

主题作业一：基础知识

学号：

姓名：

一、单选题

1. C 语言程序中可以对程序进行注释，注释部分必须用符号_____括起来。
A、‘{ 和 }’ B、‘[和]’ C、“/*”和“*/” D、“*/”和“/*”
2. 下列运算符中，优先级最低的是_____。
A、* B、!= C、+ D、=
3. 若变量已正确定义，表达式 (j=3,j++) 的值是_____。
A、3 B、4 C、5 D、0
4. 已知字符 ‘a’ 的 ASCII 码为 97，执行下列语句的输出是_____。
printf("%d,%c", 'b', 'b'+1);
A、98,b B、语句不合法 C、98,99 D、98,c
5. 要调用数学函数时，在#include 命令行中应包含_____。
A、“stdio.h” B、“string.h” C、“math.h” D、“ctype.h”
6. 判断 i 和 j 至少有一个值为非 0 的表达式是_____。
A、i!=0 && j!=0 B、i!=0||j!=0 C、!(i==0 || j==0) D、i&& j
7. 若 a 是基本整型变量，c 是单精度实型变量，输入语句_____是错误的。
A、scanf("%d,%f", &a, &c); B、scanf("d=%d, c=%f", &a, &c);
C、scanf("%d%f", &a, &c); D、scanf("%d%f", a, c);
8. C 语言源程序文件经过编译连接之后生成一个后缀为_____的文件。
A、".c" B、".obj" C、".exe" D、".bas"
9. 以下选项中不正确的整型常量是_____。
A、12L B、-10 C、1,900 D、123U
10. 表达式_____的值是0。
A、3%5 B、3/5.0 C、3/5 D、3<5
11. 若变量已正确定义并赋值，表达式_____不符合C语言语法。
A、4&&3 B、+a C、a=b=5 D、int(3.14)
12. 执行语句 printf("_____", 2); 将得到出错信息。
A、%d B、%o C、%x D、%f
13. 若x是整型变量，表达式 (x=10.0/4.0) 的值是_____。
A、2.5 B、2.0 C、3 D、2
14. 若变量已正确定义并赋值，表达式 -k++ 等价于_____。
A、-(k++) B、(-k)++ C、-(k+1) D、-(++k)
15. 在C 语言程序中，在函数内部定义的变量称为_____。
A、全局变量 B、外部变量 C、局部变量 D、内部变量
16. 下列选项哪个不能作为整常数_____。
A、0xa B、010L C、10 D、1.0
17. 下列选项与表达式!(x>0||y>0)等价的是_____。
A、!(x>0)&&!(y>0) B、!x>0&&!y>0 C、!x>0||!y>0 D、!(x>0)||!(y>0)
18. 下列选项哪个表达式的值不为零_____。
A、1/2 B、!'0' C、!EOF D、NULL
19. 若x为浮点型，则表达式x=10/4为_____。
A、2.5 B、2.0 C、3 D、2

20. 下列表达式，错误的是_____。
- A. a&b B. a^b C. &&x D. a, b
21. 若 c 是一个字符变量，下列表达式经计算后哪个选项c的值是 1 或者 0 _____。
- A. c=getchar()!=EOF B. (c=getchar())!=EOF C. c = '0' D. c = c ^ ~c
22. 根据声明int x=23;表达式(x++*1/3)的值为_____。
- A. 7 B. 8 C. 15 D. 16
23. 根据声明int a=12;表达式a+=a- =a*=a的值为_____。
- A. -120 B. 144 C. 0 D. -132
24. 根据声明int a=10, b=20;表达式!a<b的值为_____。
- A. 1 B. 0 C. 10 D. 20
25. 根据声明int a;表达式(a&3)==(a%4)的值为_____。
- A. 0 B. 1 C. 不确定 D. 表达式错误
26. 下列代码段的输出为_____。
- ```
for(a=1,i=-1;-1<=i<1;i++)
{ a++; printf("%2d",a); }
printf("%2d",i);
```
- A. 2                      B. -1                      C. 0                      D. 3
27. 下列字符哪个可以作为字符常量\_\_\_\_\_。
- A. '\ '                      B. '\080'                      C. '%d'                      D. 0xa
28. 下列选项中用于判断 ch 是否是英文字母的表达式是\_\_\_\_\_。
- A. ( 'a' <= ch <= 'z') || ( 'A' <= ch <= 'Z')
- B. (ch >= 'a' && ch <= 'z') && ( ch >= 'A' && ch <= 'Z')
- C. (ch >= 'a' && ch <= 'z') || ( ch >= 'A' && ch <= 'Z')
- D. ( 'a' <= ch <= 'z') && ( 'A' <= ch <= 'Z')
29. 下列代码段的输出为 ( )
- ```
#include <stdio.h>
void main( )
{   int y;
    double d=3.4, x;
    x=(y=d/2.0)/2;
    printf(" (%0.2f, %d)", x, y);
}
```
- A. (0.00, 1) B. (0.85, 1.7) C. (0.50, 1) D. none of above
30. 下列代码段执行结束后，变量 s 的值为_____。
- ```
int i=5, s=0;
while(i-->0) { if (i%2) s+=i; else s+=i--;}
```
- A. 15                      B. 11                      C. 10                      D. 6
31. 若变量已正确定义并赋值，符合 C 语言语法的表达式是\_\_\_\_\_。
- A. a=2++                      B. a=3,5                      C. a=a+1=3                      D. 12.3%4
32. 表达式 5>3>1 值是\_\_\_\_\_。
- A. 0                      B. 1                      C. 3                      D. 表达式语法错
33. 已知字母 A 的 ASCII 码为十进制数 65, 且 ch 为字符型变量，则执行语句 ch='A'+ '6'-'3'; 后，ch 中的值为\_\_\_\_\_。
- A. D                      B. 68                      C. C                      D. 不确定

二. 填空题

34. 表达式  $(7 < 1 > 2^2)$  的值是\_\_\_\_\_。
35. 表达式  $!!5$  的值是\_\_\_\_\_。
36. 写出一个条件表达式来计算  $x$  和  $y$  的最大值\_\_\_\_\_。
37. 表达式  $7 * 7 / 9 * 4 \% 3$  的值为\_\_\_\_\_。
38. 表达式  $10 == 10 == 10$  的值为\_\_\_\_\_。
39. 表达式  $1 < 10 < 5$  的值为\_\_\_\_\_。
40. 表达式  $\sim(10 < 1) \& 4$  的值为\_\_\_\_\_。
41. 若有定义:  $\text{int } x=2, y=5$ ; 则表达式  $x*=y-3$  的值为\_\_\_\_\_。
42. 写出计算  $y$  的条件表达式\_\_\_\_\_。

$$y = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

43. 写出下列代码段的输出结果\_\_\_\_\_。

```
float x1, x2;
x1=3/2;
x2=x1/2;
printf("%d, %.1f", (int)x1, x2);
```

44. 写出下列程序段的输出结果\_\_\_\_\_。

```
int k, x;
for (k=0, x=0; k<=9&&x!=10; k++)
 x+=2;
printf ("%d, %d", k, x);
```

45. 下列代码段将会打印出\_\_\_\_\_。

```
printf("%#o\n", (010 < 1^1) & (0xe < 1));
```

46. 下列代码段的输出结果是\_\_\_\_\_。

```
int k, s;
for(k=0, s=0; k<10; k++)
 if (s>k) break;
 else s += k;
printf("k=%d s=%d", k, s);
```

47. 下列代码段的输出为\_\_\_\_\_。

```
int k, s;
for(k=1, s=0; k<10; k++){
 if (k%2==0) continue;
 s+= k;
}
printf("k=%d s=%d", k, s);
```

48. 下列代码段将会打印出\_\_\_\_\_。

```
int a = 3, b;
if (a > 5)
 b = 7;
else if (a < 2)
 b = 6;
else
 b = 5;
printf("%d", b);
```

49. 下列代码段将会打印出\_\_\_\_\_。

```
int a = 13;
int b = 3;
float x;
x = a / b;
printf("%.1f", x);
```

50. 下列代码段将会打印出\_\_\_\_\_。

```
int j = 1, k = 2;
printf ("%d, %d#", j++, ++k);
printf ("%d, %d", --j, k--);
```

51. 下列代码段的输出为\_\_\_\_\_。

```
char x='0',y='1';
switch(!x) {
 case 0: printf("**0*#");break;
 case 1:
 switch(y){
 case '0': printf("**1*#");break;
 case '1': printf("**2*#");break;
 }
 default: printf("**3*#");
}
```

52. 假定 a=0, b=3, c=0, d=26.下列代码段中 d 的值为为\_\_\_\_\_。

```
if(!b) d=d-10;
else if(a)
if(!c) x=15;
else x=25;
```

53. 输入 **abcde#<回车>**，下列代码段的输出结果是\_\_\_\_\_。

```
char c;
for(c=getchar(); getchar()!='#'; c=getchar())
 putchar(c);
```

三. 阅读理解题

54. 若输入 **-6+15\*3/5=<回车>**， 以下程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
main()
{
 int m=0, sum=0;
 char c, oldc='+';

 do {
 c=getchar();
 if(c<='9'&& c>='0')
 m=10*m+c - '0';
 else {
 if(oldc=='+')
 sum += m;
 else
 sum -= m;
 m=0;
 oldc=c;
 printf("%d#", sum);
 }
 } while(c!='=');
}
```

55. 下列程序的输出为\_\_\_\_\_。

```
include <stdio.h>
int f()
{ static int k;
 return ++k;
}
void main()
{ int k;
 for(k=0;k<3;k++)
 printf("%d#", f());
}
```

56. 下列程序的输出为\_\_\_\_\_。

```
#include <stdio.h>
void main()
{ int k, x, s, t;
 x=3;
 s=0; t=x;
 for(k=1; k<=3; k++){
 t=t*10+x;
 s=s+t;
 printf("%d#", s);
 }
}
```

```

 }
}

```

57. 若输入为 how are you? <ENTER>, 则输出为\_\_\_\_\_。

```

#include <stdio.h>
void main()
{
 int word;
 char ch;
 word=0;
 while((ch=getchar())!='?'){
 if(ch==' ') word=0;
 else if(word==0){
 word=1;
 if(ch<='z'&&ch>='a') ch=ch-'a'+'A';
 }
 putchar(ch);
 }
}

```

58. 下列程序将会打印出\_\_\_\_\_。

```

#include "stdio.h"
void main()
{
 int j, k;
 for (j=0; j<3; j++){
 for (k=0; k<3; k++)
 printf("%2d", (k+j+1)% 3 ? (k+j+1)%3 : 3);
 printf("#");
 }
}

```

59. 下列程序的输出为\_\_\_\_\_。

```

#include <stdio.h>
void main()
{
 int i,j,k=19;
 while (i=k-1) {
 k-=3;
 if(k%5==0) { i++; continue; }
 else if(k<5) break;
 i++;
 }
 printf("i=%d,k=%d\n",i,k);
}

```

## 主题作业二：数组与指针

### 一. 单选题

- 数组定义为 `int a[3][2]={1, 2, 3, 4, 5, 6}`, 数组元素\_\_\_\_\_的值为 6。  
A、`a[3][2]`      B、`a[2][1]`      C、`a[1][2]`      D、`a[2][3]`
- 不正确的赋值或赋初值的方式是\_\_\_\_\_。  
A、`char str[]="string";`  
B、`char str[7]={ 's', 't', 'r', 'i', 'n', 'g' };`  
C、`char str[10]; str="string";`  
D、`char str[7]={ 's', 't', 'r', 'i', 'n', 'g', '\0' };`
- 设变量定义为 `char s[ ]="hello\nworld\n"`, 则数组 `s` 中有\_\_\_\_\_个元素。  
A. 12      B. 13      C. 14      D. 15
- 下列代码段的输出为\_\_\_\_\_。  
`char c[]="\\t\\r\\0will\\n";  
printf("%d",strlen(c));`  
A. 4      B. 15      C. 16      D. 11
- 下列代码段输出字符串 `str`, 则循环条件 `str[i]!='\0'` 能够被下列哪个选项替换\_\_\_\_\_。  
`char str[80]="hello, world";  
for (i = 0; str[i] != '\0'; i++)  
    putchar(str[i]);`  
A. `str[i]`      B. `i < 80`      C. `!(str[i] = '\0')`      D. `i <= 80`
- 关于数组声明“`int a[4];`”, 下列用法错误的是\_\_\_\_\_。  
A. `*a`      B. `a[0]`      C. `a`      D. `a++`
- 下列关于数组的定义, 哪种用法是错误的\_\_\_\_\_。  
A. `int a[1][3];`      B. `int x[2][2]={1,2,3,4};`  
C. `int x[2][]={1,2,4,6};`      D. `int m[][3]={1,2,3,4,5,6};`
- 以下不正确的定义语句是\_\_\_\_\_。  
A. `double x[5]={2.0,4.0,6.0,8.0,10.0};`      B. `char c2[]={ '\x10', '\xa', '\x8' };`  
C. `char c1[]={ '1', '2', '3', '4', '5' };`      D. `int y[5]={0,1,3,5,7,9};`
- 若变量已正确定义并且指针 `p` 已经指向某个变量 `x`, 则 `(*p)++` 相当于\_\_\_\_\_。  
A、`p++`      B、`x++`      C、`*(p++)`      D、`&x++`
- 若 `p1`、`p2` 都是整型指针, `p1` 已经指向变量 `x`, 要使 `p2` 也指向 `x`, \_\_\_\_\_是正确的。  
A、`p2=p1;`      B、`p2=**p1;`      C、`p2=&p1;`      D、`p2=*p1;`
- 下列程序段的输出是\_\_\_\_\_。  
`int c[]={1, 3, 5};  
int *k=c+1;  
printf("%d", *++k);`  
A、 3      B、 5      C、 4      D、 6
- 对于如下说明, 语法和语义都正确的赋值是\_\_\_\_\_。  
`int c, *s, a[]={1, 3, 5};`  
A、`c=*s;`      B、`s[0]=a[0];`      C、`s=&a[1];`      D、`c=a;`
- 设变量定义为 `int a[2]={1,3}, *p=&a[0]+1;`, 则 `*p` 的值是\_\_\_\_\_。  
A. 2      B. 3      C. 4      D. `&a[0]+1`

14. 下列程序段的输出结果是\_\_\_\_\_。
- ```
int *p, *q, k = 1, j=10 ;
p=&j; q = &k ; p = q ; (*p)++;
printf("%d",*q);
```
- A. 1 B. 2 C. 10 D. 11
15. 根据声明int a[10], *p=a;, 下列表达式错误的是_____。
- A. a[9] B. p[5] C. *p++ D. a++
16. 下列表达式, 错误的是_____。
- A. char str[10]; str="string"; B. char str[]="string";
C. char *p="string"; D. char *p; p="string";
17. 根据声明int a[10], *p=&a[1];数组 a 的最后一个元素是_____。
- A. a[10] B. p[8] C. p[9] D. p[10]
18. 下列代码段将会打印出 ()
- ```
char s[]="student";
printf("%s%d", s+3, sizeof(s));
```
- A. student7                      B. dent7                      C. dent8                      D. student8
19. 下列语句定义pf为指向float类型变量f的指针, \_\_\_\_\_是正确的。
- A. float f, \*pf = f;                      B. float f, \*pf = &f;  
C. float \*pf = &f, f;                      D. float f, \*pf =0.0;
20. 假设scanf语句执行时输入**ABCDE<回车>**, 能使puts(s)语句正确输出ABCDE字符串的程序段是\_\_\_\_\_。
- A. char s[5]="ABCDE";puts(s)                      B. char s[5]={'A','B','C','D','E'}; puts(s);  
C. char \*s;scanf("%s",s);puts(s);                      D. char \*s; s="ABCDE";puts(s);

## 二. 填空题

21. 根据声明 int a[2][3][4], 该数组元素数量为\_\_\_\_\_。

22. 下列代码段将会打印出\_\_\_\_\_。

```
int array [] = {2, 8, 6, 11, 13, 2, 8, 2};
int k;
int total = 0;
for (k = 0; k < 8; k++) {
 if (array [k] % 2) break;
 total += array [k];
}
printf ("%d", total);
```

23. 下列代码段的输出为\_\_\_\_\_。

```
char st[20]="hello\0world!";
printf("%d,%d\n",strlen(st),sizeof(st));
```

24. 写出下列程序段的输出结果。

```
char str[]="hello\tworld\n";
printf("%d, %c\n", sizeof(str), *(str+10));
```

25. 下列代码段的输出为\_\_\_\_\_。

```
int c[]={10, 30, 5}, *pc;
for(pc=c; pc<c+2; pc++) printf("%d#", *pc);
```



26. 下列代码段的输出为\_\_\_\_\_。

```
int a = 1;
int b = 2;
int *x, *y;
x = &a;
y = &b;
*x = *x + 1;
x = y;
*x = *x + 1;
printf("%d, %d", a, b);
```

27. 下列代码段的输出为\_\_\_\_\_。

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

28. 下列代码段

```
int i=0,s=0,a[10];
while((i<10)&&(a[i]>=0))
 s+=a[i++];
```

等价于代码段

```
int s,a[10],*p=a;
for(_____)
 if(*p<0) break;
```

29. 函数 f 1 ( ) 等价于函数 f 2 ( )

```
void f1(char s[], char t[])
{
 int i=0;
 while (t[i]!='\0') {s[i]=t[i];i++;}
 s[i]=0;
}

void f2(char *s, char *t)
{
 while (____);
}
```

30. 下列程序段输出 \_\_\_\_\_。

```
char s[]="abc\0def\0ghi",*sp=s;
printf("%s",sp+5);
```

三、 阅读理解题

31. 若输入 3 1 2 3 <回车> , 则以下程序的输出结果是\_\_\_\_\_。

```
#include "stdio.h"
void main()
{
 int i, j, n, a[10];
 scanf("%d", &n);
 for (i=0; i<n; i++)
```

```

 scanf("%d",&a[i]);
 for (i=0; i<n; i++)
 for (j=0; j<n; j++)
 printf("%d, ", a[(i+j)%n]);
}

```

32. 输入 123+234=357Yes?<回车>后，下列程序的输出结果是\_\_\_\_\_。

```

#include <stdio.h>
void main()
{
 int k, n1, n3[10];
 char c;
 n1=0;
 for(k=0; k<10; k++)
 n3[k]=0;
 while((c=getchar())!='\n'){
 switch(c){
 case'0': case '1': case '2': case '3': case '4':
 case'5': case '6': case '7': case '8': case '9':
 n3[c-'0']++;
 break;
 default:
 n1++;
 break;
 }
 }
 if(n1) printf("%d ", n1);
 for(k=0; k<10; k++)
 if(n3[k]) printf("%d ", n3[k]);
}

```

33. 若输入为 7 3 0 3 0 3 1 2 9 7 6 0<ENTER>，则输出为\_\_\_\_\_。

```

#include <stdio.h>
void main()
{
 int j, k, sub, x;
 int a[5];

 for(j=1; j<=2; j++){
 for(k=0; k<5; k++)
 scanf("%d", &a[k]);
 scanf("%d", &x);
 sub=-1;
 for(k=0; k<5; k++)
 if(a[k]==x) sub=k;
 printf("%d#", sub);
 }
}

```

34. 下列程序的输出为\_\_\_\_\_。

```
#include <stdio.h>
void main()
{ int i, k,m;
 int a[8]={60,75,63,92,87,64,79,21}, s[10];

 for(k=0; k<10; k++)
 s[k]=0;
 for(i=0; i<8; i++){
 k=a[i]/10;
 s[k]++;
 }
 m=s[0];
 k=1;
 while (k<10){
 if(s[k]!=0) printf("%d#", s[k]);
 if (s[k]>m) m=s[k];
 k++;
 }
 printf("%d", m);
}
```

35. 若输入 **AabD <ENTER>**，则下列程序的输出为\_\_\_\_\_。

```
#include <stdio.h>
void main()
{
 char s[81];
 int i=0;
 gets(s);
 while (s[i]!='\0'){
 if(s[i]<='z'&& s[i]>='a')
 s[i]= 'z'+ 'a'-s[i];
 i++;
 }
 puts(s);
}
```

36. 下列程序运行时输入：**8 3 7 2 1 8 3 4 1 <回车>**，输出结果为\_\_\_\_\_。

```
#include <stdio.h>
void f(int a[], int n);
main()
{
 int i, n, a[10];

 scanf("%d", &n);
 for (i=0; i<n;i++)
```

```

 scanf("%d",&a[i]);
 f(a+n/4,n/2);
 for (i=0; i<n; i++)
 printf("%d#",a[i]);
}

void f(int a[], int n)
{
 int i, j, temp;

 for(i = 1; i < n; i++)
 for (j = 0; j < n-i; j++)
 if (a[j] > a[j+1]) { temp = a[j]; a[j] = a[j+1]; a[j+1] = temp; }
}

```

37. 下列程序的输出为\_\_\_\_\_。

```

#include <stdio.h>
int z;
void p(int *x,int y)
{
 ++ *x;
 y ++;
 z = *x+y;
 printf("%d,%d,%d#",*x, y, z);
}
void main()
{
 int x=1, y=2, z=3;
 p(&y, x);
 printf("%d,%d,%d#",x, y, z);
}

```

38. 下列代码段的输出为\_\_\_\_\_。

```

#include <stdio.h>
int f(int a[], int n, int m)
{
 int i, j, sum, max=0;
 for(j=0; j<m; j++) max+=a[j];
 for (i=1; i<n; i++) {
 sum=0;
 for (j=0; j<m; j++)
 sum+= *(a+i*m+j);
 if (sum>max) max=sum;
 }
 return max;
}
void main()
{

```

```

 static int a[]={1,2,3,4,5,6,7,8,9,10,11,12};
 printf("%d", f(a, 3, sizeof(a)/sizeof(int)/3);
 }

```

39. 下列程序运行时输入: **C Java C++ VB Fortran Basic** <回车>, 输出结果为\_\_\_\_\_。

```

#include <stdio.h>
void find(char *str)
{
 char *p=str;
 int k1=0,k2=0;

 while(1) {
 if(*p!=' ' && *p!='\t' && *p!='\0') k2++;
 else {
 if (k2>k1) { k1=k2; str=p-k2;}
 k2=0;
 if (*p=='\0') break;
 }
 p++;
 }
 for(k2=0;k2<k1;k2++) printf("%c",*str++);
 return;
}

main()
{
 char a[80];
 gets(a);
 find(a);
}

```

40. 下列代码段的输出为\_\_\_\_\_。

```

#include <stdio.h>
void p(int *x,int y)
{
 ++ *x;
 y=y+2;
}
void main()
{
 int x=0, y=3;
 p(&y, y);
 printf("%d, %d", x, y);
}

```

## 主题作业三：函数与结构

### 一. 单选题

1. 针对如下定义，合法的表达式是\_\_\_\_\_。  

```
struct node {
 char s[10];
 int k;
} p[5];
```

A. p.k=2                      B. p[0]->k=2                      C. (p->s)[0]='a'                      D. p[0].s="a"
2. 针对如下定义：  

```
static struct {
 int x, y[3];
} a[3] = {{1,2,3,4},{5,6,7,8},{9,10,11,12}}, *p;
p = a+1;
```

表达式\*((int\*)(p+1)+2)的值为\_\_\_\_\_。  
A. 3                      B. 7                      C. 10                      D. 11
3. 对于以下结构定义，++p->str 中的++加在\_\_\_\_\_。  

```
struct {
 int len;
 char *str;
} *p;
```

A、指针 str 上                      B、指针 p 上                      C、str 指的内容上                      D、以上均不是
4. 根据声明 int (\*p)[10], p 是一个\_\_\_\_\_。  
A. 指针                      B. 数组                      C. 函数                      D. 数组元素
5. 若下面程序中所有的变量均已声明或定义，则下列选项中的变量能够在 fun()中使用的  
是\_\_\_\_\_。  

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

 return;
}
int z;
void main()
{
 int a,b;
 fun(a);

}
```

A. x, y                      B. x, y, z                      C. a,b,y,z                      D. a,b,x,y,z

### 二. 填空题

6. 对于下面的定义，(s[0].b)/(++p)->a 的值为\_\_\_\_\_。

```

struct {
 int a;
 int b;
} s[2]={2,4,6,8}, *p=s;

```

7. 下列程序段执行后，z 的值是 \_\_\_\_\_。

```

static struct {
 int x, y[3];
} a[3] = {{1,2,3,4},{5,6,7,8},{9,10,11,12}}, *p=a+3;
int z;
z=((int *)p-1)-1;

```

8. 下列代码段将会打印出\_\_\_\_\_。

```

char *c[3]={"FIRST", "SECOND", "THIRD"};
printf("%s", *(c+1)+2);

```

9. 下列程序段的输出结果是\_\_\_\_\_。

```

char *st[]={"abcd","efgh","ijkl","mnop"};
printf("%s", *(st+1)+1);

```

10. 下列代码段的输出为\_\_\_\_\_。

```

char *st[]={"ONE","TWO","FOUR","K"};
printf("%s, %c\n", *(st+2), **st+1);

```

11. 下列代码段的输出\_\_\_\_\_。

```

char *a[]={"678","45"}, **p=a+1;
printf("%s,%c", *p, **p-1);

```

12. 对于数组 a，\*(a[1]+1)的值为\_\_\_\_\_。

```

int a[3][2]={1,2,3,4,5,6};

```

13. 对于以下递归函数 f，调用 f(3) 的返回值是\_\_\_\_\_。

```

f(int n)
{ return ((n>0) ? 2*f(n-1)+f(n-2) : -1); }

```

14. 下列代码段将会打印出\_\_\_\_\_。

```

int i;
int f(int x)
{ static int k = 0;
 x+=k++;
 return x;
}
i=f(2);
i=f(3);
printf("%d",i);

```

15. 下列代码段将会打印出\_\_\_\_\_。

```

int f(int x)
{ return ((x>0)? x*f(x-1):3); }
printf("%d",f(f(1)));

```

16. 用 typedef 写出类型定义\_\_\_\_\_，使得 AIP 表示含有 10 个元素的整型指针数组类型。

17. 假定可执行程序文件名字为 prog，则运行命令：prog hello world 后的输出为\_\_\_\_\_。

```
#include <stdio.h>
int main(int argc, char *argv[]) {
 printf("%s#%s#", argv[0], argv[argc - 1]);
 return 0;
}
```

18. 下列程序段的输出是\_\_\_\_\_。

```
#define FB(a,b) (a*b+1)
int k=3;
k= FB(k+1,k-1)-9;
printf("%d",k);
```

### 三、 阅读理解题

19. 下列程序的输出为\_\_\_\_\_。

```
#include <stdio.h>
main()
{
 int s=0,i,a[3][3]={1,2,3,4,5,6,7,8,9};
 for(i=0; i<3; i++)
 s+= *(*a+i);
 printf("%d#", s);
 for(i=0; i<3; i++)
 s+= **(a+i);
 printf("%d#", s);
}
```

20. 下列程序的输出是\_\_\_\_\_。

```
int f (int x)
{
 if(x<=1) return 1;
 else return f(x-1)+f(x-2);
}

void main()
{
 printf("%d", f(4));
}
```

21. 下列程序的输出是\_\_\_\_\_。

```
#include <stdio.h>
int main (void)
{
 int k = 1;
 int a_function (int j);
 k = a_function (k);
 printf ("%d" , k);
}

int a_function (int j)
{
 if (j < 3) {
 j++;
 }
}
```



```

 j = a_function (j);
 }
 printf ("%d#" , j);
 return (j);
}

```

22. 下列程序的输出为\_\_\_\_\_。

```

#include <stdio.h>
void melon (int g, int * h);
int main (void)
{ int a = 1, b = 2;
 melon (a, &b);
 printf ("a = %d, b = %d", a, b);
}
void melon (int b, int * c)
{ b++;
 *c = *c + b;
}

```

23. 下列程序的输出为\_\_\_\_\_。

```

#include <stdio.h>
#include<math.h>
int prime(int n)
{
 int i,m;
 if(n==1) return 0;
 m=sqrt(n);
 for(i=2;i<=m;i++)
 if(n%i==0) break;
 return i>m;
}
void main()
{ int num,i;
 num=20;
 for(i=2;i<=num;i++){
 while(prime(i)&&(num%i==0)){
 printf("%d ",i);
 num/=i;
 }
 }
}

```

24. 下列程序的输出为\_\_\_\_\_。

```

#include <stdio.h>
void f(int *x,int *y)
{ int *p;
 p=x; x=y; y=p;
}

```

```

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

```

25. 若输入 **this is a test.<ENTER>**，则下列程序的输出\_\_\_\_\_。

```

#include <stdio.h>
#define TRUE 1
#define FALSE 0
int change(char *c,int status);
void main()
{
 int flag=TRUE;
 char ch;
 do{
 ch=getchar();
 flag=change(&ch,flag);
 putchar(ch);
 } while(ch!='.');
 printf("\n");
}
int change(char *c,int status)
{
 if(*c==' ') return TRUE;
 if(status&&*c<='z'&&*c>='a') *c+= 'A'-'a';
 return FALSE;
}

```

26. 下列程序的输出为\_\_\_\_\_。

```

#include <stdio.h>
int f(int m)
{
 static int k=0;
 int s=0;
 for(; k<=m; k++) s++;
 return s;
}
void main()
{
 int s=1;
 s=f(4);
 printf("%d#%d#", s, f(2));
}

```

27. 下列程序的输出为\_\_\_\_\_。

```

#include <stdio.h>
int fun(int x)

```

```

{
 int t;
 if(x<=0)
 t=x;
 else
 t=fun(x-1)+fun(x-2);
 return t;
}
void main()
{
 int i;
 for(i=1;i<=3;i++)
 printf("%d#", fun(i));
}

```

28. 下列程序将会打印出\_\_\_\_\_。

```

#include <stdio.h>
#define my_square_add(a, b) (a * a + b * b)
#define my_square_sub(a, b) (a * a - b * b)
int main()
{
 int a = 1, b = 2;
 printf("%d#", my_square_add(a + b, b));
 printf("%d#", my_square_sub(a + b, b));
 return 0;
}

```

29. 下列程序的输出结果是\_\_\_\_\_。

```

include <stdio.h>
int f(int x)
{
 static int k=0;
 return ++k+x;
}
main()
{
 int k;
 for(k=0;k<3;k++) printf("%d#", f(k));
}

```

30. 下列命令行参数程序生成的执行程序为 test.exe，执行 **test 123 abc<回车>**，输出结果是\_\_\_\_\_。

```

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

```

31. 假设有下列函数定义：

```
void foo(int sum)
{
 int j;
 for (j = 0; j < 10; ++j)
 sum += (j + 1) * sum;
}
```

在main函数中有如下代码段：

```
int sum = 0;
```

```
foo(sum);
```

```
printf("%d", sum);
```

执行后，输出结果为\_\_\_\_\_。

## 主题作业四：指针进阶与文件

### 一. 选择题

1. 下列声明有效的是 ( )

```
struct node {
 char s[10];
 int k;
} p[5];
```

A. p.k=2      B. p[0]->k=2      C. (p->s)[0]='a'      D. p[0].s="a"

2. static struct {

int x, y[3];

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

p = a+1;

表达式\*((int\*)(p+1)+2)的值为 ( )

A. 3      B. 7      C. 10      D. 11

### 二. 填空题

3. 对于下面的定义, (s[0].b)/(++p)->a 的值为 ( )

```
struct {
 int a;
 int b;
} s[2]={2,4,6,8}, *p=s;
```

4. 下列程序段执行后, z 的值是 \_\_\_\_\_。

```
static struct {
 int x, y[3];
} a[3] = {{1,2,3,4},{5,6,7,8},{9,10,11,12}}, *p=a+3;
int z;
z=((int*)(p-1)-1);
```

5. 已知函数原型 void print(FILE \*input, const char \*output);其中, input 为输入流, output 为输出文件名, 你需要打印标准输入流 (stdin) 到文件"temp.txt", 请写下处理这项任务的语句 ( )

### 三. 程序阅读题

6. 若下列数据存放在文本文件 a.txt 里

one?two?1234?output?

下列数据存放在文本文件 b.txt 里

one?two?1204?input?

则下列程序的输出为 ( )

```
#include <stdio.h>
#include <stdlib.h>
void main()
{ int count;
 char ch1,ch2;
```

```

FILE *f1, *f2;

 if ((f1 = fopen("a.txt", "r")) == NULL){
 printf("Can't open file : %s\n", "a.txt");
 exit(0);
 }
 if ((f2 = fopen("b.txt", "r")) == NULL){
 printf("Can't open file : %s\n", "b.txt");
 exit(0);
 }
 count=0;
 while (!feof(f1)||!feof(f2)){
 ch1=fgetc(f1);
 ch2=fgetc(f2);
 if(ch1!=ch2){
 printf("%c#%c#", ch1, ch2);
 printf("%d#", count);
 break;
 }
 if(ch1=='?') count++;
 }
 fclose(f1);
 fclose(f2);
}

```

7. 文本文件 **in.txt** 包含下列三行数据，

*This is a demo*

*hello world*

*C is fun*

在下面这个命令行程序 find.c 中，在运行 “find -n is” 后程序的输出为 ( )

```

#include <stdio.h>
void main(int argc, char *argv[])
{
 char line[1000];
 int lineno=0;
 FILE *fp;
 if((fp=fopen("in.txt", "r")) == NULL) {
 printf("File in.txt Open Error\n");
 exit(1);
 }
 if(argc==2) {
 while(fgets(line, 1000, fp)!=NULL)
 if (strstr(line, argv[1]) != NULL) {
 printf("%s", line);
 }
 }
}

```

```

 }else if(argc==3 && strcmp(argv[1],"-n")==0){
 while(fgets(line,1000,fp)!=NULL) {
 lineno++;
 if (strstr(line, argv[2]) != NULL)
 printf("%d: %s",lineno,line);
 }
 }else
 printf("Usage: find [-n] pattern\n");
}

```

#### 四. 程序（段）填空题

1. 下面的程序通过命令行参数方式比较两个文本文件是否完全相同，比较方式 **prog file1 file2**。

```

#include <stdio.h>
main(int argc, char *argv[])
{
 FILE *fp1, *fp2;
 char c1, c2;

 if (argc!=(8)) {
 printf("Usage: prog file1 file2\n");
 exit(0);
 }
 if ((fp1=fopen(argv[1], "r")) == NULL || (9)) {
 printf("Cannot open files\n");
 exit(0);
 }
 c1=fgetc(fp1);
 c2=fgetc(fp2);
 while ((10) && c1==c2) {
 c1=fgetc(fp1);
 c2=fgetc(fp2);
 }
 if ((11))
 printf("The two files are different!\n");
 else
 printf("The two files are the same!\n");
 (12)
 fclose(fp2);
 return 0;
}

```

## 主题作业一：基础知识

学号：

姓名：

### 一、单选题

- C 语言程序中可以对程序进行注释，注释部分必须用符号\_\_C\_\_括起来。  
A、‘{’和’}’ B、‘[’和’]’ C、“/\*”和”\*/” D、“\*/”和”/\*”
- 下列运算符中，优先级最低的是\_\_D\_\_。  
A、\* B、!= C、+ D、=
- 若变量已正确定义，表达式 (j=3,j++) 的值是\_\_3\_\_。  
A、3 B、4 C、5 D、0
- 已知字符 ‘a’ 的 ASCII 码为 97，执行下列语句的输出是\_\_D\_\_。  
printf(“%d,%c”,’b’,’b’+1);  
A、98,b B、语句不合法 C、98,99 D、98,c
- 要调用数学函数时，在#include 命令行中应包含\_\_C\_\_。  
A、”stdio.h” B、”string.h” C、”math.h” D、”ctype.h”
- 判断 i 和 j 至少有一个值为非 0 的表达式是\_\_B\_\_。  
A、i!=0 && j!=0 B、i!=0||j!=0 C、!(i==0 || j==0) D、i&&j
- 若 a 是基本整型变量，c 是单精度实型变量，输入语句\_\_D\_\_是错误的。  
A、scanf(“%d,%f”, &a, &c); B、scanf(“d=%d, c=%f”, &a, &c);  
C、scanf(“%d%f”, &a, &c); D、scanf(“%d%f”, a, c);
- C 语言源程序文件经过编译连接之后生成一个后缀为\_\_C\_\_的文件。  
A、”.c” B、”.obj” C、”.exe” D、”.bas”
- 以下选项中不正确的整型常量是\_\_C\_\_。  
A、12L B、-10 C、1,900 D、123U
- 表达式\_\_C\_\_的值是 0。  
A、3%5 B、3/5.0 C、3/5 D、3<5
- 若变量已正确定义并赋值，表达式\_\_D\_\_不符合 C 语言语法。  
A、4&&3 B、+a C、a=b=5 D、int(3.14)
- 执行语句 printf(“\_\_D\_\_”, 2); 将得到出错信息。  
A、%d B、%o C、%x D、%f
- 若 x 是整型变量，表达式 (x=10.0/4.0) 的值是\_\_D\_\_。  
A、2.5 B、2.0 C、3 D、2
- 若变量已正确定义并赋值，表达式 -k++ 等价于\_\_A\_\_。  
A、-(k++) B、(-k)++ C、-(k+1) D、-(++k)
- 在 C 语言程序中，在函数内部定义的变量称为\_\_C\_\_。  
A、全局变量 B、外部变量 C、局部变量 D、内部变量
- 下列选项哪个不能作为整常数\_\_D\_\_。  
A、0xa B、010L C、10 D、1.0
- 下列选项与表达式!(x>0||y>0)等价的是\_\_A\_\_。  
A、!(x>0)&&!(y>0) B、!x>0&&!y>0 C、!x>0||!y>0 D、!(x>0)||!(y>0)
- 下列选项哪个表达式的值不为零\_\_B\_\_。  
A、1/2 B、!’\0’ C、!EOF D、NULL
- 若 x 为浮点型，则表达式 x=10/4 为\_\_B\_\_。  
A、2.5 B、2.0 C、3 D、2

注意 C 和 D 的左右配对顺序

! → 算术 → 关系 → && → || → 赋值

逗号表达式从左向右结合，最后一个表达式的值为逗号表达式的值。

表达式 j++ 的值为 j 增加之前的值，即 3

i 不等于 0，或者 j 不等于 0

输入需要提供变量的地址

+a 是合法的，等价于 a

-和++是两个优先级相同的单目运算符，但单目运算符从右到左结合

EOF 是宏定义的  
#define EOF (-1)

10/4 结果为 2



20. 下列表达式, 错误的是\_\_C\_\_。

- A. a&b                      B. a^b                      C. &&x                      D. a, b

&&为双目运算符,  
需要两个操作数

21. 若 c 是一个字符变量, 下列表达式经计算后哪个选项c的值是 1 或者 0 \_\_A\_\_。

- A. c=getchar()!=EOF    B. (c=getchar())!=EOF    C. c='0'    D. c=c^~c

22. 根据声明int x=23;表达式(x++\*1/3)的值为\_\_A\_\_。

- A. 7                      B. 8                      C. 15                      D. 16

表达式 x++ 的值为 23, 不是 24

23. 根据声明int a=12;表达式a+=a- =a\*=a的值为\_\_C\_\_。

- A. -120                      B. 144                      C. 0                      D. -132

赋值运算从右到左结合  
a+=(a=(a\*=a));  
a\*=a 为 144  
a-=144 为 0  
a+=a 为 0

24. 根据声明int a=10, b=20;表达式!a<b的值为\_\_A\_\_。

- A. 1                      B. 0                      C. 10                      D. 20

!先于<  
!a<b 等价于  
(!a)<b

25. 根据声明int a;表达式(a&3)==(a%4)的值为\_\_B\_\_。

- A. 0                      B. 1                      C. 不确定                      D. 表达式错误

26. 下列代码段的输出为\_\_B\_\_。

```
for(a=1,i=-1;-1<=i<1;i++)
{ a++; printf("%2d",a); }
printf("%2d",i);
```

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

-1<=i<1 ↔ (-1<=i)<1 ↔ (-1<=i)==0  
↔ !(1<=i) ↔ i<-1

a&3 和 a%4 的结果  
都是 a 的最后两个  
bit

27. 下列字符哪个可以作为字符常量\_\_D\_\_。

- A. '\ '                      B. '\080'                      C. '%d'                      D. 0xa

B 错在: 8 不能出现在八进制数中

28. 下列选项中用于判断 ch 是否是英文字母的表达式是\_\_C\_\_。

- A. ('a' <= ch <= 'z') || ('A' <= ch <= 'Z')  
B. (ch >= 'a' && ch <= 'z') && (ch >= 'A' && ch <= 'Z')  
C. (ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')  
D. ('a' <= ch <= 'z') && ('A' <= ch <= 'Z')

a<=b<=c 等价于  
(a<=b)<=c

29. 下列代码段的输出为 ( A )

```
#include <stdio.h>
void main()
{ int y;
double d=3.4, x;
x=(y=d/2.0)/2;
printf(" (%0.2f, %d)", x, y);
}
```

- A. (0.00, 1)                      B. (0.85, 1.7)                      C. (0.50, 1)                      D. none of above

30. 下列代码段执行结束后, 变量 s 的值为\_\_D\_\_。

```
int i=5, s=0;
while(i-->0) { if (i%2) s+=i; else s+=i--; }
```

- A. 15                      B. 11                      C. 10                      D. 6

i=5, s=0;    i=4;    s=4, i=3;    i=2;    s=4, i=1;  
i=0;    s=6, i=-1;

31. 若变量已正确定义并赋值, 符合 C 语言语法的表达式是\_\_B\_\_。

- A. a=2++                      B. a=3,5                      C. a=a+1=3                      D. 12.3%4

32. 表达式 5>3>1 值是\_\_A\_\_。

- A. 0                      B. 1                      C. 3                      D. 表达式语法错

5>3>1 → (5>3)>1 → 0>1 → 0

++不能作用在常量上。  
a+1 不是一个变量, 不能被赋值。12.3 是浮点数, 只有整数能取模操作。

33. 已知字母 A 的 ASCII 码为十进制数 65, 且 ch 为字符型变量, 则执行语句 ch='A'+'6'-'3'; 后, ch 中的值为\_\_A\_\_。

- A. D                      B. 68                      C. C                      D. 不确定

ch='A'+'6'-'3'='A'+3='D'

## 二. 填空题

34. 表达式  $(7 < 1 > 2^2)$  的值是 1。

$$7 < 1 > 2^2 = 14 > 2^2 = 3^2 = (011) \wedge (010) = 1$$

35. 表达式  $!!5$  的值是 1。

$$!!5 = !(5) = !0 = 1 \quad \text{单目运算符貌似只能右到左结合}$$

36. 写出一个条件表达式来计算  $x$  和  $y$  的最大值  $x > y ? x : y$ 。

37. 表达式  $7 * 7 / 9 * 4 \% 3$  的值为 2。

38. 表达式  $10 == 10 == 10$  的值为 0。

$$10 == 10 == 10 \rightarrow 1 == 10 \rightarrow 0 \quad \text{左到右结合}$$

39. 表达式  $1 < 10 < 5$  的值为 1。

$$1 < 10 < 5 \rightarrow 1 < 5 \rightarrow 1$$

40. 表达式  $\sim(10 < 1) \& 4$  的值为 0。

$$\sim(10 < 1) \& 4 \rightarrow \sim 20 \& 4 = \sim(0x14) \& 4 \rightarrow (0xFFEB) \& 4 = 0$$

41. 若有定义:  $\text{int } x=2, y=5$ ; 则表达式  $x * y - 3$  的值为 4。

42. 写出计算  $y$  的条件表达式  $y = x > 0 ? 1 : (x = 0 ? 0 : -1)$ 。

$$y = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

条件表达式中的括号不是必须的, 但是写上可以增加可读性。

43. 写出下列代码段的输出结果 1, 0.5。

```
float x1, x2;
x1 = 3/2;
x2 = x1/2;
printf("%d, %.1f", (int)x1, x2);
```

44. 写出下列程序段的输出结果 5, 10。

```
int k, x;
for (k=0, x=0; k<=9 && x!=10; k++)
 x+=2;
printf("%d, %d", k, x);
```

45. 下列代码段将会打印出 020。

```
printf("%#o\n", (010 < 1^1) & (0xe < 1));
```

46. 下列代码段的输出结果是 4, 6。

```
int k, s;
for (k=0, s=0; k<10; k++)
 if (s>k) break;
 else s += k;
printf("k=%d s=%d", k, s);
```

47. 下列代码段的输出为 10, 25。

```
int k, s;
for (k=1, s=0; k<10; k++){
 if (k%2==0) continue;
 s += k;
}
printf("k=%d s=%d", k, s);
```

$$\begin{aligned} & (010 < 1^1) \& (0xe < 1)) \\ & = (020^1) \& (\text{二进制 } 11100) \\ & = (\text{二进制 } 010\ 000^1000\ 001) \& (\text{二进制 } 11100) \\ & = (\text{二进制 } 010\ 001) \& (\text{二进制 } 11100) \\ & = \text{二进制 } 010\ 000 = \text{八进制 } 020 \end{aligned}$$

48. 下列代码段将会打印出\_\_5\_\_。

```
int a = 3, b;
if (a > 5)
 b = 7;
else if (a < 2)
 b = 6;
else
 b = 5;
printf("%d", b);
```

49. 下列代码段将会打印出\_\_4.0\_\_。

```
int a = 13;
int b = 3;
float x;
x = a / b;
printf("%.1f", x);
```

50. 下列代码段将会打印出\_\_1, 3#1, 3\_\_。

```
int j = 1, k = 2;
printf ("%d, %d#", j++, ++k);
printf ("%d, %d", --j, k--);
```

51. 下列代码段的输出为\_\_\*2\*#\*3\*#\_\_。

```
char x='0',y='1';
switch(!x) {
 case 0: printf("**0*#");break;
 case 1:
 switch(y){
 case '0': printf("**1*#");break;
 case '1': printf("**2*#");break;
 }
 default: printf("**3*#");
}
```

case 1: 没有对应的 break, 所以进入 case1 后还会执行 default 的 printf

52. 假定 a=0, b=3, c=0, d=26. 下列代码段中 d 的值为\_\_26\_\_。

```
if(!b) d=d-10;
else if(a)
if(!c) x=15;
else x=25;
```

53. 输入 **abcde#<回车>**, 下列代码段的输出结果是\_\_ac\_\_。

```
char c;
for(c=getchar(); getchar()!='#'; c=getchar())
 putchar(c);
```

三. 阅读理解题

54. 若输入 **-6+15\*3/5=<回车>**， 以下程序的输出结果是 **\_0#-6#9#6#1#**\_\_\_\_\_。

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

```
main()
```

```
{
```

```
 int m=0, sum=0;
```

```
 char c, oldc='+';
```

```
 do {
```

```
 c=getchar();
```

```
 if(c<='9'&& c>='0')
```

```
 m=10*m+c - '0';
```

读入操作数。 c 是构成操作数的数字字符

```
 else {
```

处理操作符

```
 if(oldc=='+')
```

'+'代表累加

```
 sum += m;
```

```
 else
```

其他的都代表减

```
 sum -= m;
```

```
 m=0;
```

```
 oldc=c;
```

更新操作数  
和操作符

```
 printf("%d#", sum);
```

输出当前的结果

```
 }
```

```
 } while(c!='=');
```

```
}
```

55. 下列程序的输出为 **\_1#2#3#**\_\_\_\_\_。

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

```
int f()
```

```
{ static int k;
```

静态变量如果没有给设初值，那么初始值为缺省的 0

```
 return ++k;
```

```
}
```

```
void main()
```

```
{ int k;
```

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

```
 printf("%d#", f());
```

```
}
```

56. 下列程序的输出为 **\_33#366#3699#**\_\_\_\_\_。

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

```
void main()
```

```
{ int k, x, s, t;
```

```
 x=3;
```

```
 s=0; t=x;
```

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

```
 t=t*10+x;
```

```
 s=s+t;
```

```
 printf("%d#", s);
```

```

 }
}

```

57. 若输入为 how are you? <ENTER>, 则输出为 How Are You\_\_\_\_\_。

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

```
void main()
```

```
{ int word;
```

```
char ch;
```

```
word=0;
```

```
while((ch=getchar())!='?'){
```

```
if(ch==' ') word=0;
```

```
else if(word==0){
```

```
word=1;
```

```
if(ch<='z'&&ch>='a') ch=ch-'a'+'A';
```

```
}
```

```
putchar(ch);
```

```
}
```

```
}
```

word 为 0 表示上一个单词处理结束  
word 为 1 表示新的单词开始了, 首字母变大写

注意字符? 不 putchar

58. 下列程序将会打印出 1 2 3# 2 3 1# 3 1 2#\_\_\_\_\_。

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

```
void main()
```

```
{ int j, k;
```

```
for (j=0; j<3; j++){
```

```
for (k=0; k<3; k++){
```

```
printf("%2d", (k+j+1)%3 ? (k+j+1)%3 : 3);
```

```
printf("#");
```

```
}
```

```
}
```

注意#号紧跟在数字之后

当(k+j+1)%3 为 0 时, 输出 3, 否则输出原值

printf("#");和 for(k=0...)是并列的语句

59. 下列程序的输出为 i=6, k=4\_\_\_\_\_。

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

```
void main()
```

```
{
```

```
int i,j,k=19;
```

```
while (i=k-1) {
```

```
k-=3;
```

```
if(k%5==0) { i++; continue; }
```

```
else if(k<5) break;
```

```
i++;
```

```
}
```

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

```
}
```

i=18, 15, 12, 9, 6, 3, 0

k=16, 13, 10, 7, 4, 1

## 主题作业二：数组与

答案：B

解释：（1）多维数组元素按照顺序放置，  
a[0][0]、a[0][1]、a[1][0]、a[1][1]、a[2][0]、a[2][1]  
（2）a[i]是指向&a[i][0]的指针。

### 一、单选题

- 数组定义为 `int a[3][2]={1, 2, 3, 4, 5, 6}`，数组元素 B 的值为 6。  
A、a[3][2]      B、a[2][1]      C、a[1][2]      D、a[2][3]
- 不正确的赋值或赋初值的方式是 C。  
A、`char str[]="string";`  
B、`char str[7]={'s','t','r','i','n','g'};`  
C、`char str[10]; str="string";`  
D、`char str[7]={'s','t','r','i','n','g','\0'};`
- 设变量定义为 `char s[]="hello\nworld\n"`，则数组 s 中有 B 个元素。  
A. 12      B. 13      C. 14      D. 15
- 下列代码段的输出为 A。  
`char c[]="\\t\\r\\0will\\n";`  
`printf("%d",strlen(c));`  
A. 4      B. 15      C. 16      D. 11
- 下列代码段输出字符串 str，则循环条件 `str[i]!='\0'` 能够被下列哪个选项替换 A。  
`char str[80]="hello, world";`  
`for (i = 0; str[i] != '\0'; i++)`  
`putchar(str[i]);`  
A. `str[i]`      B. `i < 80`      C. `!(str[i] == '\0')`      D. `i <= 80`
- 关于数组声明“`int a[4];`”，下列用法错误的是 D。  
A. `*a`      B. `a[0]`      C. `a`      D. `a++`
- 下列关于数组的定义，哪种用法是错误的 C。  
A. `int a[1][3];`  
B. `int x[2][2]={1,2,3,4};`  
C. `int x[2][]={1,2,4,6};`  
D. `int m[][3]={1,2,3,4,5,6};`
- 以下不正确的定义语句是 D。  
A. `double x[5]={2.0,4.0,6.0,8.0,10.0};`  
B. `char c2[]={'x10','xa','x8'};`  
C. `char c1[]={'1','2','3','4','5'};`  
D. `int y[5]={0,1,3,5,7,9};`
- 若变量已正确定义并且指针 p 已经指向某个变量 x，则 `(*p)++` 相当于 B。  
A、`p++`      B、`x++`      C、`*(p++)`      D、`&x++`
- 若 p1、p2 都是整型指针，p1 已经指向变量 x，要使 p2 也指向 x，A 是正确的。  
A、`p2=p1;`      B、`p2=**p1;`      C、`p2=&p1;`      D、`p2=*p1;`
- 下列程序段的输出是 B。  
`int c[]={1, 3, 5};`  
`int *k=c+1;`  
`printf("%d", *++k);`  
A、3      B、5      C、4      D、6
- 对于如下说明，语法和语义都正确的赋值是 C。  
`int c, *s, a[]={1, 3, 5};`  
A、`c=*s;`      B、`s[0]=a[0];`      C、`s=&a[1];`      D、`c=a;`
- 设变量定义为 `int a[2]={1,3}, p=&a[0]+1;`，则 \*p 的值是 B。  
A. 2      B. 3      C. 4      D. `&a[0]+1`

数组名是不能被修改的指针

字符串以'\0'为结束符

字符串的长度为 12，加上结束符'\0'一共存放了 13 个元素

数组名是不能被修改的指针

数组的定义只能省略第一个长度值（即变化最慢的维度）

数组初始化的元素个数不能大于其长度

对 p 所指向的变量进行加 1 操作

Int \*k=c+1 使得 k 指向 c+1。  
++k 后 k 指向 c+2。  
所以 \*++k 的值是 5。

A: 指针 s 的地址不可测，无意义  
B: 指针 s 的地址不可测，无意义  
D: 类型不一致，无法赋值

p 定义为指针，指向 a+1

14. 下列程序段的输出结果是\_\_B\_\_。

```
int *p, *q, k = 1, j=10;
p=&j; q = &k; p = q; (*p)++;
printf("%d", *q);
```

A. 1 B. 2 C. 10 D. 11

p=q 之后, p 和 q 都指向变量 k  
(\*p)++相当于 k++,  
所以(\*p)++之后 k 的值为 2

15. 根据声明int a[10], \*p=a;, 下列表达式错误的是\_\_D\_\_。

A. a[9] B. p[5] C. \*p++ D. a++

数组名是指针, 但是不能被修改

16. 下列表达式, 错误的是\_\_A\_\_。

A. char str[10]; str="string";  
B. char str[ ]="string";  
C. char \*p="string";  
D. char \*p; p="string";

数组名是指针, 但是不能被修改

17. 根据声明int a[10], p=&a[1];数组 a 的最后一个元素是\_\_B\_\_。

A. a[10] B. p[8] C. p[9] D. p[10]

18. 下列代码段将会打印出 ( C )

```
char s[]="student";
printf("%s%d", s+3, sizeof(s));
```

A. student7 B. dent7 C. dent8 D. student8

运算符 sizeof 作用在数组上时, 结果为数组的元素个数\*每个元素的大小, 即整个数组所占的 byte 数。数组定义时如果没有明确写出长度, 则其长度由初始化的元素个数决定。当用字符串初始化字符数组时, 还要计算结束字符 '\0'。

19. 下列语句定义pf为指向float类型变量f的指针, \_\_B\_\_是正确的。

A. float f, \*pf = f;  
B. float f, \*pf = &f;  
C. float \*pf = &f, f;  
D. float f, \*pf =0.0;

A 错在初始化 pf=f  
C 错在 f 定义在后  
D 错在 pf 初值不能为浮点数

20. 假设scanf语句执行时输入ABCDE<回车>, 能使puts(s)语句正确输出ABCDE字符串的程序段是\_\_D\_\_。

A. char s[5]="ABCDE";puts(s)  
B. char s[5]={'A','B','C','D','E'}; puts(s);  
C. char \*s;scanf("%s",s);puts(s);  
D. char \*s; s="ABCDE";puts(s);

A 错在初始化  
B 错在初始化不是字符串  
C 错在指针 s 的地址不可测, 无意义

## 二. 填空题

21. 根据声明 int a[2][3][4], 该数组元素数量为\_\_24\_\_。

数量为: 2\*3\*4

22. 下列代码段将会打印出\_\_16\_\_。

```
int array [] = {2, 8, 6, 11, 13, 2, 8, 2};
int k;
int total = 0;
for (k = 0; k < 8; k++) {
 if (array[k] % 2) break;
 total += array[k];
}
printf ("%d", total);
```

碰到奇数跳出循环

23. 下列代码段的输出为\_\_5, 20\_\_。

```
char st[20]="hello\0world!";
printf("%d,%d\n",strlen(st),sizeof(st));
```

该题数组大小为 20。参考第 18 题的解释。

24. 写出下列程序段的输出结果。

```
char str[]="hello\tworld\n";
printf("%d, %c\n", sizeof(str), *(str+10));
```

13, d  
(参考第 18 题的解释)

25. 下列代码段的输出为\_\_\_\_\_。

```
int c[]={10, 30, 5}, *pc;
for(pc=c; pc<c+2; pc++) printf("%d#", *pc);
```

输出为: 10#30#  
注意 c+2 指向元素 5

26. 下列代码段的输出为\_\_\_\_\_。

```
int a = 1;
int b = 2;
int *x, *y;
x = &a;
y = &b;
*x = *x + 1;
x = y;
*x = *x + 1;
printf("%d, %d", a, b);
```

输出为： 2, 3

27. 下列代码段的输出为\_\_\_\_\_。

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

a+1 指向元素 2  
a+6 指向元素 7  
循环累加了 2、4、6  
所以输出为： 12

28. 下列代码段

```
int i=0,s=0,a[10];
while((i<10)&&(a[i]>=0))
 s+=a[i++];
```

等价于代码段

```
int s,a[10],*p=a;
for(__s=0; p<a+10; s += *p++ __)
 if(*p<0) break;
```

a+1 指向元素 2  
a+6 指向元素 7  
输出为： 12

29. 函数 f 1 ( ) 等价于函数 f 2 ( )

```
void f1(char s[], char t[])
{
 int i=0;
 while (t[i]!='\0') {s[i]=t[i];i++;}
 s[i]=0;
}

void f2(char *s, char *t)
{
 while (__*s++ = *t++ __);
}
```

30. 下列程序段输出 ef\_\_\_\_\_。

```
char s[]="abc\0def\0ghi",*sp=s;
printf("%s",sp+5);
```

字符串以'\0'为结束符

三、 阅读理解题

31. 若输入 3 1 2 3 <回车> , 则以下程序的输出结果是\_\_\_\_\_。

```
#include "stdio.h"
void main()
{
 int i, j, n, a[10];
 scanf("%d", &n); // n = 3
 for (i=0; i<n; i++)
```



```

scanf("%d",&a[i]); //a[] = {1,2,3}
for (i=0; i<n; i++)
 for (j=0; j<n; j++)
 printf("%d, ", a[(i+j)%n]);
}

```

1, 2, 3, 2, 3, 1, 3, 1, 2,

32. 输入 123+234=357Yes?<回车>后，下列程序的输出结果是\_\_\_\_\_。

```

#include <stdio.h>
void main()
{
 int k, n1, n3[10];
 char c;
 n1=0;
 for(k=0; k<10; k++)
 n3[k]=0;
 while((c=getchar())!='\n'){
 switch(c){
 case'0': case '1': case '2': case '3': case '4':
 case'5': case '6': case '7': case '8': case '9':
 n3[c-'0']++;
 break;
 default:
 n1++;
 break;
 }
 }
 if(n1) printf("%d ", n1);
 for(k=0; k<10; k++)
 if(n3[k]) printf("%d ", n3[k]);
}

```

统计字符 0~9 出现的次数，放入数组 n3 中  
其他字符的出现次数累计在变量 n1 中  
所以 n3[] 中的值为  
{0, 1, 2, 3, 1, 1, 0, 1, 0, 0}  
n1 的值为 6

因此程序输出为：  
6123111

33. 若输入为 7 3 0 3 0 3 1 2 9 7 6 0<ENTER>，则输出为\_\_3# -1#\_\_\_\_\_。

```

#include <stdio.h>
void main()
{
 int j, k, sub, x;
 int a[5];

 for(j=1; j<=2; j++){
 for(k=0; k<5; k++)
 scanf("%d", &a[k]);
 scanf("%d", &x);
 sub=-1;
 for(k=0; k<5; k++)
 if(a[k]==x) sub=k;
 printf("%d#", sub);
 }
}

```

34. 下列程序的输出为\_\_\_\_\_。

```
#include <stdio.h>
void main()
{ int i, k,m;
 int a[8]={60,75,63,92,87,64,79,21}, s[10];

 for(k=0; k<10; k++)
 s[k]=0;
 for(i=0; i<8; i++){
 k=a[i]/10;
 s[k]++;
 }
 m=s[0];
 k=1;
 while (k<10){
 if(s[k]!=0) printf("%d#", s[k]);
 if (s[k]>m) m=s[k];
 k++;
 }
 printf("%d", m);
}
```

s[] 中的值为  
{0, 0, 1, 0, 0, 0, 3, 2, 1, 1}

因此程序输出为:  
1#3#2#1#1#3

35. 若输入 **AabD** <ENTER>, 则下列程序的输出为\_\_\_\_\_。

```
#include <stdio.h>
void main()
{
 char s[81];
 int i=0;
 gets(s);
 while (s[i]!='\0'){
 if(s[i]<='z'&& s[i]>='a')
 s[i]='z'+ 'a'-s[i];
 i++;
 }
 puts(s);
}
```

s[] 中的值为"AabD"  
while 语句把 s 中小写字母(a-z)对称颠倒  
因此程序输出为:  
AzyD

36. 下列程序运行时输入: <回车>, 输出结果为\_\_\_\_\_。

```
#include <stdio.h>
void f(int a[], int n);
main()
{
 int i, n, a[10];

 scanf("%d", &n); //n = 8
 for (i=0; i<n;i++)
```

```

scanf("%d",&a[i]); //a[] = 3 7 2 1 8 3 4 1
f(a+n/4,n/2); //f(a+2,4),把 2、1、8、3 排序为 1、2、3、8
for (i=0; i<n; i++)
 printf("%d#",a[i]);
}

void f(int a[], int n) //从小到大排序
{
 int i, j, temp;

 for(i = 1; i < n; i++)
 for(j = 0; j < n-i; j++)
 if (a[j] > a[j+1]) { temp = a[j]; a[j] = a[j+1]; a[j+1] = temp; }
}

```

因此程序输出为：  
3#7#1#2#3#8#4#1#

37. 下列程序的输出为\_\_\_\_\_。

```

#include <stdio.h>
int z;
void p(int *x,int y)
{
 ++ *x;
 y ++;
 z = *x+y;
 printf("%d,%d,%d#",*x, y, z);
}
void main()
{
 int x=1, y=2, z=3;
 p(&y, x);
 printf("%d,%d,%d#",x, y, z);
}

```

程序输出为：  
3, 2, 5#1, 3, 3#

38. 下列代码段的输出为\_42\_\_\_\_\_。

```

#include <stdio.h>
int f(int a[], int n, int m)
{
 int i, j, sum, max=0;
 for(j=0; j<m; j++) max+=a[j];
 for (i=1; i<n; i++) {
 sum=0;
 for (j=0; j<m; j++)
 sum+= *(a+i*m+j);
 if (sum>max) max=sum;
 }
 return max;
}
void main()
{

```

f 函数把 a 看作是[n][m]的数组  
计算并返回最大的行和

```

 static int a[]={1,2,3,4,5,6,7,8,9,10,11,12};
 printf("%d", f(a, 3, sizeof(a)/sizeof(int)/3)); //f(a,3,4)
 }

```

39. 下列程序运行时输入: **C Java C++ VB Fortran Basic** <回车>, 输出结果为 Fortran。

```

#include <stdio.h>
void find(char *str)
{
 char *p=str;
 int k1=0,k2=0;

 while(1) {
 if(*p!=' ' && *p!='\t' && *p!='\0') k2++; //单词以空格，制表符，'\0'为分隔符
 else {
 if (k2>k1) { k1=k2; str=p-k2;} //找到一个更长的单词
 k2=0;
 if (*p=='\0') break;
 }
 p++;
 }
 for(k2=0;k2<k1;k2++) printf("%c",*str++); //输出该最长的单词
 return;
}

main()
{
 char a[80];
 gets(a);
 find(a);
}

```

40. 下列代码段的输出为 0, 4。

```

#include <stdio.h>
void p(int *x,int y)
{
 ++ *x;
 y=y+2;
}
void main()
{
 int x=0, y=3;
 p(&y, y);
 printf("%d, %d", x, y);
}

```

## 主题作业三：函数与结构

### 一. 单选题

1. 针对如下定义，合法的表达式是\_\_C\_\_。

```
struct node {
 char s[10];
 int k;
}
```

p[5];

- A. p.k=2                      B. p[0]->k=2                      C. (p->s)[0]='a'                      D. p[0].s="a"

2. 针对如下定义：

```
static struct {
 int x, y[3];
} a[3] = {{1,2,3,4},{5,6,7,8},{9,10,11,12}}, *p;
p = a+1;
```

表达式\*((int\*)(p+1)+2)的值为\_\_D\_\_。

- A. 3                      B. 7                      C. 10                      D. 11

3. 对于以下结构定义，++p->str 中的++加在\_\_A\_\_。

```
struct {
 int len;
 char *str;
} *p;
```

- A、指针 str 上                      B、指针 p 上                      C、str 指的内容上                      D、以上均不是

4. 根据声明 int (\*p)[10], p 是一个\_\_A\_\_。

- A. 指针                      B. 数组                      C. 函数                      D. 数组元素

5. 若下面程序中所有的变量均已声明或定义，则下列选项中的变量能够在 fun()中使用的  
是\_\_A\_\_。

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

 return;
}
int z;
void main()
{
 int a,b;
 fun(a);

}
```

- A. x, y                      B. x, y, z                      C. a,b,y,z                      D. a,b,x,y,z

### 二. 填空题

6. 对于下面的定义，(s[0].b)/(++p)->a 的值为\_\_0\_\_。

数组变量名也是指针

访问结构指针所指向的结构的成员变量时用->，访问结构的成员变量时用。

成员 s 是数组，不能直接赋值（但是初始化数组的时候可以，例如 char s[10]="abc"；那是由编译器在编译阶段完成的）

p 指向 a[1]；然后 p+1 指向 a[2]，即元素 9 的位置；  
然后(int\*)(p+1)则是将指针强制转换为一个整数指针，  
位置不动，还是 9 的位置；  
+2 操作则把上面的整数指针向后移动 2 个，即指向 11 的位置；

->的优先级高于++，该表达式等价于++(p->str)  
所以，++作用于 p 的成员变量 str 上。  
执行后的结果是：指针 p 不变，但是它指向的结构的成员 str 加 1

**第 4 题解答：**注意：int (\*p)[10] 和 int \*p[10]是完全不一样的  
int \*p[10]定义了一个数组，其元素类型为 int\*；此时 sizeof(p)为 40。  
int (\*p)[10]定义了一个指针，其类型是长度为 10 的整型数组；此时 sizeof(p)为 4。

变量遵循先声明再引用的原则

x 作为函数 fun 的形式参数变量，已经声明为 int 型，可以使用  
y 是函数 fun 的局部静态变量，也已经声明，也可以使用  
z 在 fun 之后定义，之前没有声明，不能使用  
a,b 则是函数 main 的局部变量，fun 不能使用。

**注意：**这里的先后顺序只得是编译器编译语句的顺序，不是语句运行的先后顺序。  
因为 if/switch 分支、for/while 循环等流程控制语句可能动态的控制程序语句的执行次序，所以语句的执行次序不是固定的。

```
struct {
 int a;
 int b;
} s[2]={2,4,6,8}, *p=s;
```

s[0] 为{2, 4}, s[1] 为{6, 8}。  
s[0].b 为 4  
p 初始指向 s[0], ++p 指向 s[1], 故 (++p) -> a 的值为 6  
因为 4/6 为 0, 所以。。。

7. 下列程序段执行后, z 的值是 8。

```
static struct {
 int x, y[3];
} a[3] = {{1,2,3,4},{5,6,7,8},{9,10,11,12}}, *p=a+3;
int z;
z=((int*)(p-1)-1);
```

结构数组 a[3]在内存中也可以看做是一个长度为 12 的整数数组: 1, 2, 3, ..., 11, 12。  
p 的初始地址为 a+3, 那么 p-1 指向 a[2], 即元素 9 的位置  
(int\*) 将 p-1 强制转换为一个整型指针, 但是还是指向 9 的位置。所以-1 作用在整型指针(int\*)(p-1)上, 得到 9 的前面一个整数元素的地址, 即 8 的地址。

8. 下列代码段将会打印出 COND。

```
char *c[3]={"FIRST", "SECOND", "THIRD"};
printf("%s", *(c+1)+2);
```

c 是一个指针数组, c[0]指向字符串"FIRST", c[1]指向字符串"SECOND", c[2]指向字符串"THIRD"。  
(c+1)为 c[1], 所以 \*(c+1)+2 指向字符 SECOND 中的 C。

参考第 8 题的解释

9. 下列程序段的输出结果是 fgh。

```
char *st[]={"abcd", "efgh", "ijkl", "mnop"};
printf("%s", *(st+1)+1);
```

参考第 8 题的解释。\*(st+2)等价于 st[2]。

10. 下列代码段的输出为 FOUR, P。

```
char *st[]={"ONE", "TWO", "FOUR", "K"};
printf("%s, %c\n", *(st+2), **st+1);
```

因为\*优先级高于+, 所以\*\*st+1 等价于 st[0][0]+1 即, 'O'+1, 即'P'

11. 下列代码段的输出 45,3。

```
char *a[]={"678", "45"}, **p=a+1;
printf("%s,%c", *p, **p-1);
```

P 指向 a+1, 所以\*p 为 a[1]。 \*\*p 为 a[1]所指向的字符串的第一个字符, 即'4'。所以\*\*p-1 为'3'

12. 对于数组 a, \*(a[1]+1)的值为 4。

```
int a[3][2]={1,2,3,4,5,6};
```

a[0]指向第 0 行。a[1]指向第 1 行, 即元素 3 的位置。  
那么 a[1]+1 指向元素 3 的后面一个位置, 即 4 的位置。

13. 对于以下递归函数 f, 调用 f(3) 的返回值是 -17。

```
f(int n)
{ return ((n>0) ? 2*f(n-1)+f(n-2) : -1); }
```

|       |      |      |      |      |
|-------|------|------|------|------|
| f(-1) | f(0) | f(1) | f(2) | f(3) |
| -1    | -1   | -3   | -7   | -17  |

14. 下列代码段将会打印出 4。

```
int i;
int f(int x)
{ static int k = 0;
 x+=k++;
 return x;
}
i=f(2);
i=f(3);
printf("%d",i);
```

初始化: k 为 0,  
调用 f(2): k 从 0 变到 1, 函数返回 2+0=2  
调用 f(3): k 从 1 变到 2, 函数返回 3+1=4

15. 下列代码段将会打印出 18。

```
int f(int x)
{ return ((x>0)? x*f(x-1):3); }
printf("%d",f(f(1)));
```

|      |      |      |      |
|------|------|------|------|
| f(0) | f(1) | f(2) | f(3) |
| 3    | 3    | 6    | 18   |

f(f(1)) = f(3) = 18

16. 用 typedef 写出类型定义 typedef int \*AIP[10];, 使得 AIP 表示含有 10 个元素的整型指针数组类型。

typedef TOld TNew[10];  
定义了一个新的类型, 名称为 TNew, 它是一个长度为 10 的数组, 其中每一个元素类型为 TOld。对于该题, Told 为 int\*, TNew 为 AIP。

而 typedef TOld (\*PNew)[10]; 则定义了一个新的类型的指针。该语句等价于:  
typedef TOld TNew[10];  
typedef TNew\* PNew;

17. 假定可执行程序文件名字为 prog，则运行命令：prog hello world 后的输出为\_\_\_\_\_。

```
#include <stdio.h>
int main(int argc, char *argv[]) {
 printf("%s#%s#", argv[0], argv[argc - 1]);
 return 0;
}
```

输出为：prog#world#

argc 为 3，

argv[0],[1],[2]分别为 “prog”, “hello”和”world”

18. 下列程序段的输出是\_\_\_\_\_ -3 \_\_\_\_\_。

```
#define FB(a,b) (a*b+1)
int k=3;
k= FB(k+1,k-1)-9;
printf("%d",k);
```

展开宏 FB(k+1,k-1)后得到：

(k+1\*k-1+1)

所以该语句为：k= (k+1\*k-1+1) -9 = -3

### 三、 阅读理解题

19. 下列程序的输出为\_6#18#\_\_\_\_\_。

```
#include <stdio.h>
main()
{
 int s=0,i,a[3][3]={1,2,3,4,5,6,7,8,9};
 for(i=0; i<3; i++)
 s+= *(*a+i);
 printf("%d#", s);
 for(i=0; i<3; i++)
 s+= **(a+i);
 printf("%d#", s);
}
```

s += a[0][i]

s += a[i][0]

a[3][3]的元素排列为：

1 2 3

4 5 6

7 8 9

\*(a+i) 等价于\*(a[0]+i)等价于 a[0][i]。

所以第一个 for 循环后，s 为 1+2+3=6

\*\* (a+i)等价于\*a[i]等价于 a[i][0];

所以第二个 for 循环后，s 为 6+1+4+7=18

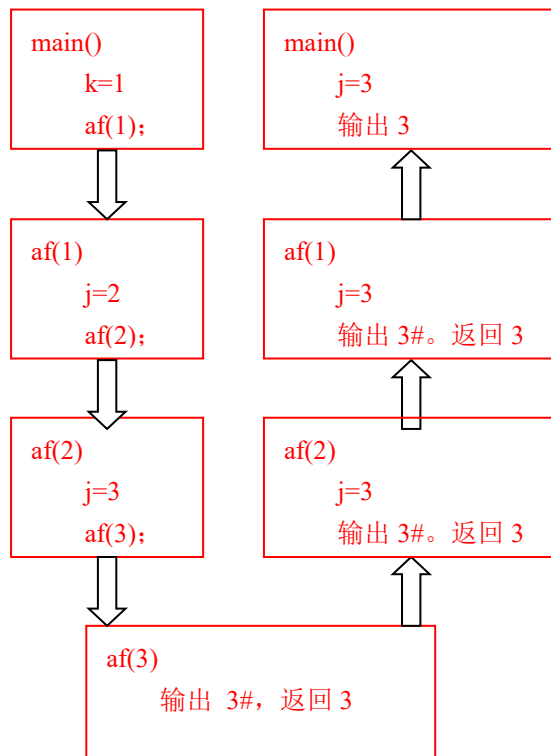
20. 下列程序的输出是\_\_5\_\_。

```
int f (int x)
{
 if(x<=1) return 1;
 else return f(x-1)+f(x-2);
}
void main ()
{
 printf("%d", f(4));
}
```

| f(0) | f(1) | f(2) | f(3) | f(4) | f(5) |
|------|------|------|------|------|------|
| 1    | 1    | 2    | 3    | 5    | 7    |

21. 下列程序的输出是\_\_3#3#3#3\_\_。

```
#include <stdio.h>
int main (void)
{
 int k = 1;
 int a_function (int j);
 k = a_function (k);
 printf ("%d" , k);
}
int a_function (int j)
{
 if (j < 3) {
```



```

 j++;
 j = a_function (j);
 }
 printf ("%d#" , j);
 return (j);
}

```

22. 下列程序的输出为 a = 1, b = 4 。

```

#include <stdio.h>
void melon (int g, int * h);
int main (void)
{
 int a = 1, b = 2;
 melon (a, &b);
 printf ("a = %d, b = %d", a, b);
}

void melon (int b, int * c)
{
 b++;
 *c = *c + b;
}

```

melon(a, &b)执行后, a 的值不变, b 的值增加了 a+1, 即 b 变为 4。

23. 下列程序的输出为 225 。

```

#include <stdio.h>
#include<math.h>
int prime(int n)
{
 int i,m;
 if(n==1) return 0;
 m=sqrt(n);
 for(i=2;i<=m;i++)
 if(n%i==0) break;
 return i>m;
}

void main()
{
 int num,i;
 num=20;
 for(i=2;i<=num;i++){
 while(prime(i)&&(num%i==0)){
 printf("%d ",i);
 num/=i;
 }
 }
}

```

Prime(n) 判断 n 是否为素数, 是返回 1, 否则返回 0

计算并输出 num 所有的素数因子  
 num =20, i=2, print 2  
 num =10, i=2, print 2  
 num =5, i=2 跳出 while  
 num =5, i=3,  
 num =5, i=4,  
 num =5, i=5, print 5  
 num =1, i=5, 跳出循环

24. 下列程序的输出为 1, 2 。

```

#include <stdio.h>
void f(int *x,int *y)
{
 int *p;
 p=x; x=y; y=p;
}

```

函数 f 里交换指针, 但是交换的结果并不能返回给调用者, 也就是说, 在调用者看来, 都是无用功, 等于啥也没做。



```

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

```

25. 若输入 **this is a test.<ENTER>**，则下列程序的输出 **This Is A Test.**。

```

#include <stdio.h>
#define TRUE 1
#define FALSE 0
int change(char *c,int status);
void main()
{
 int flag=TRUE;
 char ch;
 do{
 ch=getchar();
 flag=change(&ch,flag);
 putchar(ch);
 } while(ch!='. ');
 printf("\n");
}

```

```

int change(char *c,int status)
{
 if(*c==' ') return TRUE;
 if(status&&*c<='z'&&*c>='a') *c+= 'A'-'a';
 return FALSE;
}

```

Status 的意义是：是否要求变成大写字母

碰到空格了，后面的第一个字符要求大写  
也就是单词的第一个字母大写

26. 下列程序的输出为 **5#0#**。

```

#include <stdio.h>
int f(int m)
{
 static int k=0;
 int s=0;
 for(; k<=m; k++) s++;
 return s;
}
void main()
{
 int s=1;
 s=f(4);
 printf("%d#%d#", s, f(2));
}

```

执行 f(4)时，m=4。执行后，static int k 变成了 k=5，函数返回 5。  
所以 main 中 s=5。  
再执行 f(2)时，m=2，k=5，所以 f 返回 0。  
所以输出为 5#0#

27. 下列程序的输出为 **-1#-1#-2#**。

```

#include <stdio.h>

```

```

int fun(int x)
{
 int t;
 if(x<=0)
 t=x;
 else
 t=fun(x-1)+fun(x-2);
 return t;
}

void main()
{
 int i;
 for(i=1;i<=3;i++)
 printf("%d#", fun(i));
}

```

|        |       |      |      |      |
|--------|-------|------|------|------|
| f(-1), | f(0), | f(1) | f(2) | f(3) |
| -1     | 0     | -1   | -1   | -2   |

28. 下列程序将会打印出 9#1# 。

```

#include <stdio.h>
#define my_square_add(a, b) (a * a + b * b)
#define my_square_sub(a, b) (a * a - b * b)
int main()
{
 int a = 1, b = 2;
 printf("%d#", my_square_add(a + b, b));
 printf("%d#", my_square_sub(a + b, b));
 return 0;
}

```

宏定义展开

msadd(a+b,b) → a+b\*a+b+b\*b = 9

mssub(a+b,b) → a+b\*a+b-b\*b = 1

29. 下列程序的输出结果是 1#3#5# 。

```

#include <stdio.h>
int f(int x)
{
 static int k=0;
 return ++k+x;
}

main()
{
 int k;
 for(k=0;k<3;k++) printf("%d#", f(k));
}

```

执行 f(0)后: static k=1, 返回 1+0=1

执行 f(1)后: static k=2, 返回 2+1=3

执行 f(2)后: static k=3, 返回 3+2=5

30. 下列命令行参数程序生成的执行程序为 test.exe, 执行 **test 123 abc**<回车>, 输出结果是 abc#123# 。

```

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

```

argc 为 3

argv[0]为 test

argv[1]为 123

argv[2]为 abc

while 的条件表达式为 --argc, 所以当 argc 为 3、2 时, 该条件成立。条件成立时, 分别输出 argv[2] 和 argv[1]。因此答案为。。。

31. 假设有下列函数定义：

```
void foo(int sum)
{
 int j;
 for (j = 0; j < 10; ++j)
 sum += (j + 1) * sum;
}
```

在main函数中有如下代码段：

```
int sum = 0;
foo(sum);
printf("%d", sum);
```

执行后，输出结果为\_\_0\_\_。

Main 函数调用 foo(sum)之后，sum 的值不会被 foo 函数改变。

## 主题作业四：指针进阶与文件

### 一. 选择题

1. 下列声明有效的是 ( C )

```
struct node {
 char s[10];
 int k;
} p[5];
```

A. p.k=2      B. p[0]->k=2      C. (p->s)[0]='a'      D. p[0].s="a"

2. static struct {

```
 int x, y[3];
```

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

```
 p = a+1;
```

表达式\*((int\*)(p+1)+2)的值为 ( D )

A. 3      B. 7      C. 10      D. 11

1~4 题的解释  
参 考  
homework-3

### 二. 填空题

3. 对于下面的定义, (s[0].b)/(++p)->a 的值为 ( 0 )

```
struct {
 int a;
 int b;
} s[2]={2,4,6,8}, *p=s;
```

4. 下列程序段执行后, z 的值是 8。

```
static struct {
 int x, y[3];
} a[3] = {{1,2,3,4},{5,6,7,8},{9,10,11,12}}, *p=a+3;
int z;
z=((int*)(p-1)-1);
```

5. 已知函数原型 void print(FILE \*input, const char \*output);其中, input 为输入流, output 为输出文件名, 你需要打印标准输入流 (stdin) 到文件"temp.txt", 请写下处理这项任务的语句 (

print(stdin, "temp.txt"); )

stdin 等是 FILE \*类型, 属于标准 I/O, 高级的输入输出函数

### 三. 程序阅读题

6. 若下列数据存放在文本文件 a.txt 里

```
one?two?1234?output?
```

下列数据存放在文本文件 b.txt 里

```
one?two?1204?input?
```

则下列程序的输出为 ( 3#0#2# )

```
include <stdio.h>
include <stdlib.h>
void main()
{ int count;
 char ch1,ch2;
```

```

FILE *f1, *f2;

if ((f1 = fopen("a.txt", "r")) == NULL){
 printf("Can't open file : %s\n", "a.txt");
 exit(0);
}
if ((f2 = fopen("b.txt", "r")) == NULL){
 printf("Can't open file : %s\n", "b.txt");
 exit(0);
}
count=0;
while (!feof(f1)||!feof(f2)){
 ch1=fgetc(f1);
 ch2=fgetc(f2);
 if(ch1!=ch2){
 printf("%c#%c#", ch1, ch2);
 printf("%d#", count);
 break;
 }
 if(ch1=='?') count++;
}
fclose(f1);
fclose(f2);
}

```

发现不同的字符，输出信息

发现一个不同的，终止

7. 文本文件 **in.txt** 包含下列三行数据，

*This is a demo*

*hello world*

*C is fun*

在下面这个命令行程序 find.c 中，在运行 “find -n is” 后程序的输出为 ( )

```

#include <stdio.h>
void main(int argc, char *argv[])
{
 char line[1000];
 int lineno=0;
 FILE *fp;
 if((fp=fopen("in.txt", "r")) == NULL) {
 printf("File in.txt Open Error\n");
 exit(1);
 }
 if(argc==2) {
 while(fgets(line, 1000, fp)!=NULL)
 if (strstr(line, argv[1]) != NULL) {
 printf("%s", line);
 }
 }
}

```

程序的输出为：

1: This is a demo  
3: C is fun

注意输出有两行，因为在文件中 demo 后面有换行符\n

读入第一行时，line 中的内容是  
"This is a demo\n"

```

 }else if(argc==3 && strcmp(argv[1],"-n")==0){
 while(fgets(line,1000,fp)!=NULL) {
 lineno++;
 if (strstr(line, argv[2]) != NULL)
 printf("%d: %s",lineno,line);
 }
 }else
 printf("Usage: find [-n] pattern\n");
}

```

#### 四. 程序（段）填空题

1. 下面的程序通过命令行参数方式比较两个文本文件是否完全相同，比较方式 **prog file1 file2**。

```

#include <stdio.h>
main(int argc, char *argv[])
{
 FILE *fp1, *fp2;
 char c1, c2;

 if (argc!= (8)) {
 printf("Usage: prog file1 file2\n");
 exit(0);
 }
 if ((fp1=fopen(argv[1], "r")) == NULL || (9)) {
 printf("Cannot open files\n");
 exit(0);
 }
 c1=fgetc(fp1);
 c2=fgetc(fp2);
 while ((10) && c1==c2) {
 c1=fgetc(fp1);
 c2=fgetc(fp2);
 }
 if ((11))
 printf("The two files are different!\n");
 else
 printf("The two files are the same!\n");
 (12)
 fclose(fp2);
 return 0;
}

```

3

argc 等于命令参数个数+1

(fp2=fopen(argv[2], "r")) == NULL

!feof(fp1) && !feof(fp2)

c1!=c2 || || !feof(fp1) || !feof(fp2)

fclose(fp1);