到结束标志为止，且长度不包括结束标志，所以strlen(a)=4 ，排除法选B。

sizeof()运算符的操作数可以是类型名或变量名、数组名等，当操作数是数组名时，其结果是数组的总字节数，所以sizeof(a) =10。

31、下面是有关C语言字符数组的描述，其中错误的是

A）不可以用赋值语句给字符数组名赋字符串

B）可以用输入语句把字符串整体输入给字符数组

C）字符数组中的内容不一定是字符串

D）字符数组只能存放字符串

解析：

31、选D。考点为字符数组的使用。

字符数组名是数组首地址，是常量，不能被重新赋值，所以A正确。

可以用scanf("%s" , str) 对字符串整体输入，str可以是字符数组名或者字符指针，所以B正确。

C和D说法对立，必定有一个正确，用排除法A、B选项根本不用看。

字符数组的所有元素可以只存放普通字符，不存放结束标志。

所以D错误。

32、下面函数的功能是

```
fun( char  *a , char *b )
{   while( (*b=*a)!= '\0' )    {  a++;  b++;  }   }
```

A）将a所指字符串赋给b所指空间

B）使指针b指向a所指字符串

C）将a所指字符串和b所指字符串进行比较

D）检查a和b所指字符串中是否有'\0'

解析：

32、选A。考点为指针的概念及while循环。

While循环条件为：(*b=*a)!= '\0'，执行时先把指针a所指向的字符赋给指针b所在内存单元，如果该字符不是结束标志"\0"，执行循环体a++; b++;，指针a、b分别指向下一个字符单元。

再判断循环条件，如果成立，继续把指针a所指向的字符赋给指针b所在内存单元，直到遇到结束标志为止。所以正确答案为A。

33、设有以下函数：

void fun(int n, char *s) {……}

则下面对函数指针的定义和赋值均正确的是

A）void (*pf)( ); pf=fun;

B）void *pf( ); pf=fun;

C）void *pf( ); *pf=fun;

D）void (*pf)（int，char）； pf=&fun;

解析：

33、选A。考点为指向函数的指针的用法。

函数名代表函数的入口地址。

指向函数的指针应该定义为void (*pf)( )。

如果定义为void *pf( )，则表示函数pf 返回值为一个基类型为void的指针。

综上，所以正确答案为A。

34、有以下程序：

```
#include <stdio.h>
int  f(int n);
main( )
{  int  a=3 , s ;
   s=f(a);  s=s+f(a);
   printf("%d\n",s);
}
int  f(int n)
{  static  int  a=1;
    n+=a++;
   return  n;
}
```

程序运行后的输出结果是

A) 7　　B) 8　　C) 9　　D) 10

解析：34、选C。考点为静态局部变量的使用。

主函数和f函数中的a都为局部变量，作用域都在本函数之内，互不影响。

f 函数中的a为静态局部变量，占用固定的内存单元，下一次调用时仍可保留上次调用时的值。

也就是说，如果多次调用f 函数，a的定义只在第一次调用时有效，从第二次调用开始，a的定义相当于不存在，直接使用a的值。

主函数中调用了2次f(a)。

第一次调用，s=f(a)=f(3)

f函数：n=3　a=1　　n=n+(a++)=4　　a=2

返回n，主函数s=4

第二次调用，s=s+f(a)=4+f(3)　(a值为主函数中的a值)

f函数：n=3　a=2　　n=n+(a++)=5　　a=3

返回n，主函数s= 4+f(3)=4+5=9

最后输出s的值为9。

35、有以下程序

```
#include <stdio.h>
#define f(x) x*x*x
main()
{
   int a=3,s,t;
   s=f(a+1); t=f((a+1));
   printf("%d,%d\n", s, t );
}
```

程序运行后的输出结果是

A）10，64　　B）10，10　　C) 64，10　　D) 64，64

解析：35、选A。考点为带参数的宏定义。

宏定义中的参数没有类型，仅为字符序列，不能当作表达式运算。

宏展开时，把实参字符串原样写在替换文本中。

s=f(a+1)=a+1*a+1*a+1=10

t=f((a+1))=(a+1)*(a+1)*(a+1)=64

36、下面结构体的定义语句中，错误的是

A）struct ord { int x; int y; int z;}; struct ord a;

B）struct ord { int x; int y; int z;} struct ord a;

C）struct ord { int x; int y; int z;} a;

D）struct { int x; int y; int z;} a;

解析：

36、选B。考点为结构体变量的定义。

可以先定义结构体类型，再定义结构体变量，如A。

可以在定义结构体类型的同时定义结构体变量，如C。

可以直接定义结构体变量，没有类型名，如D。

B选项错误，定义结构体类型的同时使用此类型。

37、 设有定义：char *c;，以下选项中能够使字符型指针c正确指向一个字符串的是

A）char str[]="string "; c=str;　　B）scanf("%s",c);

C) c=getchar();　　D）c="string";

解析：

37、选A。考点为字符指针的使用。

选项A为正确用法。先将字符串存于字符数组中，然后将数组名赋给字符指针。

（ 数组名代表数组首地址，定义数组时为其分配确定地址）

选项C错误。 getchar( )函数输入一个字符给字符型变量，而不是字符指针。

选项B和D有类似的错误，两个选项并无语法错误，但运行时可能会出现问题。

原因：

在选项B和D中，字符指针没有被赋值，是一个不确定的值，指向一个不确定的内存区域，这个区域可能存放有用的指令或数据。在这个不确定的区域重新存放字符串，可能会发生无法预知的错误。

38、有以下程序

```
#include <stdio.h>
#include <string.h>
struct A
{   int a;   char b[10];   double c; };
struct A f (struct A t );
main()
{   struct A a={1001,"ZhangDa",1098.0};
    a=f(a);   printf("%d,%s,%6.1f\n",a.a, a.b, a.c );
}
struct A f( struct A t )
{ t.a=1002;   strcpy(t.b,"ChangRong");  t.c=1202.0;  return t; }
```

程序运行后的输出结果是

A）1001，ZhangDa,1098.0　　B）1002, ZhangDa,1202.0

C）1001，ChangRong,1098.0　D）1002, ChangRong,1202.0

解析：38、选D。考点为结构体类型在函数中的应用。

因为使用结构体变量，程序看似很杂乱。但在使用中，结构体变量和普通变量的作用是一样的。如果这样看，程序执行流程如下：函数调用f(a)将实参a传给形参t，函数内部对t 重新赋值，然后返回t；主函数a=f(a)，把返回值t 赋给a，然后输出a，其实就是t 的值。

39、若有以下程序段：

```
int r=8;
printf("%d\n", r>>1);
```

输出的结果是：

A) 16　　B) 8　　C) 4　　D) 2

40、下列关于C语言文件的叙述中正确的是：（　）

A) 文件由一系列数据依次排列组成，只能构成二进制文件

B) 文件由结构序列组成，可以构成二进制文件或文本文件

C) 文件由数据序列组成，可以构成二进制文件或文本文件

D) 文件由字符序列组成，其类型只能是文本文件

解析：

39、选C。考点为位运算中右移运算符的使用。

每右移一次，相当于除以2。本题中，$r = (8)_{10} = (0000\ 1000)_2$

r>>1后，$r = (0000\ 0100)_2 = (4)_{10}$

40、选C。C语言中根据数据的组织形式，分为二进制文件和ascii码文本文件。

一个C文件是一个字节序列或者二进制序列，而不是一个记录（结构）序列。

二、填空题(每空2分，共30分)

请将每空正确答案写在答题卡【1】至【15】序号的横线上，答在试卷上不得分。

1、某二叉树有5个度为2结点以及3个度为1结点，则该二叉树中共有【1】个结点。

2、程序流程图中的菱形框表示的是【2】。

3、软件开发过程主要分为需求分析，设计，编码与测试四个阶段，
其中【3】阶段产生"软件需求规格工作书"。

4、在数据库技术中，实体集之间的联系可以是一对一或一对多或多对多的，
那么"学生"和"可选课程"的联系为【4】。

5、人员基本信息一般包括：身份证号，姓名，性别，年龄等。
其中可以作为关键字的是【5】

1、答案：14　解析：度为2结点有5个，则度为0的节点（叶子节点）为5+1=6个，度为1的节点有3个，总节点数为5+6+3=14个。（也可以画示意图数节点个数）

2、答案：逻辑条件

3、答案：需求分析

4、答案：多对多

解析：一个学生可以选多门课程，一门课程可以被多个学生选。

5、答案：身份证号　解析：关键字的值不能重复，只能是唯一的。

6、若有定义语句：int a=5; 则表达式：a++的值是__【6】__。

7、若有定义语句：double x=17；int y；， 当执行y =(int) (x/5)%2；之后用的值为__【7】__。

8、以下程序运行后的输出结果是__【8】__。

```
#include <stdio.h>
main( )
{   int  x=20;
    Printf("%d  " , 0<x<20 ) ;
    Printf("%d\n" , 0 <x&&x<20 ) ;  }
```

6、答案：5　解析：a++先使用a的值再增1。

7、答案：1　解析：y =(int)(3.　)%2 = 3%2 =1。

8、答案：1 0

解析：

" <" 的结合性为从左到右。 计算0<x<20， x=20，0<x为1，然后1<20结果为1。

关系运算符优先级大于逻辑运算符。0 <x&&x<20相当于（0 <x）&&（x<20），结果为0。

9、以下程序运行后的输出结果是 【9】

```
#include <stdio.h>
main( )
{   int  a=1; b=7;
    do {
            b=b/2;  a+=b;
        }while(b>1);
    printf("%d\n",a);
}
```

9、答案：5

解析：考点为do while语句的用法。

程序执行流程：

a=1 b=7

执行循环体，b=b/2=3 a=4；

b>1条件成立，再执行循环体，b=b/2=1 a=5；

此时， b>1条件不成立，输出a的值5 。

(10) 以下程序

```
#include <stdio.h>
main()
{   int f , f1 , f2 , i ;
    f1=0 ; f2=1 ;
    printf(" %d %d " , f1 , f2 ) ;
    for( i=3; i<=5; i++ )
    {   f = f1+ f2 ;   printf("%d",f) ;
        f1=f2 ; f2=f ;
    }
    printf("\n");
}
```

程序运行后的输出结果是__【10】__

10、答案：0 1 123

解析：

程序执行流程：

f1=0 f2=1 输出：0 1

For循环变量中的 i 只起到控制循环次数的作用，循环3次。

i=3 f =f1+f2=0+1=1 输出1；

f1=f2=1 f2=f=1

i=4 f =f1+f2=1+1=2 输出2；

f1=f2=1 f2=f=2

i=5 f =f1+f2=1+2=3 输出3；

f1=f2=2 f2=f=3

注意循环内输出格式控制符没有空格，所以123连续输出。

11、有以下程序

```
#include <stdio.h>
int a=5;
void fun(int b)
{
    int a=10;
    a + = b;
    printf("%d",a);
}
main()
{   int c=20;
    fun(c);    a + = c ;
    printf("%d\n",a);
}
```

程序运行后的输出结果【11】

11、答案：3025

解析：考点为全局变量和局部变量的用法。

全局变量的作用域为定义开始到文件结束。

局部变量的作用域为定义其函数的内部。

当局部变量与全局变量同名时，在局部变量作用域内，全局变量不起作用。

程序执行流程：

调用函数fun(c)，实参c传给形参b，b=20，局部变量a起作用，a=10，

a+=b a=a+b=10+20=30，输出a值30。

返回到主函数：

a+=c 此时a为全局变量。

a=a+c=5+20=25，输出a值25。

注意格式控制符无空格，所以连续输出。

12、设有定义：

```
struct person
{  int ID ;    char name[12]; } p;
```

请将scanf（“%d”，__【12】__）；语句补充完整，使其能够为结构体变量p的成员ID正确读入数据。

12、答案：&p.ID

解析：考点为结构体成员的引用方法。

结构体变量成员的引用方法：　　结构体变量．成员名

13、有以下程序

```
#include <stdio.h>
main()
{  char a[20]= " how are you? " , b[20] ;
   scanf (“&s”,b);      printf(“%s %s\n”,a,b);
}
```

程序运行时从键盘输入：How are you？<回车>

则输出结果为：___【13】___。

13、答案：How are you？ How

解析：考点为字符串的输入输出。

C语言中用字符数组存放字符串，此题中字符数组a的赋值和输出属于正常用法。关键是字符数组b的输入，C语言把空格作为字符串输入的分隔符，所以字符数组b只能接收到How。（详细参考谭浩强教材P145）

14、有以下程序：

```
#include <stdio.h>
typedef  struct
{ int  num;  double s; } REC;
void   fun1(REC x)   { x.num=23;  x.s=88.5; }
main()
{  REC   a=（16，90.0）；
   fun1（a）；
   printf（%d\n，a.num）；
}
```

程序运行后输出的结果是 【14】

14、答案：16

解析：考点为函数的参数传递。C语言中函数参数传递是值传递，是把实参的值传给形参，是单向传递，形参的改变不会影响到实参的值。

此程序特殊的地方是实参和形参都是结构体变量，用法和普通变量是一样的。

15、有以下程序

```
#include <stdio.h>
fun(int x)
{   if(x/2>0)  fun(x/2);
    printf("%d ",x);
}
main()
{   fun(6);   printf("\n"); }
```

程序运行后的输出结果是【15】

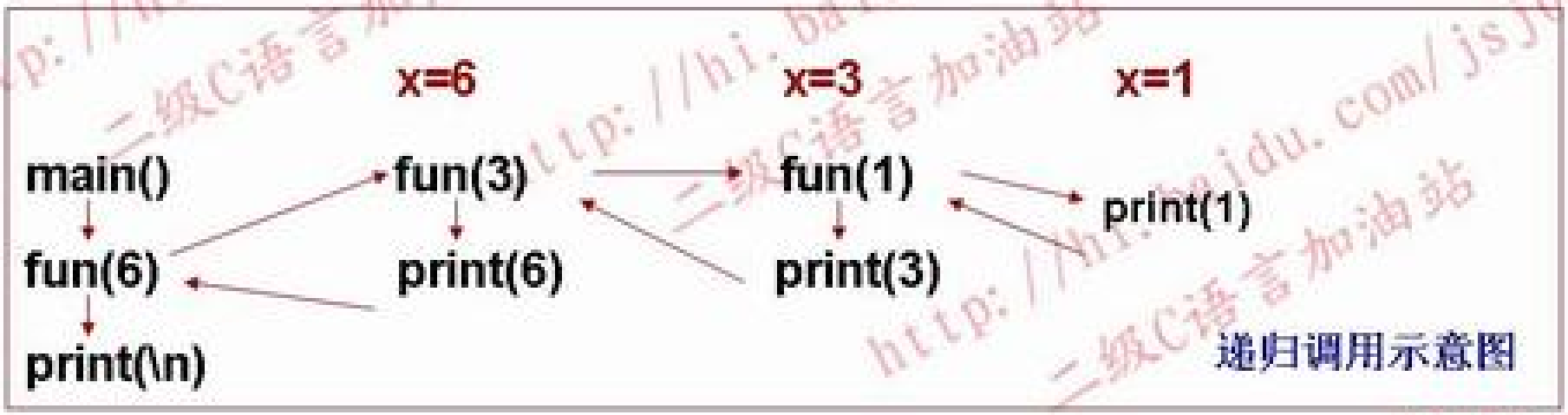
15、答案：1 3 6

解析：

考点为函数的递归调用。程序的执行过程是先递推、后递归的过程。

当x=1时，条件不成立，递推终止。

程序执行流程参照示意图。



递归调用示意图

参考答案

选择题：

1-5：	CBDAB	6-10：	ACBCD
11-15：	DCABB	16-20：	DCAAB
21-25：	DCACD	26-30：	BACCB
31-35：	DADCA	36-40：	BADCC

填空题：

1、14　　2、逻辑条件　　3、需求分析

4、多对多　　5、身份证号　　6、5

7、1　　8、1 0　　9、5

10、0 1 123　　11、3025　　12、&p.ID

13、How are you? How　　14、16

15、1 3 6