/\*-------------------------------------------------------

【程序填空】

---------------------------------------------------------

下面程序接受键盘上的输入，直到按↙键为止，这些字符被原样

输出，但若有连续的一个以上的空格时只输出一个空格，请填(3)

空使程序完整。

-------------------------------------------------------\*/

#include<stdio.h>

void main( )

{

char cx , front='\0' ;

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

while ((【?】)!='\n')

{

if (cx!=' ') putchar(cx) ;

if (cx==' ')

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

if (【?】)

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

putchar(【?】);

front=cx ;

}

}

/\*-------------------------------------------------------

【程序填空】

---------------------------------------------------------

del函数的作用是删除有序数组a中的指定元素x，n为数组a的元素

个数，函数返回删除后的数组a元素个数。主函数调用del删除5，

输出删除后的数组，请填(3)空使程序完整。

-------------------------------------------------------\*/

#include<stdio.h>

void main()

{ int del (int a[10],int n,int x);

int x[10]={1,2,3,4,5,6,7,8,9,10};

int i,y;

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

y=【?】;

for(i=0;i<y;i++)

printf("%d ",x[i]);

}

int del (int a[10],int n,int x)

{ int p=0,i ;

while (x>=a[p]&&p<n)

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

【?】;

for (i=p-1; i<n; i++)

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

【?】;

return (n-1) ;

}

/\*-------------------------------------------------------

【程序填空】

---------------------------------------------------------

下面程序的功能是输入10个数，找出最大值和最小值所在的位置，

并把两者对调，然后输出调整后的10个数。

请填（4）空使程序完整。

--------------------------------------------------------\*/

#include<stdio.h>

void main()

{ int a[10],max,min,i,j,k ;

k=0;j=0;

for (i=0; i<10; i++)

scanf("%d",&a[i]);

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

【?】=a[0];

for (i=0; i<10; i++)

{

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

if (a[i]<min) { min=a[i];【?】; }

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

if (a[i]>max) { max=a[i];【?】; }

}

a[j]=max;

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

【?】;

for (i=0; i<10; i++)

printf("%d ",a[i]);

}

/\*-------------------------------------------------------

【程序填空】

---------------------------------------------------------

以下程序的功能是计算函数f=x/y+y/z ,请填(3)空使程序完整。

-------------------------------------------------------\*/

# include <stdio.h>

main ( )

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

{ 【?】; /\*函数原型声明\*/

float x,y,z,f ;

scanf("%f,%f,%f",&x,&y,&z);

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

f = fun (【?】);

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

f += fun (【?】);

printf("f=%f",f);

}

float fun(float a,float b)

{

return (a/b) ;

}

/\*-------------------------------------------------------

【程序填空】

---------------------------------------------------------

以下程序是将字符串b的内容连接字符数组a的内容后面，形成新

字符串a，请填(2)空使程序完整。

-------------------------------------------------------\*/

#include<stdio.h>

void main ( )

{ char a[40]="Great ", b[ ]="Wall";

int i=0,j=0 ;

while (a[i]!='\0') i++ ;

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

while (【?】) {

a[i]=b[j] ; i++ ; j++ ;

}

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

【?】;

printf("%s\n",a);

}

/\*-------------------------------------------------------

【程序填空】

---------------------------------------------------------

下面程序用"插入法"对数组a进行由小到大的排序，请填(3)空使

程序完整。

经典算法提示：

简单插入排序算法的基本思想使将数组处理n-1次，第k次处理是

将第k个元素插入到目前的位置。第k次的元素是这样插入的:在

第k次处理时，前面的元素a[0],a[1],…,a[k-1]必定已排成了升

序，将a[k]与a[k-1],a[k-2],…a[0]逐个比较（由后向前），若

有a[j]<a[k],则a[k]插入到a[j]之后，否则a[k]维持原位不变。

-------------------------------------------------------\*/

#include<stdio.h>

void main ( )

{ int a[10]={191,3,6,4,11,7,25,13,89,10} ;

int i,j,k ;

for (i=1; i<10; i++) {

k = a[i] ;

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

j = 【?】;

while (j>=0 && k<a[j] ) {

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

【?】;

j--;

}

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

【?】=k ;

}

for (i=0; i<10; i++) printf("%d ",a[i]);

}

/\*-------------------------------------------------------

【程序填空】

---------------------------------------------------------

已知程序的功能是输出数组a 中的最大值，及该元素所在的行号

和列号，请填(3)空使程序完整。

-------------------------------------------------------\*/

#include<stdio.h>

void main( )

{ int i,j,row=0,col=0,max;

int a[4][3]={{4,5,6},{10,1,23},{2,-8,9},{9,7,18}};

max=a[0][0];

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

for(【?】)

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

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

if(【?】)

{

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

【?】;

row=i;

col=j;

}

printf("%d,%d,%d \n",max,row,col);

}

/\*-------------------------------------------------------

【程序填空】

---------------------------------------------------------

下面程序接受键盘上的输入，直到按↙键为止，这些字符被原样

输出，但若有连续的一个以上的空格时只输出一个空格，请填(3)

空使程序完整。

-------------------------------------------------------\*/

#include<stdio.h>

void main( )

{

char cx , front='\0' ;

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

while ((【?】)!='\n')

{

if (cx!=' ') putchar(cx) ;

if (cx==' ')

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

if (【?】)

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

putchar(【?】);

front=cx ;

}

}

/\*-------------------------------------------------------

【程序填空】

---------------------------------------------------------

以下程序是将字符串b的内容连接字符数组a的内容后面，形成新

字符串a，请填(2)空使程序完整。

-------------------------------------------------------\*/

#include<stdio.h>

void main ( )

{ char a[40]="Great ", b[ ]="Wall";

int i=0,j=0 ;

while (a[i]!='\0') i++ ;

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

while (【?】) {

a[i]=b[j] ; i++ ; j++ ;

}

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

【?】;

printf("%s\n",a);

}

/\*-------------------------------------------------------

【程序填空】

---------------------------------------------------------

下面程序用"插入法"对数组a进行由小到大的排序，请填(3)空使

程序完整。

经典算法提示：

简单插入排序算法的基本思想使将数组处理n-1次，第k次处理是

将第k个元素插入到目前的位置。第k次的元素是这样插入的:在

第k次处理时，前面的元素a[0],a[1],…,a[k-1]必定已排成了升

序，将a[k]与a[k-1],a[k-2],…a[0]逐个比较（由后向前），若

有a[j]<a[k],则a[k]插入到a[j]之后，否则a[k]维持原位不变。

-------------------------------------------------------\*/

#include<stdio.h>

void main ( )

{ int a[10]={191,3,6,4,11,7,25,13,89,10} ;

int i,j,k ;

for (i=1; i<10; i++) {

k = a[i] ;

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

j = 【?】;

while (j>=0 && k<a[j] ) {

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

【?】;

j--;

}

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

【?】=k ;

}

for (i=0; i<10; i++) printf("%d ",a[i]);

}

/\*-------------------------------------------------------

【程序填空】

---------------------------------------------------------

已知程序的功能是输出数组a 中的最大值，及该元素所在的行号

和列号，请填(3)空使程序完整。

-------------------------------------------------------\*/

#include<stdio.h>

void main( )

{ int i,j,row=0,col=0,max;

int a[4][3]={{4,5,6},{10,1,23},{2,-8,9},{9,7,18}};

max=a[0][0];

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

for(【?】)

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

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

if(【?】)

{

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

【?】;

row=i;

col=j;

}

printf("%d,%d,%d \n",max,row,col);

}

下面程序可求出矩阵a的主对角线上的元素之和，请填(2)空使程序完整。

**习题一**

    main ( )

    { int a[3][3]={1,3,5,7,9,11,13,15,17} , sum=0, i, j ;

       for (i=0 ; i<3 ; i++)

main ( )

{

Int a[3][3]={1,3,5,7,9,11,13,15,17} , sum=0,i,j;

for (i=0;i<3;i++)

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

if (i==j)

sum=sum+a[i][j];

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

}

           for (j=0 ; j<3 ; j++)

              if (【1 】)

                sum=sum+【2 】；

       printf(“sum=%d”,sum);

}

下面程序将十进制整数base转换成n进制，请填(3)空使程序完整。

main ( )

main ( )

{

int i=0,base,n,j,num[20]={0};

scanf("%d",&n);

scanf("%d",&base);

do {

i++;

num[i]=base%n;

base=base/n;

} while (base!=0);

**for (j=i;j>=1;j--)**

printf("%d",num[j]);

}

{ int i=0,base,n,j,num[20]={0};

   scanf(“%d”,&n);

   scanf(“%d”,&base);

   do {

      i++;

      num[i]=【1 】;

      base=【2 】;

   } while (base!=0);

   for (【3 】)

      printf(“%d”,num[j]) ;

}

main ( )

{

int a[10],max,min,i,j,k=0;

for (i=0; i<10; i++)

scanf("%d",&a[i]);

max=min=a[0];

for (i=0; i<10; i++)

{

if (a[i]<min) { min=a[i]; k=i; }

if (a[i]>max) { max=a[i]; j=i; }

}

a[k]=max;

a[j]=min;

for (i=0; i<10; i++)

printf("%d",a[i]);

}

下面程序的功能是输入10个数，找出最大值和最小值所在的位置，并把两者对调，然后输出调整后的10个数，请填(3)空使程序完整。

main ( )

{ int a[10],max,min,i,j,k ;

  for (i=0; i<10; i++)

     scanf(“%d”,&a[i]);

  max=min=a[0];

  for (i=0; i<10; i++) {

     if (a[i]<min) { min=a[i]; 【1 】; }

     if (a[i]>max) {max=a[i]; 【2 】; }

  }

  【3 】;

  for (i=0; i<10; i++)

     printf(“%d”,a[i]);

}

下面程序用“插入法”对数组a进行由小到大的排序，请填(3)空使程序完整。

经典算法提示：

简单插入排序算法的基本思想使将数组处理n-1次，第k次处理是将第k个元素插入到目前的位置。第k次的元素是这样插入的:在第k次处理时，前面的元素a[0],a[1],…,a[k-1]必定已排成了升序，将a[k]与a[k-1],a[k-2],…a[0]逐个比较（由后向前），若有a[j]<a[k],则a[k]插入到a[j]之后，否则a[k]维持原位不变。

main ( )

{ int a[10]={191,3,6,4,11,7,25,13,89,10} ;

main ( )

{ int a[10]={191,3,6,4,11,7,25,13,89,10} ;

int i,j,k ;

for (i=1; i<10; i++)

{

k = a[i];

j = i-1;

while (j>=0 && k<a[j])

{

a[j+1]=a[j];

j--;

}

a[j+1]=k; //a[3]=a[5]最小的放后面

}

for (i=0; i<10; i++) printf("%d",a[i]);

}

   int i,j,k ;

   for (i=1; i<10; i++) {

      k = a[i] ;

      j = 【1 】;

      while (j>=0 && k<a[j] ) {

         【2 】;

          j--;

      }

      【3 】=k ;

   }

   for (i=0; i<10; i++) printf(“%d”,a[i]);

}

main ( )

{

int a[3]={5,9,10} ;

int b[5]={12,24,26,37,48} ;

int c[10],i=0,j=0,k=0 ;

while (i<3 && j<5)

if (a[i]>b[j])

{c[k]=b[j] ; k++ ; j++ ;}

else

{c[k]=a[i] ; k++ ; i++ ;}

**while (i<3 && j>=5)**

{ c[k]=a[i] ; i++ ; k++ ; }

**while (i>=3 && j<5)**

{ c[k]=b[j] ; j++ ; k++ ; }

for (i=0; i<k; i++) printf("%d ",c[i]);

}

下面程序用“两路合并法”把两个已按升序（由小到大）排列的数组合并成一个新的升序数组，请填(3)空使程序完整。

main ( )

{

int a[3]={5,9,10} ;

   int b[5]={12,24,26,37,48} ;

   int c[10],i=0,j=0,k=0 ;

   while (i<3 && j<5)

      if (【1 】) {

         c[k]=b[j] ; k++ ; j++ ;

      } else {

         c[k]=a[i] ; k++ ; i++ ;

      }

   while (【2 】) {

      { c[k]=a[i] ; i++ ; k++ ; }

   while (【3 】) {

      { c[k]=b[j] ; j++ ; k++ ; }

   for (i=0; i<k; i++) printf(“%d ”,c[i]);

}

下面程序的功能是将二维数组a中每个元素向右移一列，最右一列换到最左一列，移后的结果保存到b数组中，并按矩阵形式输出a和b,请填(3)空使程序完整。

main ( )

{

int a[2][3]={{4,5,6},{1,2,3}}, b[2][3]={0}, i,j;

for (i=0; i<2 ; i++)

{

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

{

printf("%5d",a[i][j]);

b[i][j+1]=a[i][j];

}

printf("\n");

}

for (i=0;i<2;i++) b[i][0]=a[i][2];

for(i=0;i<2;i++)

{

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

{

printf("%5d",b[i][j]);

}

printf("\n");

}

}

main ( )

{ int a[2][3]{{4,5,6},{1,2,3}}, b[2][3], i,j;

   for (i=0; i<2 ; i++)

     for (j=0; j<3 ; j++ ) {

         printf(“%5d”,a[i][j]);

        【1 】;

     }

     printf(“\n”);

   }

   for (【2 】) b[i][0]=a[i][2];

   for (i=0; i<2 ; i++) {

      for (j=0; j<3 ; j++) {

          printf(“%5d”,b[i][j]);

          【3 】;

      }

      printf(“\n”);

   }

}

下面程序的功能是在一个字符串中查找一个指定的字符，若字符串中包含该字符则输出该字符在字符串中第一次出现的位置（下标值），否则输出－1，请填(2)空使程序完整。

# include <string.h>

main ( )

{ char c=’a’ ; /\* 需要查找的字符\*/

   char t[50] ;

main ( )

{ char c='a' ; /\* 需要查找的字符\*/

char t[50] ;

int i,j,k;

gets(t) ;

i = strlen(t);

for (k=0; k<i; k++)

**if (t[k]==c)**

{ j = k ; break ;}

else j=-1;

printf("%d",j);

}

   int i,j,k;

   gets(t) ;

   i = 【1 】;

   for (k=0; k<i; k++)

      if (【2 】)

         { j = k ; break ;}

      else j=-1;

   printf(“%d”,j);

}

下面程序用“顺序查找法”查找数组a中是否存在某一关键字，请填(2)空使程序完整。

main ( )

{ int a[10]={25,57,48,371,123,291,18,22,44};

main ( )

{

int a[10]={25,57,48,371,123,291,18,22,44};

int i=0,x ;

scanf("%d",&x);

while (a[i]!=x &&i <10) i++ ;

if(i<10)printf("found=%d\n",i);

else

printf("can't found!\n");

}

   int i=0,x ;

   scanf(“%d”,&x);

   while (a[i]!=x&&【1 】) i++ ;

   if (【2 】) printf(“found=%d\n”,i);

   else printf(“can’t found!\n”);

}

**习题二**

以下程序是求矩阵a,b的乘积，结果存放在矩阵C中并按矩阵形式输出，请填(3)空使程序完整。

main ( )

{ int a[2][2]={-1,-2,-3,-4}, b[2][3]={2,10,9,4,5,119};

int i,j,k,s=0,c[2][3]={0};

for (i=0; i<2; i++)

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

{ s=0;

for (k=0; k<2; k++) {s+=a[i][k]\*b[k][j]; }

c[i][j]=s;

}

for (i=0; i<2; i++)

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

printf("%6d",c[i][j]);

printf("\n");

}

}

main ( )

{ int a[2][2]={-1,-2,-3,-4}, b[2][3]={2,10,9,4,5,119};

   int i,j,k,s,c[2][3];

   for (i=0; i<2; i++)

      for (j=0; j<3; j++) {

         【1 】

          for (k=0; k<2; k++) s+=【2 】;

          c[i][j]=s;

      }

   for (i=0; i<3; i++)

      {for (j=0; j<2; j++)

         printf(“%6d”,c[i][j]);

         【3 】;

      }

}

以下程序是将字符串b的内容连接字符数组a的内容后面，形成新字符串a，请填(2)空使程序完

main ( )

{ char a[40]="Great ", b[ ]="Wall";

int i=0,j=0 ;

while (a[i]!='\0') i++ ;

while (b[j]!='\0')

{

a[i]=b[j] ; i++ ; j++ ;

}

a[i+1]=’\0’;

printf("%s\n",a);

}

整。

main ( )

{ char a[40]=”Great ”, b[ ]=”Wall”;

   int i=0,j=0 ;

   while (a[i]!=’\0’) i++ ;

   while (【1 】) {

      a[i]=b[j] ; i++ ; j++ ;

   }

   【2 】;

   printf(“%s\n”,a);

}

下面函数用“折半查找法”从有10个数的a数组中对关键字m查找，若找到，返回其下标值，否则返回－1，请填（2）空使程序完整。

经典算法提示：

折半查找法的思路是先确定待查元素的范围，将其分成两半，然后比较位于中间点元素的值。如果该待查元素的值大于中间点元素的值，则将范围重新定义为大于中间点元素的范围，反之亦反。

int search(int a[10],int m)

{ int x1=0,x2=9,mid ;

int search(int a[10],int m)

{ int x1=0,x2=9,mid ;

while (x1<=x2)

{

mid = (x1+x2)/2;

if (m<a[mid]) x2=mid;//这是升序，若果是降序则是X1=MID

else if (m>a[mid]) x1=mid;// 这是升序，若果是降序则是X2=MID，

else return (mid) ;

}

return (-1) ;

}

   while (x1<=x2) {

      mid = (x1+x2)/2;

      if (m<a[mid]) 【1 】;

      else if (m>a[mid]) 【2 】;

      else return (mid) ;

   }

   return (-1) ;

}

del函数的作用是删除有序数组a中的指定元素x，n为数组a的元素个数，函数返回删除后的数组a元素个数，请填(2)空使程序完整。

int del (int a[10],int n,int x)

{ int p=0,i ;

   while (x>=a[p]&&p<n) 【1 】;

int del (int a[10],int n,int x)

{

int p=0,i ;

   while (x>=a[p]&&p<n) p++;

   for (i=p-1; i<n; i++) a[i]=a[i+1];

   return (n-1) ;

}//只能删除升序序列

   for (i=p-1; i<n; i++) 【2 】;

   return (n-1) ;

}

以下程序的功能是计算函数f=x/y+y/z ,请填(3)空使程序完整。

# include <stdio.h>

【1 】;

# include <stdio.h>

#include “math.h”;

main ( )

{ float x,y,z,f ;

   scanf(“%f,%f,%f”,&x,&y,&z);

   f = fun (x,y);

   f += fun (y,z);

   printf(“f=%f”,f);

}

float fun(float a,float b)

{ return (a/b) ;

}

main ( )

{ float x,y,z,f ;

   scanf(“%f,%f,%f”,&x,&y,&z);

   f = fun (【2 】);

   f += fun (【3 】);

   printf(“f=%d”,f);

}//要改%f输出

float fun(float a,float b)

{ return (a/b) ;

}

avg函数的作用是计算数组array的平均值返回，请填(3)空使程序完整。

float avg(float array[10])

float avg(float array[10])

{

int i;

float avgr,sum=0;

for (i=0;i<10;i++)

sum+=array[i];

avgr=sum/10 ;

return(avgr);

}

{ int i ;

   float avgr , sum=0 ;

   for (i=0; 【1 】; i++)

      sum+=【2 】;

   avgr = sum / 10 ;

   【3 】;

**习题三**

下列程序的输出结果是16.00，请填（1）空使程序完整。

main ( )

main ( )

{

int a=9,b=2;

   float x=6.6 , y=1.1 , z;

   z=a/2+b\*x/y+1/2;

   printf(“%5.2f\n”,z);

}

{ int a=9,b=2;

   float x=【1】 , y=1.1 , z;

   z=a/2+b\*x/y+1/2;

   printf(“%5.2f\n”,z);

}

以下程序输出x,y,z三个数中的最小值，请填（2）空使程序完整。

main ( )

{ int x=4,y=5,z=8 ;

   int u,v;

   u = x<y ? x:y;

   v = u<z ? u:z ;

   printf (“%d”,v);

}

main ( )

{ int x=4,y=5,z=8 ;

   int u,v;

   u = x<y ? 【1 】 ;

   v = u<z ? 【2 】 ;

   printf (“%d”,v);

}

下述程序用“碾转相除法”计算两个整数m和n的最大公约数。该方法的基本思想是计算m和n相除的余数，如果余数为0则结束，此时的被除数就是最大公约数。否则，将除数作为新的被除数，余数作为新的除数，继续计算m和n相除的余数，判断是否为0，等等，请填（3）空使程序完整。

main ( )

{ int m,n,w;

   scanf(“%d,%d”,&m,&n);

main ( )

{ int m,n,w;

   scanf(“%d,%d”,&m,&n);

   while (n)

{

      w=m%n；

      m=n；

       n=w；

   }

   printf(“%d”,m);

}

   while (n) {

      w=【1 】；

      m=【2 】；

       n=【3 】；

   }

   printf(“%d”,m);

}

下面程序的功能是输出1至100之间每位数的乘积大于每位数的和的数，请填（3）空使程序完整。

main ( )

{ int n,k=1,s=0,m ;

  for (n=1 ; n<=100 ; n++)

{

      k=1 ; s=0 ;

      m=n；

      while (m)

{

          k\*=m%10;

          s+=m%10;

          m=m/10；

      }

      if (k>s) printf(“%4d”,n);

  }

}

main ( )

{ int n,k=1,s=0,m ;

  for (n=1 ; n<=100 ; n++) {

      k=1 ; s=0 ;

      【1 】；

      while (【2 】) {

          k\*=m%10;

          s+=m%10;

          【3 】；

      }

      if (k>s) printf(“%4d”,n);

  }

}

下面程序段的功能是计算1000！的末尾有多少个零，请填(1)空使程序完整。

main ( )

{

     int i,k,m;

     for (k=0,i=5; i<=1000; i+=5)

     { m = i ;

        while (m%5==0) { k++; m=m/5 ; }

     }printf(“%d”,k);

}

main ( )

{

     int i,k,m;

     for (k=0,i=5; i<=1000; i+=5)

     { m = i ;

        while (【1 】) { k++; m=m/5 ; }

     }printf(“%d”,k);

}

下面程序接受键盘上的输入，直到按↙键为止，这些字符被原样输出，但若有连续的一个以上的空格时只输出一个空格，请填(3)空使程序完整。

main ( )

main ( )

{

    char cx , front=’\0’ ;

while (cx!=’\n’)

{

       if (cx!=’ ’) putchar(cx) ;

       if (cx==’ ’)

          if (cx!=’’)

             putchar(‘’)；

       front=cx ;

}

{

    char cx , front=’\0’ ;

    while (【1 】!=’\n’) {

       if (cx!=’ ’) putchar(cx) ;

       if (cx==’ ’)

          if (【2 】)

             putchar(【3 】)；

       front=cx ;

}