

第三章习题参考答案

一、单项选择题

1.D 2.B 3.A 4.C 5.C 6.D 7.B 8.D 9.B 10.C 11.A 12.D 13.C 14.C 15.C
16.C 17.B 18.C 19.C 20.D 21.A 22.D 23.D 24.D A 25.D
26.A 27.B

二、填空题

28. 补码 29. 10-308~10308 15~16 位 30. 0 或 1 31. 非 从右向左
32. 函数调用 33. 题目错误, 该表达式的值不可能为 1。 34. 1 35. 65, 89

三、应用题

36. 10 9

37. 11 38. 此题与 33 题内容基本相同, 只能回答 0。

0

0

12

1

39. 区别是: 求负: 单目, 右结合; 减法: 双目, 左结合

取址: 单目, 右结合; 按位与: 双目, 左结合

第四章习题参考答案

一、单项选择题

1.D 2.C 3.D 4.A 5. D 6.B 7.A 8.C 9.B 10.B

二、填空题

11. 一条 ;

12. 5.169000

13. (1) -200 2500 (2) i=-200, j=2500 (3) i=-200 j=2500

14. a=98, b=765.000000, c=4321.000000

15. 三种可能的输入是:

(1) 100<回车> 25.81<回车> 1.89234<回车>

(2) 100<空格> 25.81<空格> 1.89234<回车>

(3) 100<TAB> 25.81<TAB> 1.89234<回车>

16. 0, 0, 3

17. 3; 18. 输入语句的格式是: scanf("%lf%lf%lf");

19. 13 13.000000 13.000000

20. 两种方法: (1) 增加一个临时变量 t。t=a; a=c; c=t; (2) a=a+c; c=a-c; a=a-c;

三、编程题

4. 21

main()

```
{ printf("*****\n");  
  printf("very good! \n");  
  printf("*****\n");  
}
```

4. 22

main()

```
{ int hour,min;
```

```

scanf("%d,%d",&hour,&min);
min=60*hour+min;
printf("%d\n",min);
}

```

4.23

```

main( )
{ float c , f;
  scanf("%f",&c);
  f=32+9.0*c/5;
  printf("%f\n",f);
}

```

按题目要求分别输入-10, 0, 15 和 34 计算相应的华氏温度。

4.24

```

main()
{ float r,l,s;
  scanf("%f",&r);
  l=2*3.141592*r;
  s=3.14159*r*r;
  printf("r=%f, l=%f, s=%f\n",r,l,s);
}

```

4.25

```

#include<math.h>
main( )
{ float a,b,c,s,area;
  scanf("%f,%f,%f",&a,&b,&c);
  if((a+b>c)|| (a+c>b)|| (b+c>a)) /* 判断两边之和大于第三边 */
  { s=(a+b+c)/2;
    area=sqrt(s*(s-a)*(s-b)*(s-c)); /*调用开方函数, 计算面积 */
    printf("area=%f\n",area);
  }
}

```

4.26

```

main( )
{ int a=3,b=4,c=5;
  float d=1.2,e=2.23,f=-43.56;
  printf("a=%3d,b=%-4d,c=**%d\n",a,b,c);
  printf("d=%3.1f\ne=%6.2f\nf=%-10.4f**",d,e,f);
}

```

4.27

```

main( )
{ int a,b,c,t ;

```

```

scanf ("%d,%d,%d", &a, &b, &c);
printf ("a=%d, b=%d, c=%d\n", a, b, c);
t=a;a=b;b=c;c=t;
printf ("a=%d, b=%d, c=%d\n", a, b, c);
}

```

4.28

```

main( )
{ float a, b, c, aver;
  printf("Please input three numbers:\n");
  scanf ("%f,%f,%f", &a, &b, &c);
  aver=(a+b+c)/3;
  printf("average of %5.2f,%5.2f and %5.2f is %5.2f", a, b, c, aver);
}

```

4.29

```

main( )
{ int a, b, c, x, y;
  scanf ("%d %d %d", &a, &b, &c);
  x=a*b;y=x*c;
  printf ("a=%d, b=%d, c=%d\n", a, b, c);
  printf ("x=%d, y=%d\n", x, y);
}

```

第五章参考答案

一、单项选择题

1D 2.C 3.B 4.B 5.D 6.A 7.B 8.D

二、填空题

9. 非0 0; 10. k == 0

11. if (x>4 || x<-4) printf("%d",x); else printf("error!");
或: (x>4 || x<-4) ? printf("%d",x) : printf("error!");

12. if ((x%2!=0) && ((x>10 && x<10) || (x>200 && x<210)))

13. 4; 14. 10! Wrong ! 11

15. \$\$\$a=0; 16 2, 1; 17 a>0&&a<b

三、编程题

5-18

```

main( )
{ int a, b, c;
  scanf ("%d,%d,%d", &a, &b, &c);
  if ( ( a>b)&&(a>c) )
    if (b<c) printf("min=%d\n",b);
    else printf("min=%d\n",c);
  else if ( (a<b)&&(a<c) ) printf("min=%d\n",a);
  else printf("min=%d\n",b<c?b:c);
}

```

```
}
```

5-19: 本程序使用 goto 语句, 仅供参考, 请勿模仿。

```
main( )
{
    long a;
    int k;
    scanf ("%ld",&a);
    printf ("%ld\n",a);
loop: if ( a != 0 )
    {
        k = a%10;
        printf ("%d",k);
        a = a/10;
        goto loop;
    }
}
```

5-20

(1) 用简单的 if 结构

```
main( )
{
    int x, y;
    scanf ("%d",&x);
    if ( x<0 && x>-5 )      y=x;
    if ( x<5 && x=0 )      y=x-1;
    if ( x>=5 && x<10 )    y=x+1;
    printf ("%d,%d\n", x, y);
}
```

(2) 用嵌套的 if 结构

```
main( )
{
    int x, y;
    printf("Please input x : -5-----10\n");
    scanf("%d",&x);
    if(x<5)
        if(x<0)
        {
            if(x>-5) y=x;
            else y=x-1;
        }
        else
        if(x<10) y=x+1;
    printf("%d,%d\n", x, y);
}
```

(3) 用 if-else 分支结构

```
main( )
{
    int x, y;
```

```

scanf ("%d",&x);
if ( x<0 && x>-5 )      y=x;
else if ( x<5 && x>=0 )  y=x-1;
else if ( x>=5 && x<10 ) y=x+1;
printf ("%d,%d\n",x,y);
}

```

(4) 用 switch()结构

```

main( )
{ int x, y, k;
  printf ("Please input x : -5-----10\n");
  scanf ("%d",&x);
  if ( x<0 ) k=-1;
  else      k=x/5;
  switch (k)
  { case 1: y=x+1; break;
    case 0: y=x-1; break;
    case -1: y=x; break;
  }
  printf ("%d,%d\n", x, y);
}

```

5-21

```

main( )
{ int y0, m0, d0, y1, m1, d1, age;
  printf("Please input current day:\n");
  scanf("%d,%d,%d",&y1,&m1,&d1);
  printf("Please input birthday:\n");
  scanf("%d,%d,%d",&y0,&m0,&d0);
  age = y1-y0;
  if ( m1<m0 )      age--;
  else if (m1==m0&&d1<d0) age--;
  printf("age=%d\n",age);
}

```

5-22

```

main( )
{ int a, b, c;
  printf("Please input a,b,c:\n");
  scanf ("%d,%d,%d",&a,&b,&c); /* 输入三角形的三条边 */
  if (a+b>c&&a+c>b&&b+c>a) /* 判断两边之和大于第三边 */
  { if (a==b&&b==c) /* 判断等边三角形 */
    printf(" 等边三角形\n");
    else if (a==b||a==c||b==c) /* 判断等腰三角形 */

```

```

printf("等腰三角形\n");
else if(a*a+b*b=c*c||a*a+c*c=b*b||b*b+c*c=a*a) /* 判断直角三角形 */
printf("直角三角形\n");
else printf("一般三角形\n");
else printf("不能组成三角形\n");
}

```

5-23

```

main( )
{ int x, y, k;
scanf ("%d", &x);
k = x/10;
switch ( k )
{ case 1: y=1; break;
case 2: y=2; break;
case 3: y=3; break;
case 4: y=4; break;
case 5: y=5; break;
default: y=6; break;
}
printf("x=%d,y=%d\n", x, y);
}

```

5.24

```

main( )
{ int m, k;
float n;
printf ("Please input 购买金额 m :\n");
scanf ("%d",&m);
k = m/100;
switch (k)
{ case 0: n=0; break;
case 1: n=0.015; break;
case 2: n=0.025; break;
case 3: n=0.035; break;
case 4: n=0.045; break;
case 5: n=0.055; break;
default: n=0.06; break;
}
printf("%5d,%7.2f,%7.2f", m, m-n*m, m*n);/*输出购买金额、实际支付金额和返还金额 */
}

```

5-25: 本程序使用 goto 语句实现, 仅供参考, 请勿模仿。

```

main( )

```

```

{ int year=1995, month, days=0;
loop: if ( year <= 2000 )
{
    days = days+7*31+4*30+28;           /* 每年增加 365 天 */
    if ( year%4==0&&year%100!=0 || year%400==0 ) /* 判断闰年 */
        days++;
    year++;
    goto loop;
}
days = days-21;                       /* 仅计算到 2000 年 12 月 10 日 */
printf("The total days is :%d\n", days);
}

```

5-26: 本程序使用 goto 语句实现，仅供参考，请勿模仿。

```

main ( )
{ long a;
  int b, k=0, m;
  printf ("Please input a<99999:\n");
  scanf ("%ld",&a);
  b = a%10;
  if ( a/10==0 )
  { k=1;
    printf("%ld\n",a);
  }
  else
  { printf("%d\n",b);
    m=a;
    loop: if ( m!=0 )
        { k++;
          m=m/10;
          goto loop;
        }
  }
  printf ("%ld,%d\n", a,k);
}

```

5-27: 本程序使用 goto 语句实现，仅供参考，请勿模仿。

```

#include<math.h>
main( )
{ long a, m, n=0;
  int i=0, k;
  scanf("%ld",&a);
  m=a;
  loop: if ( m != 0 )

```

```

        {   k = m%10;
            k = k+2;
            if ( k >= 9 )
                k=k%10;
            m = m/10;
            n = k*pow(10,i)+n;
            i++;
            goto loop;
        }
    printf ("%ld,%ld\n", a, n);
}

```

第六章习题参考答案

一、单选题

1. B 2. C 3. C 4. B 5. C 6. B 7. C 8. A

二、填空题

9. 无穷
10. 7 4
11. 外层循环 5 次；对于每次外层循环，内层循环 4 次。
12. 11； 13. 1.600000
14. ****#\$
15. 8 5 2
16. ① float d=1.0 ② k++ ③ k<=n
17. ① x>=0 ② x<amin

三、编程题

18.

```

main( )
{   int i, k, s=0;
    for ( i=1,k=1; i<=101;i +=2,k*=-1 )
        s = s+i*k;
    printf ("%d\n", s);
}

```

19.

```

main( )
{   int i, k, p=0, n=0, s=0;
    for ( i=1; i<=10; i++ )
    {   printf ("i=%d: ", i);
        scanf ("%d",&k);
        s += k;
        if ( k>0 ) p += k;
        if ( k<0 ) n += k;
    }
    printf ("s=%d,p=%d,n=%d\n", s, p, n);
}

```

20.


```

main( )
{
    int i;
    printf("Please Enter Number: ");
    scanf("%d",&i);
    while ( i > 0 )
    {
        printf ("%d,",i%10);
        i /= 10;
    }
}

```

21.

(1) 注释：此题的关键是要保证乘积不超过 int 型允许的最大数。由于仅要求结果的最后三位，被乘数的后三位参加运算；则将乘数后三位的每一位分别和被乘数相乘，然后部分积再相加。

```

main( )
{
    int i, a, b, c1, c2, c3, y, y1, y2, y3;
    printf ("Please Enter a,b:");
    scanf ("%d,%d",&a,&b);
    y = a;
    c1 = a%10; /* c1 保存乘数个位上的数 */
    c2 = a/10%10; /* c2 保存乘数十位上的数 */
    c3 = a/100%10; /* c3 保存乘数百位上的数 */
    for ( i=1; i<b; i++ ) /* 求 a 的 b 次乘方 */
    {
        y %= 1000; /* 被乘数保留三位 */
        y1 = y*c1; /* 被乘数和乘数个位相乘 */
        y2 = y*c2%100; /* 被乘数和乘数十位相乘，结果保留两位 */
        y3 = y*c3%10; /* 被乘数和乘数百位相乘，结果保留一位 */
        y = y1+y2*10+y3*100; /* 部分积相加 */
    }
    printf ("%d\n",y%1000);
}

```

(2) 解法 2。

```

main ( )
{
    int a, b, k;
    long last;
    scanf ("%d%d", &a, &b);
    for ( last=k=1; k<=b; k++)
        last = last * (a%1000) % 1000;
    printf ("last=%ld\n", last );
}

```

6-22.

略

6-23.

(1) 注释：变量 t 储存阶乘，60! 已经超出单精度实型数的范围，所以 t 被说明为 double 型。

```

main( )

```

```

{  int    i=1;
   double t=1;
   float  e=0;
   do
   {      t = t*i++;
         e = e+1./t;
   } while ( i <= 60 );
   printf("%f\n",e);
}

```

(2)

```

main( )
{  int i=1;
   double t=1;
   float e=0;
   do
   {      e = e+1./t;
         t = t*++i;
   } while ( 1/t >= 1.0e-4 );
   printf ("%f\n",e);
}

```

6-24.

```

main( )
{  int n, x1, x2, x3, xn, i;
   printf("Enter n:");
   scanf ("%d",&n);
   if ( n<=3 )
       xn=n;
   else
   {   x1=1; x2=2; x3=3;
       for ( i=4; i<=n; i++ )
       {   xn=x1+x2+x3;
           x1=x2;
           x2=x3;
           x3=xn;
       }
   }
   printf ("Xn=%d\n",xn);
}

```

25.

```

main( )
{  int i, j, k, l;
   for ( i=1; i<=9; i++ )
       for ( j=0; j<=9; j++ )

```

```

        { if ( j==i ) continue;
          for ( k=1; k<=9; k++ )
            { if ( k==i || k==j ) continue;
              for ( l=0; l<=9; l+=2 )
                { if ( l==i || l==j || l == k ) continue;
                  printf ( "%d\n", i*1000+j*100+k*10+l );
                }
              }
            }
        }
    }
}

```

26.

注释：题中水果的价格以元计价，为了避免计算中出现实数，则改为以角为计价单位。

```

main( )
{ int i, j, k;
  for ( i=0; i<=100; i++ )
    for ( j=0; j<=100-i; j++ )
      if ( i*4+j*40+(100-i-j)*2 == 400 )
        printf ( "apple=%d watermelon=%d pear=%d\n", i, j, 100-i-j );
}

```

6-27.

```

main( )
{ int i, j, n;
  printf( "\nPlease Enter n: ");
  scanf ( "%d", &n );
  for ( i=1; i<=n; i++ )
  { for ( j=1; j<=n+i-1; j++ )
    { if ( j > n-i ) printf ( "* ");
      else          printf ( " ");
    }
    printf ( "\n" );
  }
  for ( i=1; i<n; i++ )
  { for ( j=1; j<=2*n-i-1; j++ )
    { if ( j>i ) printf ( "* ");
      else      printf ( " ");
    }
    printf ( "\n" );
  }
}

```

第七章习题参考答案

一、单选题

1. B 2. B 3. D 4. B 5. C 6. C 7. C 8. D

二、填空题

9. □□0□□4□□4□□3(□表示空格)

10. 1

2

4					
8					
16					
32					
64					
128					
256					
512					
11.	0	1	1	2	3
	5	8	13	21	34
	55	89	144	233	377
	610	987	1597	2584	4181

12. 以下程序的功能：把 a 数组的行和列元素互换后存入 b 数组。请填空使程序正确。

```
main( )
{ int i, j, a[2][3] = {1, 2, 3, 4, 5, 6}, b[3][2];          /* 该空不填 */
for ( i=0; i<2; i++ )
{ for (j=0; j<3 ; j++ ) /* 填 j<3 */
    { printf ("%5d ", a[i][j] )}
    b[j][i]=a[i][j] ; } /* 填 b[j][i]=a[i][j] */
printf("\n");}
for(i=0; i<3 ;i++) /* 填 i<3 */
{ for(j=0;j<=1;j++) printf("%5d ",b[i][j]);
    printf("\n"); }
}
```

13. 以下程序的功能：输入 30 个人的年龄，统计 18 岁、19 岁、……、25 岁各有多少人。填空，使程序正确。

```
#define N 8
main( )
{ int i,n,age,a[30]={0};
for(i=0;i<N;i++)/* 边输入年龄 age,边统计该年龄的人数 */
{ scanf("%d",&age); a[age]++ ; } /* 填 a[age]++ */
    printf(" age number\n");
for( i=18;i<26 ;i++) /* 填 i=18;i<26 */
    printf("%5d %6d\n",i,a[i]);
}
```

14. 以下程序的功能：在给定数组中查找某个数，若找到,则输出该数在数组中的位置，否则输出“can not found! ”。填空，使程序正确。

```
main( )
{ int i,n,a[8]={25,21,57,34,12,9,4,44};
```

```
scanf("%d",&n); /* 指定要找的数 n */
for(i=0;i<8;i++)
    if(n==a[i]) /* 找到 n 并输出它的下标 i */
    { printf("The index is %d\n",i);
        break ; } /* 填 break */
if( i>=8 ) printf("can not found!\n"); /* 填 i>=8 */
}
```

15. 以下程序的功能:把两个按升序排列的数组合并成一个按升序排列的数组。填空,使程序正确。

```
main( )
{ int i=0, j=0, k=0, a[3]={5,9,19}, b[5]={12,24,26,37,48}, c[10];
while (i<3 && j<5)
if( a[i]>b[j] ) { c[k]=b[j];k++;j++;} /* 填 a[i]>b[j] */
else { c[k]=a[i];k++;i++;}
while( i<3 ) { c[k]=a[i];k++;i++;} /* 填 i<3 */
while( j<5 ) { c[k]=b[j];k++;j++;} /* 填 j<5 */
for ( i=0; i<k; i++ ) printf("%3d",c[i]);
}
```

16. 求出如下所示矩阵中各行元素之和,并以矩阵形式输出原矩阵及相应行元素之和。填空,使程序正确。

```
3   5   6
2   1   4
8   7   1
main( )
{ int i, j;
static int a[3][4]={ {3,5,6,0}, {2,1,4,0}, {8,7,1,0} };
for(i=0;i<3;i++)
for(j=0;j<3;j++)
a[i][3]+= a[i][j] ; /* 填 a[i][j] */
for(i=0;i<3;i++)
for( j=0;j<4;j++ ) /* 填 j=0;j<4;j++ */
{ printf("%3d",a[i][j]);
if( j==3 ) printf( "\n" );} /* 填 j==3 和 "\n" */
}
```

三、编程题

17. 编写程序,求 N 阶矩阵的两条对角线元素值的和。

```
#define N 3
main( )
{ int i, j, s=0, a[N][N];
for ( i=0; i<N; i++ ) /* 产生 a 数组 */
for ( j=0; j<N; j++ )
```

```

        scanf("%d",&a[i][j]);
for ( i=0; i<N; i++ ) /* 两条对角线上的元素求和 */
for ( j=0; j<N; j++ )
    if ( i==j || i+j==N-1 ) s+=a[i][j];
printf("s=%d\n",s);
}

```

18. 定义一个数组,分别赋予从2开始的30个偶数,然后按顺序每5个元素求一个平均值,并放在数组的末尾。

```

main( )
{ int i,j,k=2; float a[40],s;
  for ( i=0; i<30; i++ )
      { a[i]=k;k+=2; } /* 产生a数组 */
  j=30; s=0;
  for ( i=0; i<30; i++ ) /* 每5个数产生一个平均值依次放入a[j], j=30,31,...,35 */
      { s+=a[i];
        if ( (i+1)%5==0 ) { a[j++]=s/5; s=0; }
      }
  k=30;
  for ( i=0; i<30; i++ ) /* 输出a数组,每5个数及其平均值占一行 */
      { printf("%.1f",a[i]);
        if ( (i+1)%5==0 ) printf(" average=%.1f\n",a[k++]); }
}

```

19. 产生30个随机数到数组中,任意指定位置k,从第k个数开始依次后移3个位置。输出移动前后的数组。

```

#include "stdlib.h"
#define N 9
main( )
{ int i,j,k,a[N+3],s;
  for(i=0;i<N;i++) /*产生N个数,存入数组中并输出 */
      { a[i] = rand();
        printf("%6d",a[i]);
      }
  printf("\n");
  scanf("%d",&k); /* 指定位置K */
  for ( i=N-1+3; i>=k+2; i--) /* 从a[N-1]开始到a[k-1]依次向后移动3个位置 */
      a[i]=a[i-3];
  for ( i=0; i<N+3; i++ ) /* 输出移动后的数组 */
      printf("%6d",a[i]);
  printf("\n");
}

```

20. 编写程序,产生30个随机数到数组中,删除其中的最大值,输出删除前后的数组。

```

#define N 30
main( )
{ int i,k,a[N];
  for ( i=0; i<N; i++ ) /* 产生 a 数组并输出 */
    {a[i] = rand();
     printf("%6d",a[i]);
     if ( (i+1)%6==0 ) printf("\n");}
  printf("\n");
  k=0;
  for ( i=0; i<N; i++ ) /* 找最大值的下标 k */
    if ( a[i] > a[k] ) k=i;
  printf("max=a[%d]=%d\n",k,a[k]);
  for ( i=k; i<N-1; i++ ) /* 删除 a[k] */
    a[i]= a[i+1];
  for ( i=0; i<N-1; i++ ) /* 输出删除 a[k] 后的数组 */
    { printf("%6d",a[i]);
    if((i+1)%6==0) printf("\n");
  }
  printf("\n");
}

```

21. 编写程序,输入任意 10 进制 4 位正整数, 将其化成二进制数。

```

main( )
{ int j,k=0,a[16]={0},s;
  scanf("%d",&s); /* 输入待转化的 10 进制 4 位正整数 s */
  while(s>0)      /* 按“除二取余”得到各位二进制数 a[k]*/
    { a[k]=s%2; k++;
      s=s/2; }
  for(j=15;j>=0;j--) /* 按倒排方式输出二进制数的各位数字 */
    printf("%2d",a[j]);
}

```

22. 编写程序,产生 30 个 50 以内的随机整数到 5 行 6 列数组中, 输出那些在行和列上均为最小的元素。

```

main( )
{ int i,j,p,q,a[5][6],f;
  for(i=0;i<5;i++) /* 产生 a 数组并输出 */
    {for(j=0;j<6;j++)
      {a[i][j]=rand()%50;
       printf("%5d",a[i][j]);}
     printf("\n");}
  f=0; /* 标志量, f=0:未找到目标, f=1:找到目标 */
  for(i=0;i<5;i++)
    {p=0; /* p 记录 i 行最小值列下标 */

```

```

    for(j=1;j<6;j++) /* 找 i 行最小值 a[i][p] */
        if(a[i][j]<a[i][p]) p=j; /* p 记录 i 行最小值列下标 */
    q=0;
    for(j=1;j<5;j++) /* 找 p 列最小值 a[q][p] */
        if(a[j][p]<a[q][p]) q=j; /* q 记录 p 列最小值行下标 */
    if(q==i) /* a[q][p] 是 i 行最小元素,也是 p 列最小元素 */
        {f=1; printf("min=a[%d][%d]=%d\n",q,p,a[q][p]);}
}
}

```

23. 编写程序,产生 30 个[10, 100]中的随机整数到 5 行 7 列数组 a 的前 6 列中,求每行元素值的和,并把和值记录在各行的最后一个元素,如: a[2][6]存放的是 2 行的和。然后将和值最大的行与首行对调。

```

main( )
{ int i, j, p, q, a[5][7], f;
  for(i=0;i<5;i++) /* 产生 a 数组*/
  for(j=0;j<6;j++)
      a[i][j]=rand()%91+10;
  for(i=0;i<5;i++) /* 产生 i 行的和并存放在 a[i][6] */
  { a[i][6]=0;
    for(j=0;j<6;j++)
        a[i][6]+= a[i][j];}
  for(i=0;i<5;i++) /* 输出 a 数组*/
  {for(j=0;j<7;j++)
      printf("%5d",a[i][j]);
    printf("\n");}
  p=0;
  for(i=1;i<5;i++) /* 找行和最大的行 p */
      if(a[i][6]>a[p][6]) p=i;
  for(j=0;j<7;j++) /* 0 行与 p 行交换位置 */
      {q= a[0][j]; a[0][j]= a[p][j]; a[p][j]=q;}
  for(i=0;i<5;i++) /* 输出交换后的 a 数组*/
  {for(j=0;j<7;j++)
      printf("%5d",a[i][j]);
    printf("\n");}
}

```

24. 编写程序,产生 30 个[1, 100]中的随机整数到 5 行 6 列数组中,任意输入一个正整数 k,输出数组中与 k 大小相差 5 以内的所有元素。

```

#include "math.h"
main( )
{ int i, j, k, a[5][6];
  for(i=0;i<5;i++) /* 产生 a 数组*/
  for(j=0;j<6;j++)

```



```

a[i][j]=rand()%100+1;
for(i=0;i<5;i++) /* 输出 a 数组*/
{ for(j=0;j<6;j++)
    printf("%5d",a[i][j]);
printf("\n");}
scanf("%d",&k);
for(i=0;i<5;i++) /* 输出与 k 的差小于 5 的数组元素 */
for(j=0;j<6;j++)
    if(abs(k-a[i][j])<5) printf("%5d",a[i][j]);
printf("\n");
}

```

25. 编写程序，产生 30 个[1, 100]中的随机整数到 5 行 6 列数组中，按升序重新排序，并按列的顺序存放到另一个数组中。输出排序前后的情况。

```

main( )
{ int i, j, t, k=0, a[5][6], b[5][6], c[30];
  for ( i=0; i<5; i++ ) /* 产生 a 数组*/
for ( j=0; j<6; j++ )
    c[k++] = a[i][j] = rand()%100+1;
  for ( i=0; i<29; i++ ) /* c 数组从小到大排序 */
  { k=i;
for ( j=i+1; j<30; j++ )
    if ( c[k]>c[j] ) k=j;
if ( k!=i ) { t=c[k]; c[k]=c[i]; c[i]=t; }
  }
  k=0;
  for ( j=0; j<6; j++ ) /* 产生 b 数组*/
for ( i=0; i<5; i++ )
    b[i][j] = c[k++];
  for ( i=0; i<5; i++ ) /* 输出 a 数组*/
  { for ( j=0; j<6; j++ )
      printf("%5d",a[i][j]);
printf("\n");
  }
for ( i=0; i<5; i++ ) /* 输出 b 数组*/
{ for ( j=0; j<6; j++ )
    printf("%5d",b[i][j]);
printf("\n");
  }
}

```

26. 编写程序，产生 30 个[1, 100]中的随机整数到 5 行 6 列数组中，求其中最大值和最小值，并把最大值元素与右上角元素对调，把最小值元素与左下角元素对调。输出重排前后的情况。

```

main( )
{ int i, j, t, k=0, a[5][6];
  for ( i=0; i<5; i++ ) /* 产生 a 数组并输出*/
  { for ( j=0; j<6; j++ )
    { a[i][j] = rand()%100+1;
    printf("%5d", a[i][j]);
    }

    printf("\n");
  }

  m1=m2=n1=n2=0; /* m1, m2 记录最大值下标; n1, n2 记录最小值下标 */
  for ( i=0; i<5; i++ ) /* 产生最大和最小值*/
  for ( j=0; j<6; j++ )
  { if ( a[m1][m2]>a[i][j] ) { m1=i; m2=j; }
    if ( a[n1][n2]>a[i][j] ) { n1=i; n2=j; }
  }

  t=a[0][5]; a[0][5]= a[m1][m2]; a[m1][m2]=t; /* 最大值与右上角 a[0][5]交换 */
  if ( n1==0 && n2==5 ) /* 若最小值在 a[0][5], 则最小值被交换到 a[m1][m2] */
  { t=a[4][0]; a[4][0]= a[m1][m2]; a[m1][m2]=t; } /* a[m1][m2]与 a[4][0]交换 */
  else
  { t=a[4][0]; a[4][0]= a[n1][n2]; a[n1][n2]=t; } /* 最小值与左下角 a[4][0]交换 */
  for ( i=0; i<5; i++ ) /* 输出交换后的 a 数组 */
  { for ( j=0; j<6; j++ )
    printf("%5d", a[i][j]);
    printf("\n");
  }
}

```

第八章参考答案

一、单选题

1. A 2. C 3. A 4. D 5. A

二、填空题

6. 以下程序的输出是_bfi__。 7. 以下程序的输出是 CDABC 。

8. 以下程序的输出是 ahAMa。 9. 以下程序的输出是 BLMNP。

10. 填空

```

#include    "ctype.h"
#include    "stdio.h"
main( )
{ char str[81]; int i;
  for ( i=0; i<80; i++ )
  { str[i] = getchar( );
    if ( str[i] =='\n' ) break;
  }

  str[i] = '\0' ;
  i=0;

```

```

        while ( str[i] ) putchar( str[i++]);
    }

```

11. 填空

```

#include "stdio.h"
main( )
{ int j,k; char a[10][80],b[10];
  for ( j=0; j<10; j++ ) gets(a[j]);
  for ( j=0; j<10; j++ )
  { b[j]=a[j][0];
    for ( k=1; a[j][k]!='\0'; k++ )
    if ( b[j]<a[j][k] ) b[j]=a[j][k];
  }
  for ( j=0; j<10; j++ )
    printf("%d %c\n",j,b[j]);
}

```

12. 填空

```

main( )
{ int j,k; char a[80],m;
  gets(a);
  for(j=k=0; a[j]!='\0';j++)
  if(a[j]!='c' && a[j]!='C') a[k++]=a[j];
  a[k]='\0';
  printf("%s\n",a);
}

```

三、编程题

13. 编写程序，实现 gets() 函数的功能。

```

#include "stdio.h"
gets1(char s[ ])
{int j=0;
  char c;
  while((c=getchar())!='\n')
    s[j++]=c;
  s[j]='\0';
}
main()
{char s[80];
  gets1(s);
  printf("%s\n",s);
}

```

14. 编写程序，实现 puts() 函数的功能。

```

#include "stdio.h"
puts1(char s[ ])

```

```

{ int j=0;
  while ( s[j]!='\0' )
    { putchar(s[j]); j++; }
  putchar('\n');
}
main( )
{ char s[80];
  scanf("%s",s);
  puts1(s);
}

```

15. 编写程序，判断给定字符串是否回文。回文是指顺读和倒读都一样的字符串。

```

#include "stdio.h"
#include "string.h"
main( )
{ char a[80];
  int i, j, k, p;
  gets(a); k=strlen(a);
  p=1;
  for ( i=0, j=k-1; i<j; i++, j-- ) /* 检查是否回文。P=1: 是; p=0: 否 */
    if ( a[i]!=a[j] ) { p=0; break; }
  if ( p!=0 ) printf("a=%s, yes\n", a);
  else printf("a=%s, no\n", a);
}

```

16. 编写程序，任意输入一个字符串，将其中的字符按从小到大的顺序重排。

```

#include "stdio.h"
#include "string.h"
main( )
{ char a[80];
  int i, j, k, t, p;
  gets(a); k=strlen(a);
  for ( i=0; i<k-1; i++ ) /* 选择法从小到大排序 */
  { p=i;
    for ( j=i+1; j<k; j++ )
      if ( a[p]>a[j] ) p=j;
    if ( p!=i ) { t=a[p]; a[p]=a[i]; a[i]=t; }
  }
  printf("a=%s\n", a);
}

```

17. c 编写程序，任意输入一个字符串，将其中的最大字符放在字符串的第 2 个字符位置，将最小字符放在字符串的倒数第 2 个字符位置。

```

#include "stdio.h"

```

```

#include "string.h"
main( )
{
    char a[80],max,min;
    int i,j,k,t,p;
    gets(a); k=strlen(a);
    max=min=0;
    for ( i=1; i<k; i++ )    /* 找最大值和最小值的下标 max、min */
    {
        if ( a[max]<a[i] ) max=i;
        if ( a[min]>a[i] ) min=i;
    }
    t=a[1]; a[1]=a[max]; a[max]=t;    /* 最大值 a[max]与 a[1]交换 */
    if ( min==1 ) /*若最小值是 a[1]，则 a[1]已交换到 a[max]，故 a[max]与 a[k-2]交换 */
    { t=a[k-2]; a[k-2]=a[max]; a[max]=t; }
    else { t=a[k-2]; a[k-2]=a[min]; a[min]=t; }
    printf("a=%s\n",a);
}

```

18.c 编写程序，任意输入 a、b 两个字符串，将 b 串中的最大字符插入到 a 串的最小字符的前面。

```

#include "stdio.h"
main( )
{
    char a[80],b[80];
    int i,j,k1,max,min;
    gets(a); gets(b);
    k1=strlen(a);
    max=min=0;
    for ( i=1; b[i]!='\0'; i++ ) /* 找 b 串中的最大字符下标 */
        if ( b[max]<b[i] ) max=i;
    for ( i=1; a[i]!='\0'; i++ ) /*找 a 串的最小字符下标 */
        if ( a[min]>a[i] ) min=i;
    for ( i=k1+1; i>min; i-- ) /* a 串自 a[k1]到 a[min]向后移一个位置 */
        a[i]=a[i-1];
    a[min]=b[max]; /* 在 a[min]处插入 b[max] */
    printf("b[max]=%c,a=%s\n",b[max],a);
}

```

19.c 编写程序，输入一个 3 位正整数，计算其各位数字的和值，取该和值被 13 除的余数，若余数为零，则输出****，否则输出对应的月份英文单词。输出形式如下（以整数 539 和 247 为例）：

539: 5+3+9=17, 17%13=4, April

247: 2+4+7=13, 13%13=0, ****

```

main( )
{
    int i,j,k,n,t,p;
    char a[13][20] = {"****", "January", "February", "March", "April", "May", "June",
    "July", "August",

```

```

"September", "October", "November", "December" };
scanf("%d",&n);
i=n/100; j=n/10%10; k=n%10;
p=i+k+j;
t=p%13;
printf("%d: %d+%d+%d=%d, %d%%13= %d,%s\n",n,i,j,k,p,p,t,a[t]);
}

```

8-20. c 编写程序，任意输入 5 个字符串存放到 2 维数组中，按字符串的长度从短到长顺序输出它们。

```

#include "string.h"
main( )
{ int i,j,k,n,t,p,b[5];
  char a[5][80];
  for ( i=0; i<5; i++ ) /* 输入各字符串，并在 b 数组中记录各串的行号 */
  { gets(a[i]); b[i]=i; }
  for ( i=0; i<4; i++ ) /* 选择法按字符串的长度从短到长排序 */
  { k=i;
    for ( j=i+1; j<5; j++ )
      if ( strlen(a[b[k]])>strlen(a[b[j]]) ) k=j;
    if ( k!=i ) { t=b[k]; b[k]=b[i]; b[i]=t; }
  }
  for ( i=0; i<5; i++ )
    printf("%d,%s\n",b[i],a[b[i]]);
}

```

21. 编写程序，任意输入 5 个字符串存放到 2 维数组中，删除其中最小字符串的第 2 个字符。输出原 5 个字符串和删除前后的最小字符串。

```

#include "string.h"
main( )
{ int i,min;
  char a[5][80];
  for ( i=0; i<5; i++ )
    gets(a[i]);
  min=0;
  for ( i=1; i<5; i++ ) /* 找最小串行号 */
    if ( strcmp(a[min],a[i])>0 ) min=i;
  for ( i=1; a[min][i]!='\0'; i++ ) /* 删除最小串的第二字符 */
    a[min][i] = a[min][i+1];
  for ( i=0; i<5; i++ )
    printf ("%s\n",a[i]);
  printf("min string=%s\n", a[min]);
}

```

22. c 请编写输入以下图案的程序，图案的行数由输入的值确定。

```
      A
     BBB
    CCCCC
   DDDDDDD
  EEEEEEEEE
```

```
#include "stdio.h"
main( )
{   int i, j, n;
    char c='A', d=' ';
    scanf("%d", &n);
    for ( i=1; i<=n; i++ )    /* 对 n 行循环 */
    {   for ( j=1; j<=n-i; j++ )   printf(" ");
        for ( j=1; j<=2*i-1; j++ )   printf("%c", c);
        printf("\n");
        c++;
    }
}
```

第九章参考答案

一、选择题

1. B 2. C 3. C 4. A 5. A 6. C 7. C 8. D 9. D 10. B
11. A 12. C 13. A 14. C 15. B

二、填空题

16. 变量的存储类型有 4 种，它们是 自动变量、寄存器变量、静态变量和外部变量。
17. 下面的函数 sum (int n) 完成计算 1~n 的累加和。

```
sum (int n)
{   if (n<=0)
        printf ("data error\n");
    if (n==1)    ① ;    /* return (1) */
    else         ② ;    /* return ( n+sum(n-1) ) */
}
```

18. 下面的函数是一个求阶乘的递归调用函数。

```
facto (int n)
{   if ( n == 1 )    ① ;    /* return (1) */
    else return (    ② );    /* return(n*facto(n-1)) */
}
```

三、编程题

19. 编写一个判断一个整数是否是素数的函数，使用该函数编写验证 1000 以内的哥德巴赫猜想是成立。

```
#include <stdio.h>
main( )
{   int i, j;
    for ( i=6; i<=1000; i+=2 )
        for ( j=3; j<=i/2; j+=2 )
```

```

        if ( flag(j) && flag(i-j) ) {
            printf("%d=%d+%d\n", i, j, i-j);
            break;
        }
    }
}

flag (n)          /* 函数 flag 的功能是判断整数 n 是否为素数 */
{
    int n;
    {
        int i;
        if ( n%2 == 0) return(0);
        for ( i=3; i<n/2; i++ )
            if ( n%i==0 )
                return(0);
        return(1);
    }
}

```

20. 编写一个程序，调用函数已知一个圆筒的半径、外径和高，计算该圆筒的体积。

参考程序：

```

#include <stdio.h>
double v ( double r, double h )
{
    return (3.1415926*r*r*h);
}

main( )
{
    double r1, r2, h, v();    /* r1: 外圆半径  r2: 内圆半径  h: 圆筒的高 */
    printf ("Enter r1 r2 h:");
    scanf ("%lf%lf%lf", &r1, &r2, &h);
    printf ("v=%lf\n", v(r1,h)-v(r2,h) );
}

```

21. 编写一个求水仙花数的函数，求 100 到 999 之间的全部水仙花数。所谓水仙花数是指一个三位数，其各位数字立方的和等于该数。例如：153 就是一个水仙花数：

$$153 = 1*1*1 + 5*5*5 + 3*3*3$$

参考程序：

```

main( )
{
    int i=0, j, k, a, b, c, s;
    for (a=1; a<=9; a++)
        for (b=0; b<=9; b++)
            for (c=0; c<=9; c++)
                {
                    j=100*a+10*b+c;
                    if ( ex(j) )
                        printf ("%d=%d*d*d+%d*d*d+%d*d*d\n", j, a, a, a, b, b, b, c, c, c);
                }
}

int ex (int m)

```



```

{   int sum=0, z, k;
    z=m;
    while (z>0)   {
        k= z%10;
        sum += k*k*k;
        z /= 10;
    }
    return ( m==sum );
}

```

22. 请编写一个函数，输出整数m的全部素数因子。例如：m=120 时，因子为：

2, 2, 2, 3, 5

参考答案：

```

main ( )
{   int m;
    printf ("\nEnter m=");
    scanf ("%d", &m);
    primedec (m);
}

primedec(m)
int m;
{   int k=2;
    while (k<=m)
        if (m%k == 0 )
        {   printf ("%d, ", k);
            m = m/k;
        }
        else k++;
}

```

23. 已知某数列前两项为 2 和 3，其后继项根据当前的前两项的乘积按下列规则生成：① 若乘积为一位数，则该乘积就是数列的后继项；② 若乘积为二位数，则乘积的十位和个位数字依次作为数列的后继项。当 N=10，求出该数列的前十项为：

2 3 6 1 8 8 6 4 2 4。

参考程序：

```

#include "stdio.h"
#define MAXNUM 100
void sum (n, pa)
int n, pa[];
{   int count, temp;
    pa[0] = 2;
    pa[1] = 3;
    count=2;
    while ( count < n )

```

```

    { temp = pa[count-1] * pa[count-2];
      if ( temp<10 )
        pa[count++] = temp;
      else
        { pa[count++] = temp/10;
          if ( count<n )
            pa[count++] = temp%10;
          }
    }
  }
}

main( )
{ int n, i, num[MAXNUM];
  do
  { printf ("Input N=? (2<N<=%d):", MAXNUM);
    scanf ("%d", &n);
  } while ( n<=2 || n>MAXNUM );
  printf ("\nsum(%d)=", n);
  sum(n, num);
  for ( i=0 ; i<n; i++ )
    printf ("%4d", num[i]);
  printf ("\n");
}

```

24. 求组合数。编程计算：

参考答案：

```

#include "stdio.h"

main( )
{ int m, n;
  long cmn, facto( );          /* 说明函数 facto 的类型 */
  printf ("Input m and n =");
  scanf ("%d%d", &m, &n);
  cmn = facto(m) / (facto(n) * facto(m-n)); /* 三次调用 facto 函数 */
  printf ("The combination is %ld\n", cmn);
}

long facto (x)                 /* 定义 facto 函数。从在源程序中的位置 */
int x;                          /* 来说，facto 函数定义在后，调用在前 */
{ long y;
  for ( y=1; x>0; --x )
    y*=x;
  return(y);
}

```

25. 已知 ackermann 函数，对于 $m \geq 0$ 和 $n \geq 0$ 有如下定义：

$\text{ack}(0, n) = n + 1$

```

ack(m, 0)=ack(m-1, 1)
ack(m, n)=ack(m-1, ack(m, n-1))

```

请编程输入 m 和 n，求出 ack(m, n) 之值。

参考程序：

```

#include <stdio.h>

ackerman ( int m, int n )
{
    if ( m==0 ) return (n+1);
    if ( n==0 ) return ( ackerman ( m-1, 1) );
    else      return ( ackerman ( m-1, ackerman(m, n-1) ) );
}

main( )
{
    int m, n;
    printf ("Enter ackerman m n:");
    scanf ("%d%d", &m, &n);
    printf ("ackerman(%d,%d)=%d\n", m, n, ackerman(m, n) );
}

```

26. 用递归的方法打印杨辉三角形。

```

          1
        1  1
       1  2  1
      1  3  3  1
     1  4  6  4  1
    1  5 10 10  5  1
   .....

```

参考答案：

杨辉三角形中的数，正是 $(x+y)^N$ 次方幂展开式中各项的系数。本题作为程序设计中具有代表性的题目，求解的方法很多（可以使用一维数组，也可以使用二维数组），这里仅给出一种使用递归求解的方法。

从杨辉三角形的特点出发，可以总结出：

① 第 N 行有 N+1 个值（设起始行为第 0 行）；

② 对于第 N 行的第 J 个值： $(N \geq 2)$

当 $J=1$ 或 $J=N+1$ 时： 其值为 1

当 $J \neq 1$ 且 $J \neq N+1$ 时： 其值为第 N-1 行的第 J-1 个值与第 N-1 行第 J 个值之和

将这些特点提炼成数学公式可表示为：

$$c(x, y) = 1 \quad x=1 \text{ 或 } x=N+1$$

$$c(x, y) = c(x-1, y-1) + c(x-1, y) \quad \text{其它}$$

下面程序就是根据以上递归的数学表达式编制的。

```

#include <stdio.h>

main( )
{
    int i, j, n=13;
    printf("N=");
    while ( n>12 )
        scanf ("%d", &n);          /* 最大输入值不能大于 12 */
}

```

```

        for ( i=0; i<=n; i++ )                /* 控制输出 N 行 */
        { for ( j=0; j<12-i; j++ )
            printf("  ");                      /* 控制输出第 i 行前面的空格 */
            for ( j=1; j<i+2; j++ )
                printf("%6d", c(i, j));        /* 输出第 i 行的第 j 个值 */
            printf("\n");
        }
    }

    int c(x, y)                                /* 求杨辉三角形中第 x 行第 y 列的值 */
    { int x, y;
      { int z;
        if ( (y==1) || (y==x+1) )
            return(1);                        /* 若为 x 行的第 1 或第 x+1 列, 则输出 1 */
        else                                  /* 否则: 其值为前一行中第 y-1 列与第 y 列值之和 */
            z = c(x-1, y-1) + c(x-1, y);
        return(z);
      }
    }

```

27. 编写一递归程序实现任意正整数向八进制数的转换。

参考程序:

```

#include <stdio.h>

main ( )
{ int m;
  printf ("\nEnter m=");
  scanf ("%d", &m);
  printhex (m);
}

printhex ( int m )
{ if ( m>=8 )
    printhex (m/8);
  printf ("%d", m%8);
}

```

28. 请编写一个模拟轮盘赌的程序(a roulette program)。该轮盘赌机器随机的在 0 到 35 之间选一个数字, 游戏者可以打奇 / 偶赌或打具体数的赌。赢得奇 / 偶数赌时游戏者以 2 比 1 获奖, 但如果轮盘赌机器选中 0 则所有奇 / 偶赌都输掉。如果游戏者打一个具体数的赌, 而轮盘赌机器正好选中此数, 则他以 35 比 1 获奖。

参考程序: 略。

29. 验证卡布列克运算。任意一个四位数, 只要它们各个位上的数字是不全相同的, 就有这样的规律:

- ① 将组成这个四位数的四个数字由大到小排列, 形成由这四个数字构成的最大的四位数;
- ② 将组成这个四位数的 4 个数字由小到大排列, 形成由这 4 个数字构成的最小的四位数 (如果四个数字中含有 0, 则得到的数不足四位);
- ③ 求两个数的差, 得到一个新的四位数。

重复以上过程，最后得到的结果总是 6174。

参考答案：

根据原题中描述的算法，可以将整个进行卡布列克运算的过程做成递归，这样原算法描述中的“重复执行卡布列克运算”就可以变成了递归调用。可以很容易写出递归算法如下。

进行“卡布列克运算”

若 结果不是 6174

则 递归进行“卡布列克运算”

否则 结束整个程序

程序如下：

```
#include <stdio.h>

int count=0;                                /* 计数器 */
main( )
{ int n;
  printf("Enter a number:");
  scanf("%d",&n);                          /* 输入任意正整数 */
  vr6174(n);                                /* 调用函数进行验证 */
}

vr6174 (num)
{ int num;
  { int each[4],max,min;
    if ( num!=6174 && num ) /* 若不等于 6174 且不等于 0 则进行卡布列克运算 */
    { parse_sort(num, each); /* 将整数分解，数字存入 each 数组中 */
      max_min(each,&max,&min); /* 求数字组成的最大值和最小值 */
      num=max-min;           /* 求最大值和最小值的差 */
      printf(" [%d]: %d-%d=%d\n", ++count, max, min, num);
                                /* 输出该步计算过程 */
      vr6174(num);             /* 递归调用自身继续进行卡布列克运算 */
    }
  }
}

parse_sort(num, each) /* 函数将整数分解，数字存入 each 数组中 */
{ int num,*each;
  { int i,*j,*k,temp;
    for ( i=0; i<=4; i++ ) /* 将 NUM 分解为数字 */
    { j=each+3-i;
      *j=num%10;
      num/=10;
    }
    for ( i=0; i<3; i++ ) /* 对各个数字从小到大进行排序 */
      for ( j=each,k=each+1; j<each+3-i; j++,k++ )
        if ( *j>*k )
          { temp=*j; *j=*k; *k=temp; }
    return;
  }
}

max_min ( each, max, min ) /* 将分解的数字还原为最大整数和最小整数 */
```

```
int *each, *max, *min;
{ int *i;
  *min = 0;
  for ( i=each; i<each+4; i++ )          /* 还原为最小的整数 */
    *min = *min * 10 + *i;
  *max = 0;
  for ( i=each+3; i>=each; i-- )        /* 还原为最大的整数 */
    *max = *max * 10 + *i;
  return;
}
```