

# C 程序设计语言练习题

## 第一部分

1. 以下说法正确的是\_\_\_\_\_  
A) C 程序运行时，总是从第一个定义的函数开始执行  
**B) C 程序运行时，总是从 main() 开始执行**  
C) C 源程序中的 main() 必须放在程序的开始部分  
D) 一个 C 函数中只允许一对花括号
2. 下列用户标识符中正确的一组是\_\_\_\_\_  
A) Name, char, a\_bc, Int **B) abc\_c, Int, \_USA, print**  
C) read, Const, type, long D) int, integer, Double, short\_int
3. 下列标识符中错误的一组是\_\_\_\_\_  
**A) “a”, ‘a’, +a, -a** B) abc\_c, x5y, \_USA, printf  
C) read, Const, type, define D) include, integer, Double, short\_int
4. 下列 4 组整型常量中，错误的一组是\_\_\_\_\_  
**A) xcdf 017 0xe 123** B) 160 0xffff 011 0L  
C) 01 32768u 0617 0x163 D) 0x48a 0205 0x0 -256
5. 下面属于 C 语句的是\_\_\_\_\_  
A) printf(“%d\n”, a) B) /\*This is a statement\*/  
**C) x=x+1;** D) #include<stdio.h>
6. 设 char 型占 1 字节，则 unsigned char 所能表示的数据范围是\_\_\_\_\_  
**A) 0~255** B) -128~127 C) 1~256 D) -128~128
7. 下列属于 C 语言合法的字符常数是\_\_\_\_\_  
A) ‘97’ B) “A” **C) ‘\t’** D) “\0”
8. 已知字母 B 的 ASCII 码为十进制的 66，下面的程序输出是\_\_\_\_\_  
main()  
{char ch1, ch2;  
ch1= ‘B’ + ‘4’ - ‘3’ ;  
ch2= ‘B’ + ‘5’ - ‘3’ ;  
printf(“%d, %c\n”, ch1, ch2);  
}  
**A) 67, D** B) B, C C) C, D D) 不确定
9. 设 x 是 int 型变量，y 是 float 型变量，用下面的语句给这两个变量输入值：scanf(“i=%d, f=%f”, &x, &y); 为了将 10 和 76.25 分别赋给 x 和 y，则正确的输入是(<CR>为回车)  
**A) 10 76.25<CR>** **B) i=10, f=76.25<CR>**  
**C) 10<CR>76.25<CR>** **D) x=10, y=76.25<CR>**
10. 若变量 a, b, c 已正确说明为 float 型，要使语句 scanf(“%f %f %f”,

&a, &b, &c); 将 11.0、22.0 和 33.0 依次赋给变量 a, b, c, 下列不正确的是\_\_\_\_\_

A) 11<回车> B) 11.0, 22.0, 33.0<回车>

22<回车>

33<回车>

C) 11.0<回车>

A) 11 22<回车>

22.0 33.0<回车>

33<回车>

11. 设 `int a=7, b=8;` 则 `printf(“%d,%d,(a+b), (b, a+b)");` 的输出为

A) 7, 15

B) 8, 15

C) 15, 7

D) 出错

12. 若已定义 x 和 y 为 double 型, 则表达式 `x=1, y=x+3/2` 的值为

A) 1

B) 2

C) 2.0

D) 2.5

13. C 语言中要求运算量必须是整型的运算符是

A) +

B) /

C) %

D) -

14. 若有以下定义和语句

```
int u=010, v=0x10, w=10;
```

```
printf(“%d,%d,%d\n”, u, v, w);
```

则输出结果为\_\_\_\_\_

A) 8, 16, 10

B) 10, 10, 10

C) 8, 8, 10

D) 8, 10, 10

15. 若变量已正确定义且 k 的的值是 4, 执行表达式 `j=k--` 后, j, k 的值是\_\_\_\_\_

A) j=4, k=4

B) j=4, k=3

C) j=3, k=4

D) j=3, k=3

16. 下列语句中符合 C 语言语法的语句是

A) `a=7+b+c=a+7;`

B) `a=7+b++=a+7;`

C) `a=7+b, b++, a+7;`

D) `a=7=b, c=a+7;`

17. 设变量 t 为 int 型, 下列选项中, 不正确的赋值语句是\_\_\_\_\_

A) `++t;`

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

C) `k=i==1;`

D) `a=b+c=1;`

18. 设 a 为 int 型变量, 则执行下列语句后, a 的值是\_\_\_\_\_

```
a=10; a+=a-=a-a;
```

A) 10

B) 20

C) 40

D) 30

19. 变量已正确定义, 要将 a 和 b 中的数据交换, 则下列不正确的是\_\_\_\_\_

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;`

20. 为表达关系  $x \geq y \geq z$ , 应使用的 C 语言表达式为\_\_\_\_\_

A) `(x>=y)&&(y>=z)` B) `(x>=y)AND(y<=z)` C) `x>=y>=z` D) `(x>=y)&(y>=z)`

21. 为表示“a 和 b 都大于 0”, 应使用的 C 语言表达式是\_\_\_\_\_

A) `(a>0)&&(b>0)`

B) `a&& b`

C) `(a>0)|| (b>0)`

D) `(a>0)|(b>0)`

22. 结构化程序设计使用的基本程序控制结构为\_\_\_\_\_.

A) 模块结构、选择结构和递归结构

B) 条件结构、顺序结构和过程结构

C) 顺序结构、选择结构和循环结构

D) 转移结构、嵌套结构和递归结构

23. 若所有变量均已正确定义, 下面的程序段运行后 x 的值是\_\_\_\_\_.

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

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

```

else if (b);
if (c) x=3;
else x=4;

```

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

24. 下列与  $y = (x > 0 ? 1 : x < 0 ? -1 : 0)$  ; 功能相同的 if 语句是\_\_\_\_\_.

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

25. 以下程序的输出结果是\_\_\_\_\_.

```

main()
{int a=0,i;
  for (i=1; i<5; i++)
  {switch (i)
   {case 0:
    case 3:  a+=2;
    case 1:
    case 2:  a+=5;
    default: a+=5;
   }
  }
  printf( "%d\n", a);
}

```

- A) 37**      B) 13      C) 10      D) 20

26. 写出下面程序段的输出结果。 **-1**

```

main()
{ int x=2;
  while(x--);
  printf( "%d\n", x);
}

```

27. 下面程序的输出结果是\_\_\_\_\_。

```

main()
{int i;
  for (i=0;i<10;i++);
}

```

```
printf ( "%d,i);
}
```

- A) 0                      B) 123456789                      C) 023456789                      **D) 10**

28. 下面程序段的运行结果为\_\_\_\_\_

```
for (i=1; ++i<4;);
printf( "%d\n", i);
```

- A) 3                      **B) 4**                      C) 5                      D) 不定

29. 在执行以下程序时，如果从键盘上输入 ABCdef 后回车，则输出\_\_\_\_\_

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

- A) ABCdef                      **B) abcDEF**                      C) abc                      D) DEF

30. 下列程序段中，不是死循环的是\_\_\_\_\_

- A) `int i=100;`  
`while(1)`  
`{ i=i%100+1;`  
`if (i>100) break;`  
`}`
- B) `int k=0;`**  
`do`  
`{++k;`  
`} while (k>=0);`
- C) **`for (;;) ;`**
- D) `int s=3379;`  
`while (s++%2+s%2)`  
`s++;`

31. 以下程序的输出结果是\_\_\_\_\_

```
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**

32. 执行以下程序时, 为了使输出结果为  $t=4$ 。则给  $a$  和  $b$  输入值应满足的条件是\_\_\_\_\_

```
main()
{ int s, t, a, b;
  scanf( "%d%d", &a, &b);
  s=1, t=1;
  if (a>0) s=s+1;
  if (a>b) t=s+t;
  else if (a==b) t=5;
        else t=2*s;
  printf( "t=%d\n", t);
}
```

- A)  $a>b$       B)  $a<b<0$       **C)  $0<a<b$**       D)  $0>a>b$

33. 下面程序的输出结果\_\_\_\_\_

```
main ()
{int x=100, a=10, b=20, ok1=5, ok2=0;
 if (a<b)
    if (b!=15)
        if (!ok1=5)
            x=1;
    else
        if (ok2) x=10;
 x=-1;
 printf( "%d\n", x);
}
```

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

34. 定义  $\text{int } a=10$ , 下列循环的输出结果是\_\_\_\_\_

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

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

35. 写出下面程序段的输出结果。  **$1+3+5+7=16$**

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

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

```
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( "\nmax=%f\namin=%f\n" ,amax,amin);
}
```

37. 以下程序运行后，如果从键盘输入 1298，则输出结果为\_\_\_\_\_8921\_\_\_\_\_

```
main()
{int n1,n2;
 scanf( "%d" ,&n2);
 while(n2!=0)
 {n1=n2%10;
  n2=n2/10;
  printf( "%d" , n1);
 }
}
```

38. 写出以下程序段的输出结果

(1) 200

```
int x=100,y=200;
printf( "%d" , (x>y)>x:y);
```

(2) x<=y

```
unsigned x=1;
int y=-1;
if(x>y)
    printf( "x>y" );
else
    printf( "x<=y" );
```

(3) TRUE

```
int x=10;
if(x=20) printf( "TRUE" );
else printf( "FALSE" );
```

39. 写出下面程序段的输出结果。 10

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

40. 写出下面程序段的输出结果。 01112234

```
main()
{ int i;
  for(i=0; i<5; i++)
  switch(i)
  { case 1: printf("%d", i);
    case 2: printf("%d", i);
    default: printf("%d", i);
  }
}
```

41. 表达式  $x=(y=6, y+6, y++)$ ,  $(x=6)+8$  的值是多少? 14

42. 字符串 “`\\n\\101\\0345\\'`” 占几个字节? 7

43. 假设  $i=0, j=0, k=0$ , 经过运算  $i++\&\&j++\&\&k++$  后各变量的值分别是多少?

1 0 0

## 第二部分

44. 数组定义为 `int a[3][2]={1, 2, 3, 4, 5, 6}`, 值为 6 的数组元素是\_\_\_\_\_

A) `a[3][2]` B) `a[2][1]` C) `a[1][2]` D) `a[2][3]`

45. 设有数组定义 `char array[]="China"`; 则数组 `array` 所占的空间为\_\_\_\_\_

A) 4 个字节 B) 5 个字节 C) 6 个字节 D) 7 个字节

46. 以下程序段的输出结果是\_\_\_\_\_。

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

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

47. 以下能对二维数组 `x` 正确进行初始化的语句是\_\_\_\_\_。

A) `int x[2][ ]={{1, 0, 1}, {5, 2, 3}};`

B) `int x[ ][3]={{1, 2, 3}, {4, 5, 6}};`

C) `int x[2][4]={{1, 2, 3}, {4, 5}, {6}};`

D) `int x[ ][ ][ ]={{1, 0, 1}, {}, {1, 1}};`

48. 以下程序的输出结果是\_\_\_\_\_。

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

```
main()
{char w[][10]={ "ABCD" , "EFGH" , "IJKL" , "MNOP" }, k;
  for (k=1;k<3;k++)
    printf( "%s\n" ,&w[k][k]);
}
```

- A) ABCD                      B) ABCD                      C) EFG                      **D) FGH**  
     FGH                      EFG                      JK                      KL  
     KL                      IJ                      0  
                                 M

49. 下面的程序运行后，输出的结果是\_\_\_\_\_

(1) **3+4+5+6+0**

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

(2) **4 27 16**

```
main()
{ int i, j;
  int a[6]={ "12, 4, 17, 24, 27, 16"}, b[6]={27, 13, 4, 25, 23, 16};
  for(i=0; i<6; i++)
  { for(j=0; j<6; j++)
    { if(a[i]==b[j]) break;
      if(j<6) printf( "%d" , a[i]);
    }
    printf( "\n" );
  }
}
```

(3) **8 20**

```
#include<string.h>
main()
{char st[20]= "Good bye\0\t\ ' \\" ;
  printf( "%d %d\n, strlen(st), sizeof(st));
}
```

(4) 输入 ABC 输出 **ABC6789**

```
#include <stdio.h>
#include <string.h>
main( )
```



```
{ char ss[10]= "12345" ;
  gets(ss); strcat (ss, " 6789" ); printf( "%s\n" ,ss);
}
```

50. 下面程序的功能是将一个字符串按逆序存放，请填空：

```
#include<string.h>
main()
{char str[80],m;
  int i,j;
  gets(str);
  for (i=0,j=strlen(str);i<_j-1____;i++,j--)
    {m=str[i];
      str[i]=_str[j-1]____; str[j-1]=m;
    }
  printf( "%s\n" ,str);
}
```

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

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

52. 下面的程序用选择排序的方法将从键盘输入的 n 个整数按升序排列并输出排序的结果。请填空：

```
#define MAX 100
main()
{int i, j ,index,n,a[MAX],t;
  scanf( "%d" ,&n);
  for (i=0;i<n;i++)
    scanf ( "%d" ,&a[i])
  for (i=0;i<__n-1____;i++)
```

```

    { index=i;
      for(j=i+1;j<n;j++)
        if( a[j] < a[index])
          index= j ;
      if (i!=index)
        {t=a[i];a[i]=a[index];a[index]=t;}
    }
  for (i=0;i<n;i++)
    printf ( "%4d" ,a[i]);
}

```

53. 下面程序的输出是\_\_\_\_\_.

```

main ()
{ char s[] = "12134211" ;
  int v1=0, v2=0, v3=0, v4=0, k;
  for (k=0;s[k];k++)
    switch(s[k])
    { default:  v4++;
      case '1': v1++;
      case '3': v3++;
      case '2': v2++;
    }
  printf ( "v1=%d, v2=%d, v3=%d, v4=%d\n" , v1, v2, v3, v4);
}

```

- A) v1=4, v2=2, v3=1, v4=1                      B) v1=4, v2=9, v3=3, v4=1  
 C) v1=5, v2=8, v3=6, v4=1                      D) v1=8, v2=8, v3=8, v4=8

54. 若函数为 int 型, 变量 z 为 float 型, 则该函数体内的语句 return(z); 返回的值是\_\_\_\_\_。

- A)** int 型                      B) float 型                      C) 不定                      D) 不确定

55. 在 C 语言中, 函数的隐含存储类型是\_\_\_\_\_。

- A) auto                      B) static                      C) extern                      D) 无存储类型

56. 在 C 语言中, 形参的缺省存储类型是\_\_\_\_\_。

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

57. 有如下函数调用语句:

```
func(rec1, rec2+rec3, (rec4, rec5));
```

该函数调用语句中, 含有的实参个数是\_\_\_\_\_。

- A)** 3                      B) 4                      C) 5                      D) 有语法错误

58. 若调用函数的实参是一个数组名, 则向被调用函数传送的是\_\_\_\_\_。

- A) 数组的长度                      **B)** 数组的首地址  
 C) 数组每个元素的地址                      D) 数组每个元素中的值

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

A) 地址传递

**B) 单项值传递**

C) 由实参传递给形参, 再由形参传递给实参

D) 传递方式由用户指定

60. 在以下有关 C 语言函数的描述中, 正确的是\_\_\_\_\_。

**A) 在 C 语言中, 调用函数时, 只能把实参的值传给形参, 形参的值不能传给实参**

B) C 函数即可以嵌套定义又可以递归调用

C) 函数必须有返回值, 否则不能使用函数

D) C 程序中有调用关系的所有函数, 必须放在同一个源程序文件中

61. 以下程序输出结果是\_\_\_\_\_。

(1) 31

```
fun(int x, int y, int z)
{z=x*x+y*y; }
```

```
main()
{int a=31;
 fun(5,2,a);
 printf(“%d”,a);
}
```

(2) 13+8

```
func(int a, int b)
{ int c;
  c=a+b;
  return c;
}
```

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

(3) 2+1+8+9

```
void reverse(int a[],int n)
{int i,t;
 for(i=0;i<n/2;i++)
 {t=a[i];a[i]=a[n-1-i];a[n-1-i]=t;}
}
```

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

(4) 3,2 2,3

```
void fun(int x,int y)
{x=x+y; y=x-y; x=x-y;}
```

```
printf(“%d,%d”,x,y);
}
```

<pre> main() {int x=2,y=3; (5) 2 long fib(int x) {  switch(x)     { case 0:return 0;       case 1:case 2:return 1;     } return(fib(x-1)+fib(x-2)); } (6) 1+2+3+4+5 int f( ) { static int i=0;   int s=1;   s+=i; i++;   return s; } (7) 7+8+9 f(int a) {int b=0;   static int c=3;   b++; c++;   return (a+b+c); } (8) 4+3+2 long fun5(int n) { long s;   if (n==1  n==2) s=2;   else s=n+fun5(n-1);   return s; } (9) 5 6 int f(int x,int y,int cp,int dp) { cp=x*x+y*y;   dp=x*x-y*y; } (10) 5 </pre>	<pre> fun(x,y); printf("%d,%d\n",x,y); }  main() { long k;   k=fib(3);   printf( "k=%d\n ",k); }  main() { int i,a=0;   for(i=0;i&lt;5;i++) a+=f();   printf("%d\n",a); }  main() {int a=2, i;   for (i=0;i&lt;3;i++)     printf( "%d\n" ,f(a)); }  main() {long x;   x=fun5(4);   printf( "%ld\n" ,x); }  main() { int a=4,b=3,c=5,d=6;   f(a,b,c,d);   printf( "%d %d\n" ,c,d); } </pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

```
#include <string.h>
void f(char p[][10], int n)
{ char t[20]; int i, j;
  for(i=0; i<n-1; i++)
    for (j=i+1; j<n; j++)
      if(strcmp(p[i], p[j])<0)
        { strcpy(t, p[i]);
          strcpy(p[i], p[j]);  strcpy(p[j], t);
        }
}
main()
{char p[][10]={"abc", "aabdfg", "abbd", "dcdbe", "cd"}; int i;
  f(p, 5); printf("%d\n", strlen(p[0]));
}
```

62. 以下叙述中正确的是哪个？

- 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

### 第三部分

63. 已知: int x; 则下面说明指针变量 pb 的语句\_\_\_\_\_是正确的。

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

64. 有如下程序段:

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

执行该程序段后, a 值是\_\_\_\_\_。

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

65. 若有说明: int i, j=2, \*p=&i;, 则能完成 i=j 赋值功能的语句是\_\_\_\_\_。

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

66. 已知: int a, \*p=&a; 则为了得到变量 a 的值, 下列错误的表达式为\_\_\_\_\_。

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

67. 已知 static int a[2][3]={2, 4, 6, 8, 10, 12}; 正确表示数组元素地址的是\_\_\_\_\_。

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

68. 已知 int a[3][4], \*p=**\*a**; p+=6; 那么\*p 和\_\_\_\_\_的值相同。

- A) \*(a+6)      B) \*(&a[0]+6)      **C) \*(a[1]+2)**      **D) \*(&a[0][0]+6)**

69. 已知说明 int (\*p)[M]; 其中 p 是\_\_\_\_\_。

- 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

70. 若有定义和语句: `int c[4][5], (*cp)[5]; cp=c;` 则对 c 数组元素的引用正确的是哪个?

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

71. 设有以下定义:

```
int a[4][3]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
```

```
int (*prt)[3]=a, *p=a[0];
```

则下列能够正确表示数组元素 `a[1][2]` 的表达式是\_\_\_\_\_。

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

72. 若有以下定义和语句:

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

则对 w 数组元素的非法引用是\_\_\_\_\_。

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

73. 若有以下说明和定义:

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

```
void main()
```

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

```
:
```

```
}
```

在必要的赋值之后，对 `fun()` 函数的正确调用语句是\_\_\_\_\_。

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

74. 已知 `double *p[6];` 它的含义是\_\_\_\_\_。

A) p 是指向 double 类型变量的指针

B) p 是 double 类型数组

C) p 是 double 类型指针数组

D) p 是 double 类型数组指针

75. 设有定义: `int n=0, *p=&n, **q=&p;` 则\_\_\_\_\_是正确的赋值语句。

A) `p=1;`    B) `*q=2;`    C) `q=p;`    D) `*p=5;`

76. 以下选项中，错误的赋值是\_\_\_\_\_

A) `char s1[10]; s1=" Ctest";`

B) `char s2[]={ 'C' , 't' , 'e' , 's' , 't' };`

C) `char s3[20]=" Ctest";`

D) `char *s[4]=" Ctest\n";`

77. 若有定义和语句:

```
int **pp, *p, a=10, b=20;
```

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

则输出的结果是\_\_\_\_\_。

A) 10, 20    B) 10, 10    C) 20, 10    D) 20, 20

78. 以下程序编译连接后生成的可执行文件是 `ex1.exe`，若运行时输入带参数

的命令是: `ex1 abcd efg 10<回车>` , 则运行的结果是什么? 9

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

79. 说明语句 `int (*p)();` 的含义是\_\_\_\_\_。

- A) p 是一个指向一维数组的指针变量
- B) p 是一个指针变量, 指向一个整型数据
- C) p 是一个指向函数的指针, 该函数的返回值是一个整型数据
- D) 以上都不对

80. 说明语句 `int *(*p)();` 的含义是\_\_\_\_\_。

- A) p 是一个指向 int 型数组的指针
- B) p 是一个指针变量, 它构成了指针数组
- C) p 是一个指向函数的指针, 该函数的返回值是一个整型数据
- D) p 是一个指向函数的指针, 该函数的返回值是一个指向整型数据的指针

81. 说明语句 `void *f();` 的含义是\_\_\_\_\_。

- A) 函数 f 的返回值是一个通用型的指针
- B) 函数 f 的返回值可以是任意的数据类型
- C) 函数 f 没有返回值
- D) 指针 f 指向一个函数, 该函数无返回值

82. 已知 `char **s;` 正确的语句是\_\_\_\_\_。

- A) `s= "book"`                      B) `*s= "book"` ;
- C) `**s= "book"`                      D) `strcpy(*s, "book")` ;

83. 阅读以下函数, 说出每个函数的作用

(1) 计算 `1+2+3+.....+n`

```
sum(int n)
{ if(n==1) return 1;
  else return sum(n-1)+n;
}
```

(2) 计算字符串的长度

```
int strtry(char *s)
{ if(*s==' \0' ) return 0;
  else return strtry(s+1)+1;
}
```

(3) 有 n 个元素的数组逆置

```
void change(int *a, int n)
```

```

{ int t;
  t=*a; *a=*(a+n-1); *(a+n-1)=t;
  if(n>2) change(a+1,n-2);
}

```

#### (4) 计算字符串的长度

```

ABC(char *ps)
{ char *p;
  p=ps;
  while(*p++);
  return p-ps;
}

```

84. 对于基类型相同的两个指针变量之间，不能进行的运算是\_\_\_\_\_。

A) <                      B) =                      C) +                      D) -

85. 不合法的 main 函数命令行参数表示形式是

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

86. 写出下面各程序段的输出结果

(1)

```

static char a[]=" Basic" ;
char *p;
for(p=a; p<a+5; p++) printf( "%s\n", p);
for(p=a; p<a+5; p++) printf( "%c\n", *p);

```

Basic

asic

sic

ic

c

B

a

s

i

c

(2) a=36 b=16 c=19

```

void main()                                printf("a=%d, b=%d, c=%d", a, b, c)
{ int a=1, b=2, c;                          ;
  c=func(a, &b);                             }
  b=func(c, &a);                             int func(int a, int *p)
  a=func(b, &c);                             { a++;

```



```

        *p=a+2;
        return(*p+a);
    }
(3) 2 1 4 3
void fun (int *x,int *y)
{printf( "%d %d" ,*x,*y);
  *x=3;
  *y=4;
}

main()
{int x=1,y=2;
  fun(&y,&x);
  printf( "%d %d" , x,y);
}

(4) 5
ss(char *s)
{char *p=s;
  while (*p) p++;
  return (p-s);
}

main()
{ char *a= "abded" ;
  int i;
  i=ss(a);
  printf( "%d\n" , i);
}

(5) 0
void fun(int *n)
{ while((*n)--);
  printf( "%d" , ++(*n));
}

main()
{int a=100;
  fun(&a);
}

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

(7) 9
main()
{int a[2][3]={1,3,5,7,9,11},*s[2],**pp,*p;
  s[0]=a[0],s[1]=a[1];
  pp=s;
  p=(int*)malloc(sizeof(int));

```

```

    **pp= s[1][1];
    p=*pp;
    printf( "%d\n" ,*p);
}

```

#### (8) 字符串的长度

```

func( char str[] )                main()
{int num=0;                        {char str[10],*p=str;
    while (*(str+num)!= '\0' )    gets(p);
        num++;                    printf( "%d\n" ,func(p));
    return (num);                  }
}

```

#### (9) 6385

```

main()
{ char ch[2][5]={ "6934" , "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' && p[I][j]<= '9' ; j+=2)
      s=10*s+p[I][j]- '0' ;
  printf( "%d\n" , s);
}

```

#### (10) 4

```

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

```

### 87、填写程序

(1) 下列程序输出数组中的最大值，由 s 指针指向该元素

```

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

```

(2) 函数 sstrcmp() 的功能是对两个字符串进行比较。当 s 所指字符串和 t 所指字符串相等时，返回值为 0；当 s 所指字符串大于 t 所指字符串时，返回值大于 0；当 s 所指字符串小于 t 所指字符串时，返回值小于 0 (功能等同于库函数 strcmp())。

```

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

```

(3) 下面的程序完成的功能是：从键盘输入一行字符，反序输出。

```

#include<stdio.h>
struct node { char data; struct node *link;}*head;
main()
{ char ch;    struct node *p;
  head=NULL;
  while((ch==getchar())!=' \n' )
    { p=(struct node*)malloc(sizeof(struct node));
      p->data=ch;
      p->link=__head__;
      head=__p__;
    }
    __p=head__;
  while(p!=NULL)
    { printf( "%c" ,p->data); __p=p->link__; }
}

```

(4) 以下程序输入 10 本书的名称和单价，按照单价进行排序后输出。

```

#define NUM 10
struct book { char name[20]; int price;};
main()
{ struct book term, books[NUM];
  int count;
  for(count=0; count<NUM;count++ )

```

```

    { printf( "Please enter book name and price. book %d=", count+1);
      scanf( "%s%d%c", _ books[count].name, &books[count].price _);
    }
    sortbook(books, NUM);
    printf( "-----BOOK LIST-----\n" );
    for(count=0; count<NUM; count++);
        printbook(_books+count 或&books[count]__);
}
sortbook(struct book *pbook, int count)
{ int i;
  struct book temp,*q, __ *pb, *pend=pbook+count____;
  for(i=0; i<count-1; i++)
  { pb=pbook+i;
    for(q=pb; q<pend; q++)
      if(q->price>pb->price) __ pb=q __
      __ temp=*pb; *pb=*(pbook+i);*(pbook+i)=temp;__;
  }
}
printbook(struct book *pbook)
{ printf( "%-20s %6d\n", pbook->name, pbook->price); }

```

88. 以下对结构体类型变量的定义中，不正确的是\_\_\_\_\_。

- |                      |                         |
|----------------------|-------------------------|
| A) typedef struct aa | B) #define AA struct aa |
| {int n;              | AA{int n;               |
| float m;             | float m;                |
| }AA;                 | }td1;                   |
| AA td1;              |                         |
| C) struct            | D) struct               |
| {int n;              | {int n;                 |
| float m;             | float m;                |
| }aa;                 | }td1;                   |
| struct aa td1;       |                         |

89. 对于设有下列说明，则不正确的是\_\_\_\_\_。

```

struct ex
{int x; float y; char z;}example;

```

- |                      |                     |
|----------------------|---------------------|
| A) struct 是结构体类型的关键字 | B) example 是结构体类型名  |
| C) x, y, z 都是结构体成员名  | D) struct ex 是结构体类型 |

90. 以下选项中不能正确把 c1 定义成结构体变量的是\_\_\_\_\_。

- A) typedef struct  
{int red;  
int green;  
int blue;  
}COLOR;  
COLOR c1;
- B) struct color c1  
{int red;  
int green;  
int blue;  
};
- C) struct color  
{int red;  
int green;  
int blue;  
}c1;
- D) struct  
{int red; int green;  
int blue;  
}c1;

91. 已知:

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

若有 p=&data, 则对 data 中成员 a 的正确引用是\_\_\_\_\_。

- A) (\*p).data.a      B) (\*p).a      C) p->data.a      D) p.data.a

92. 若有说明:

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

则能正确定义结构体数组并赋初值的语句是\_\_\_\_\_。

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

93. 有说明:

```
struct person{char name[9]; int age};
```

```
struct person
class[5]={"Joju", 17, "Paul", 19, "Mary", 18, "Adam", 16};
```

能输出字母 M 的语句是\_\_\_\_\_。

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

94. 下列程序的输出结果是\_\_6\_\_。

```

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

```

95. 有以下说明和定义语句：

```

struct student { int age; char num[8];};
struct student stu[3]={20,"200401"},{21,"200402"},{19,"200403"};
struct student *p=stu;

```

引用结构体变量成员的错误表达式是\_\_\_\_\_。

- A) (p++)->num      B) p->num      C) (\*p).num      **D) stu[3].age**

96. 当说明一个结构体变量时系统分配给它的内存空间是\_\_\_\_\_。

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

97. 当说明一个共用体变量时系统分配给它的内存空间是\_\_\_\_\_。

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

98. 有以下说明和定义语句：

```

struct student { int num,age;};
struct student stu[3]={1001,20},{1002,19},{1003,21}};
struct student *p=stu;

```

错误表达式是\_\_\_\_\_。

- A) (p++)->num      B) p->age      C) (\*p).num      **D) p=&stu.age**

99. 已知学生结构为：

```

struct student{ int no; char name[20]; char sex;
  struct { int yaer,month,day; }birth;
}s;

```

以下\_\_\_\_\_是正确的赋值方式。

- A) year=1984; month=11; day=11;  
B) birth.year=1984; birth.month=11; birth.day=11;  
C) s.year=1984; s.month=11; s.day=11;  
**D) s.birth.year=1984; s.birth.month=11; s.birth.day=11;**

100. 若有以下程序段：

```

struct dent{ int n; int *m; };
int a=1,b=2,c=3;
struct dent s[3]={101,&a},{102,&b},{103,&c}};
struct dent *p=s;

```

则以下表达式中值为 2 的是\_\_\_\_\_。

- A) (p++)->m      B) \*(p++)->m      C) (\*p).m      D) **\*(++p)->m**

101. 定义:

```
typedef ARRAY1[10];
```

则整型数组 a[10]、b[10]、t[10]可以定义为\_\_**ARRAY a, b, c;**\_\_\_\_\_。

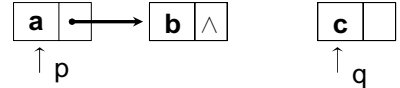
102. 若已建立下面的链表结构, 指针、分别指向图中所示结点, 则不能将所指的结点插入到链表尾的一组语句是\_\_\_\_\_。

- A) q->next=NULL; p=p->next; p->next=q;

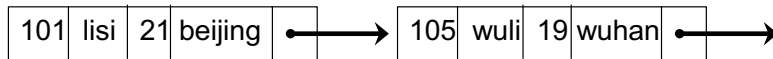
- B) p=p->next; q->next=p->next; p->next=q;

- C) p=p->next; q->next=p; p->next=q;**

- D) p=(\*p).next; (\*q).next=(\*p).next; (\*p).next=q;



103. 如何定义如下形式的链表?



```
struct xxxx
```

```
{ int num; char name[20]; int age; char addr[20];
```

```
    struct xxxx *next;
```

```
};
```

104. 若要使 STP s; 等价于 char \*s;, 应进行如下说明\_\_\_\_\_。

- A) typedef STP char \*s;

- B) typedef \*char STP

- C) typedrf STP \*char;

- D) typedef char \* STP;**

105. 若有如下定义, 则变量 a 所占的内存字节数是\_\_**10**\_\_

```
.uoion U {char st[6]; int i; long l;};
```

```
struct A { int c; union U u; }a;
```

106. 设有定义: enum t1 {a1,a2=7,a3,a4=15} time;

则枚举常量 a1 和 a3 的值分别是\_\_**0 8**\_\_。

107. 以下枚举类型的定义中正确的是\_\_\_\_\_。

- A) enum a={one, two, three};

- B) enum a {one, two, three};**

- C) enum a={ "one", "two", "three" };

- D) enum a { "one", "two", "three" };

108. 设 int i=2, j=1, k=3; 表达式 i&&(i+j)&k|i+j 的值为\_\_**1**\_\_。

109. 设 int b=2; 表达式 (b<<2)/(b>>1) 的值是\_\_**8**\_\_。

110. C 语言可以处理的文件类型是\_\_\_\_\_。

- 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) 程序文件和数据文件

111. 若要打开 A 盘 user 子目录下名为 abc.txt 的文本文件进行读、写操作, 应该调用函数\_\_\_\_\_。

- A) fopen("a:\user\abc.txt", "r");

- B) fopen("a:\\user\\abc.txt", "r+");**

C) `fopen("a:\user\abc.txt", "rb");` D) `fopen("a:\\user\\abc.txt", "w");`

112. 设有以下结构:

```
struct st
{ char name[8]; int num; float s[4]; } student[50];
```

并且数组 `student` 中的元素都已有值, 若要将这些元素写到文件指针 `fp` 指向的文件中, \_\_\_\_\_ 是错误的。

A) `fwrite(student, sizeof(struct st), 50, fp);`

B) `fwrite(student, 50*sizeof(struct st), 1, fp);`

C) `fwrite(student, sizeof(struct student), 50, fp);`

D) `for( i=0; i<50; i++) fwrite(student, sizeof(struct st), 1, fp);`

113. 若 `fp` 是指向某文件的指针, 且已读到文件末尾, 则函数 `feof(fp)` 的返回值是\_\_\_\_\_。

A) EOF B) -1 C) 1 D) NULL

114. 能把整数以二进制形式存放到文件中的函数是\_\_\_\_\_

A) `fprintf` B) `fread` C) `fwrite` D) `fputc`

115. 写出下面各程序段的输出结果

(1) 6

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

(2) hello

```
void fun(char *fname, char *st)
{ FILE *my; int i;
  my=fopen(fname, "w");
  for(i=0; i<strlen(st); i++)
    fputc(st[i], my);
  fclose(my);
}
main()
{ fun("test.t", "new world");
  fun("test.t", "hello");
}
```

(3) 3

```
#include<stdio.h>
main()
```



```

{ FILE *fp;
  int i, a[4]={1, 2, 3, 4}, b;
  fp=fopen("data.dat", "wb");
  for(i=0; i<4; i++)    fwrite(&a[i], sizeof(int), fp);
  fclose(fp);
  fp=fopen("data.dat", "rb");
  fseek(fp, -2L*sizeof(int), SEEK_END);
  fread(&b, sizeof(int), 1, fp);
  fclose(fp);
  printf( "%d" , b);
}

```