班级：计算机173 姓名：张博翔 学号：201706060334

1. 选择

1、C 2、C 3、B 4、D 5、D 6、B 7、D 8、B 9、C 10、B 11、B 12、B 13、A 14、B 15、C 16、A 17、C 18、B 19、A 20、B 21、D 22、B 23、C 24、C 25、D 26、C 27、C 28、D 29、A 30、B 31、D 32、C 33、C 34、B 35、B 36、D 37、A 38、A 39、C 40、A 41、A 42、D 43、B 44、C 45、B 46、A 47、B 48、C 49、D 50、A 51、C 52、B 53、D 54、C 55、D 56、A 57、A 58、D 59、D 60、D 61、B 62、C 63、B 64、B 65、B 66、B 67、A 68、B 69、A 70、B 71、B 72、C 73、B 74、C 75、C 76、B 77、C 78、B 79、B 80、A 81、C 82、B 83、D 84、B 85、A 86、C 87、A 88、D 89、C 90、C 91、A 92、B 83、B 94、B 95、D 96、B 97、B 98、A 99、A 100、B

1. 改错

1、、求一维数组a中所有元素的平均值,结果保留两位小数。 例如，当一维数组 a 中的元素为：10,4,2,7,3,12,5,34,5,9 程序的输出应为：The aver is: 9.10 。

#include <conio.h>

#include <stdio.h>

int main()

{

int a[10]={10,4,2,7,3,12,5,34,5,9},i;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

double aver,s;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

s = 0;

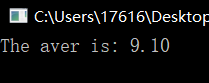
for ( i=0; i<10; i++)

s += a[i];

aver = s / i;

printf("The aver is: %.2f\n", aver);

}



2、求二维数组 a 中的最大值和最小值。 例如，当二维数组 a 中的元素为： 4 4 34 37 3 12 5 6 5 程序的输出应为：The max is: 37 The min is: 3 。

#include <conio.h>

#include <stdio.h>

int main()

{

int a[3][3]={4,4,34,37,3,12,5,6,5},i,j,max,min;

max = min = a[0][0];

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

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

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

{

if ( max < a[i][j] )

max = a[i][j];

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

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

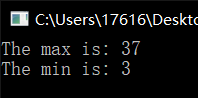
min = a[i][j];

}

printf("The max is: %d\n", max);

printf("The min is: %d\n", min);

}



3、求一维数组 a 中的最大元素及其下标。 例如，当一维数组 a 中的元素为：1,4,2,7,3,12,5,34,5,9，

- 18 -

程序的输出应为：The max is: 34,pos is: 7 。

#include <conio.h>

#include <stdio.h>

int main()

{

int a[10]={1,4,2,7,3,12,5,34,5,9},i,max,pos;

max = a[0];

pos = 0;

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

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

if (max < a[i])

{

max = a[i];

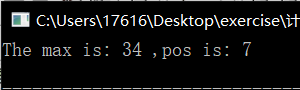
/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

pos = i;

}

printf("The max is: %d ,pos is: %d\n", max , pos);

}



4、求二维数组 a 中的最小值。 例如，当二维数组 a 中的元素为： 4 2 34 7 3 12 5 6 5 程序的输出应为：The min is: 2 。·

#include <conio.h>

#include <stdio.h>

int main()

{

int a[3][3]={4,2,34,7,3,12,5,6,5},i,j,min;

min = a[0][0];

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

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

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

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

{

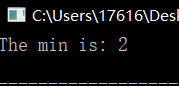
/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

min = a[i][j];

}

printf("The min is: %d\n", min);

}



5、求一维数组 a 中所有元素的平均值。 例如，当一维数组 a 中的元素为：10,4,2,7,3,12,5,34,5,9 程序的输出应为：The aver is: 9.10 。

#include <conio.h>

#include <stdio.h>

int main()

{

int a[10]={10,4,2,7,3,12,5,34,5,9},i;

double aver,s;

s = 0;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

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

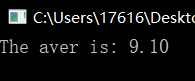
/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

s = s + a[i];

aver = s / i;

printf("The aver is: %.2f\n", aver);

}



6、输入一个百分制成绩，打印出五级记分成绩。考试成绩在 90 分或 90 分以上为优秀，80～89 分为 良好，70～79 为中等，60～69 为及格，低于 60 分为不及格。

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int score,t;

printf("Please enter a score:");

do

{

scanf("%d",&score);

}while(score<0||score>100);

t=score/10;

/\*\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

switch(t)

{

case 10:

case 9:printf("优秀!\n");

break;

case 8:printf("良好!\n");

break;

case 7:printf("中等!\n");

break;

case 6:printf("及格!\n");

break;

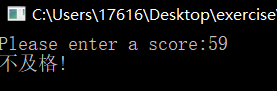
/\*\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

default:

printf("不及格!\n");

}

}



7、输出 100～200 之间既不能被 3 整除也不能被 7 整除的整数并统计这些整数的个数，要求每行输出 8 个数。

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int i;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

int n = 0;

for(i=100;i<=200;i++)

{

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

if(i%3!=0&&i%7!=0)

{

if(n%8==0) printf("\n");

printf("%6d",i);

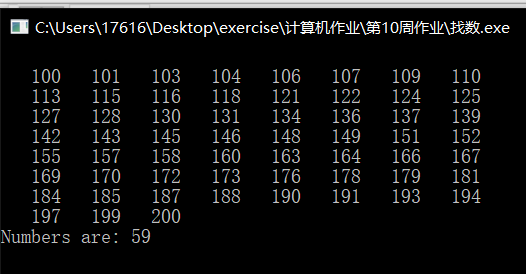
n++;

}

}

printf("\nNumbers are: %d\n",n);

}



8、学习优良奖的条件如下：所考 5 门课的总成绩在 450 分（含）以上；或者每门课都在 88 分（含） 以上。输入某学生 5 门课的考试成绩，输出是否够学习优良奖的条件。

#include <stdio.h>

int main()

{

int score,sum=0;

int i,n=0;

for(i=1;i<=5;i++)

{

scanf("%d",&score);

sum+=score;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

if(score>=88)

n++;

}

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

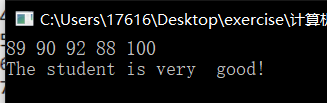
if(sum>=450 || n==5 )

printf("The student is very good!\n");

else

printf("The student is not very good!\n");

}



9、输出 200～300 之间的所有素数，要求每行输出8 个素数。

#include <stdio.h>

#include <math.h>

int main()

{

int m,j,n=0,k;

for(m=200;m<=300;m++)

{

k=sqrt(m);

for(j=2;j<=k;j++)

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

if(m%j==0) break;

if(j<=k)

{

if(n%8==0)

printf("\n");

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

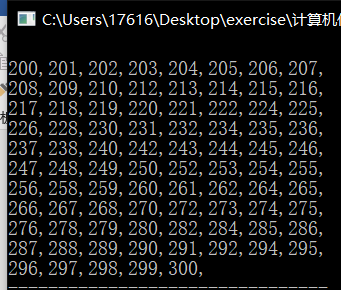
printf("%d,",m);

n++;

}

}

}



10、求出 a 所指数组中最小数(规定最小数不在 a[0]中)，最小数和 a[0]中的数对调。 例如数组中原有的数为：7、10、12、0、3、6、9、11、5、8, 输出的结果为：0、10、12、7、3、6、9、11、5、8。

#include <conio.h>

#include <stdio.h>

#define N 20

int main()

{

int a[N]={7,10,12,0,3,6,9,11,5,8}, n=10, i, k,m,min,t;

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

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

printf("\n");

min= a[0];

m=0;

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

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

if (a[k]<min )

{

min = a[k];

m = k;

}

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

t = a[0];

a[0]=a[m];

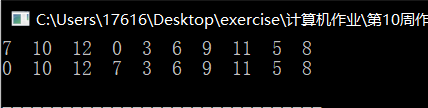
a[m] = t;

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

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

printf("\n");

}



11、求一维数组 a 中的最小元素及其下标。 例如，当一维数组 a 中的元素为：1,4,2,7,3,12,5,34,5,9， 程序的输出应为：The min is: 1,pos is: 0 。

#include <conio.h>

#include <stdio.h>

int main()

{

int a[10]={1,4,2,7,3,12,5,34,5,9},i,min,pos;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

min = a[0];

pos = 0;

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

if (min > a[i])

{

min = a[i];

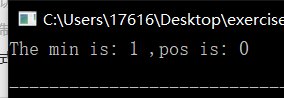
/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

pos = i;

}

printf("The min is: %d ,pos is: %d\n", min , pos);

}



12、求一维数组 a 中值为偶数的元素之和。 例如，当一维数组 a 中的元素为：10,4,2,7,3,12,5,34,5,9 ， 程序的输出应为：The result is: 62。

#include <conio.h>

#include <stdio.h>

int sum ( int arr[ ],int n )

{

int i,s;

s = 0;

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

if (arr[i] % 2 == 0)

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

s = s + arr[i];

return (s);

}

int main()

{

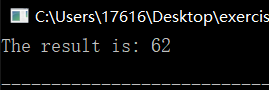
int a[10]={10,4,2,7,3,12,5,34,5,9},s;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

s = sum(a ,10);

printf("The result is: %d\n", s);

}



13、求一维数组 a 中的最大元素及其下标。 例如，当一维数组a 中的元素为：1,4,2,7,3,12,5,34,5,9， 程序的输出应为：The max is: 34,pos is: 7 。

#include <conio.h>

#include <stdio.h>

int main()

{

int a[10]={1,4,2,7,3,12,5,34,5,9},i,max,pos;

max = a[0];

pos = 0;

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

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

if (max < a[i])

{

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

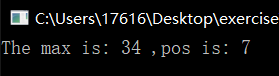
max = a[i];

pos =i;

}

printf("The max is: %d ,pos is: %d\n", max , pos);

}



14、求一维数组 a 中值为奇数的元素之和。 例如，当一维数组 a 中的元素为：10,4,2,7,3,12,5,34,5,9, 21 ,19 程序的输出应为：The result is: 69。

#include <conio.h>

#include <stdio.h>

int sum( int b[ ],int n )

{

int i,s = 0;

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

if (b[i] % 2 == 1)

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

s = s + b[i];

return (s);

}

int main()

{

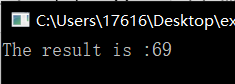
int a[12]={10,4,2,7,3,12,5,34,5,9,21,19},n;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

n = sum(a,12);

printf("The result is :%d\n",n );

}



15、求一维数组 a 中的最大元素及其下标。 例如，当一维数组 a 中的元素为：34,4,2,7,3,12,5,8,5,9， 程序的输出应为：The max is: 34,pos is: 0 。

#include <conio.h>

#include <stdio.h>

int max;

int maxarr(int arr[ ])

{

int pos,i;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

max = arr[0];

pos = 0;

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

if (max < arr[i])

{

max = arr[i];

pos = i;

}

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

return pos;

}

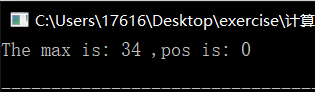
int main()

{

int a[10]={34,4,2,7,3,12,5,8,5,9};

printf("The max is: %d ,pos is: %d\n", max , maxarr(a));

}



1. 填空
2. 求二分之一的圆面积，函数通过形参得到圆的半径，函数返回二分之一的圆面积（注意：圆面积公式 为：S=3.14159\*r\*r，在程序中定义的变量名要与公式的变量相同）。例如，输入圆的半径值：2.5，输出为 s=9.817469。

#include <stdio.h>

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

double fun ( float r)

{

return 3.14159 \* r\*r/2.0 ;

}

int main()

{

float x;

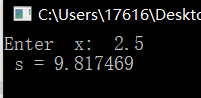
printf ( "Enter x: ");

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

scanf ( "%f", &x);

printf (" s = %f\n ", fun ( x ) );

}



2、计算并输出下列级数的前n 项之和Sn，直到 Sn 大于 q 为止，q 的值通过形参传入。 Sn = 2/1 + 3/2 + 4/3 + …… + (n+1)/n 例如，若 q 的值为 50.0，则函数值为50.416691。

#include <stdio.h>

float fun( float q )

{

int n;

float s;

n = 2;

s = 2.0;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

while (s <= q)

{

s=s+(float)(n+1)/n;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

n++;

}

return s;

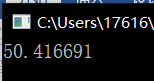
}

int main()

{

printf("%f\n", fun(50));

}



3、统计整数n 的各个位上出现数字1、2、3 的次数，并通过外部（全局）变量 c1、c2、c3 返回主函数。

例如，当 n=123114350 时，结果应该为：c1=3 c2=1 c3=2。

#include <stdio.h>

int c1,c2,c3;

void fun(long n)

{

c1 = c2 = c3 = 0;

while (n)

{

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

switch(n % 10)

{

case 1: c1++;

break;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

case 2: c2++;

break;

case 3: c3++;

}

n /= 10;

}

}

int main()

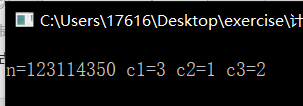
{

int n=123114350;

fun(n);

printf("\nn=%d c1=%d c2=%d c3=%d\n",n,c1,c2,c3);

}



4、程序的功能是计算 y = 0! + 1! + 2! + 3! + 4! + …… + n! 如输入 n 的值为 5 的话,则输出 y值为 154

#include <stdio.h>

int fun(int n)

{

int i;

int s;

s=1;

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

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

s = s \* i;

return s;

}

int main()

{

int s;

int k,n;

scanf("%d",&n);

s=0;

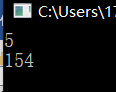
for (k=0; k<=n; k++)

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

s = s + fun(k);

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

}



5、计算并输出下列多项式的值。 S = 1 + 1/(1+2) + 1/(1+2+3) + …… + 1/(1+2+3+……+50) 例如，若主函数从键盘给n 输入 50 后，则输出为S=1.960784。

#include <stdio.h>

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

double fun(int n)

{

int i,j;

double sum=0.0, t;

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

{

t=0.0;

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

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

t += j;

sum+= 1.0/t;

}

return sum;

}

int main()

{

int n;

double s;

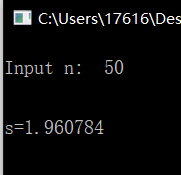
printf("\nInput n: ");

scanf("%d",&n);

s=fun(n);

printf("\n\ns=%f\n\n",s);

}



6、计算两个整数n 和m（m<1000）之间所有数的和。 n和m 从键盘输入。例如，当n=1，m=100 时，sum=5050， 当n=100，m=1000 时，sum=495550。

#include <stdio.h>

#include <conio.h>

int main()

{

int n,m;

int sum;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

sum = 0;

printf("\nInput n,m\n");

scanf("%d,%d",&n,&m);

while( n<=m )

{

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

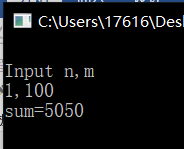
sum += n;

n++;

}

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

}



7、计算 N\*N 矩阵的主对角线元素和副对角线元素之和，并作为函数值返回。(要求：先累加主对角线元 素中的值，然后累加副对角线元素中的值。) 例如，若 N=3，有下列矩阵： 1 2 3 4 5 6 7 8 9 fun 函数首先累加1、5、9，然后累加3、5、7，函数的返回值为30。

#include <stdio.h>

#define N 3

int fun(int t[][N], int n)

{

int i, sum;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

sum = 0;

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

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

sum+=t[i][i];

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

sum+= t[i][n-i-1] ;

return sum;

}

int main()

{

int t[][N]={1,2,3,4,5,6,7,8,9},i,j;

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

{

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

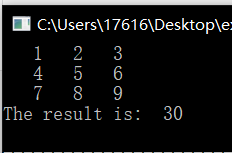
printf("%4d",t[i][j]);

printf("\n");

}

printf("The result is: %d\n",fun(t,N));

}



8、打印出 1 至1000 中满足其个位数字的立方等于其本身的所有整数。本题的结果为：1 64 125 216 729。

#include <stdio.h>

int main()

{

int i,g;

for(i=1;i<1000;i++)

{

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

g=i % 10;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

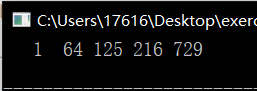
if(g \* g \* g == i)

printf("%4d",i);

}

printf("\n");

}



9、把数组 a（大小为M）中前M-1 个元素中的最小值放入a 的最后一个元素中

#include <stdio.h>

#define M 11

int main()

{

int a[M],i;

for(i=0;i<M-1;i++)

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

a[M-1]=a[0];

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

for(i=1; i < M - 1;i++)

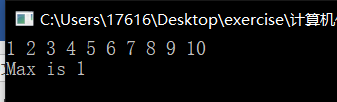
/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

if(a[i] < a[M - 1])

a[M-1]=a[i];

printf("Max is %d\n",a[M-1]);

}



10、统计一维数组a 中质数的个数。

- 29 -

例如：如果数组 a 的元素为：2,3,5,7,8,9,10,11,12,13， 则程序的输出应为：prime number(s) is(are): 6。

#include <math.h>

#include <conio.h>

#include <stdio.h>

int prinum( int a[])

{

int count,i,j,k;

count = 0;

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

{

k=a[i]-1;

for ( j=2; j<=k; j++)

if (a[i] % j == 0)

break;

if(j >= k+1)

count++;

}

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

return count;

}

int main()

{

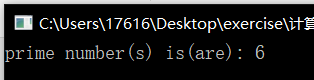
int a[10]={2,3,5,7,8,9,10,11,12,13},n;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

n = prinum(a);

printf("prime number(s) is(are): %d\n", n );

}



11、求一维数组 a 中质数之和。 例如：如果数组 a 的元素为：2,3,5,7,8,9,10,11,12,13,15,17, 则程序的输出应为：Sum is: 58。

#include <math.h>

#include <conio.h>

#include <stdio.h>

int main()

{

int a[12]={2,3,5,7,8,9,10,11,12,13,15,17},i,j,k,s;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

s = 0;

for (i=0; i<12; i++)

{

k=(int)sqrt(a[i]);

for (j=2;j <= k; j++)

if (a[i] % j == 0)

break;

if( j>k )

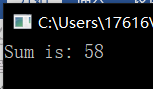
/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

s=s+a[i];

}

printf("Sum is: %d\n",s);

}



12、从键盘上输入两个正整数x,y，求它们的最大公约数。 例如：如果从键盘上输入24,36， 程序的输出应为：max is : 12。

#include <math.h>

#include <conio.h>

#include <stdio.h>

int main()

{

int x,y,t,i;

printf("Please enter two numbers:");

scanf("%d,%d",&x,&y);

if(x < y)

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

{

t = x;

x = y ;

y = t;

}

t = x % y;

while( t )

{

x = y;

y=t;

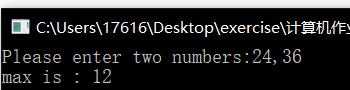
t = x % y;

}

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

printf("max is : %d\n",y);

}



13、求一维数组 a 中素数之和。 例如：如果数组 a 的元素为：2,3,5,7,8,9,10,11,12,13,15,17, 则程序的输出应为：Sum is: 54。

#include <math.h>

#include <conio.h>

#include <stdio.h>

int main()

{

int a[12]={2,3,5,7,8,9,10,11,12,13,15,17},i,j,k,s;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

s = 0;

for ( i=0; i<12; i++)

{

k=sqrt(a[i]);

for ( j=2;j <= k; j++)

if (a[i] % j == 0)

break;

if( j<=k )

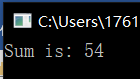
/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

s=s+a[i];

}

printf("Sum is: %d\n",s);

}



14、查找 n 在数组 a 中最后一次出现的位置(数组首元素的位置为零)。 例如：如果 a 数组中的元素为：1,5,2,5,6,8,7,4,3,0， 当n=5 时，程序的输出结果为：5 is No.3 。 当n=10 时，程序的输出结果应为：10 not found !。

#include <stdio.h>

int main()

{

int a[10]={1,5,2,5,6,8,7,4,3,0};

int i,k,n,f=0;

scanf("%d",&n);

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

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

if(a[i] == n)

{

f=1;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

k = i;

}

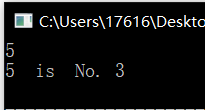
if(f)

printf("%d is No. %d\n", n,k);

else

printf(" %d not found !\n",n);

}



15、求两个正整数x,y的最大公约数和最小公倍数。 例如：如果x=24,y=36，程序的输出应为：

#include <math.h>

#include <conio.h>

#include <stdio.h>

int main()

{

int x,y,t,max,min,i,n1,n2;

printf("Please enter two numbers:");

scanf("%d,%d",&x,&y);

if(x > y)

{

t = x; x = y; y = t;

}

n1=x;

n2=y;

t = n2 % n1;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

while(t)

{

n2 = n1 ;

n1 = t ;

t = n2 % n1;

}

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

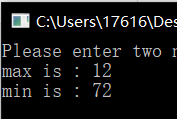
max = n1;

min = x \* y / max ;

printf("max is : %d\n",max);

printf("min is : %d\n",min);

}



1. 设计

1、将字符串中所有的大写字母转换为小写，其它字符不变（不使用转换函数）。 例如，当字符串为"This Is a c Program" 输出："this is a c program"

#include <stdio.h>

void fun(char str1[])

{

/\*\*\*\*\*\*\*\*\*\*\*begin\*\*\*\*\*\*\*\*\*\*\*/

int i = 0;

while(str1[i] != '\0')

{

if(str1[i] > 'A' && str1[i] < 'Z')

{

str1[i] = str1[i] + ' ';

}

i++;

}

/\*\*\*\*\*\*\*\*\*\*\*\*end\*\*\*\*\*\*\*\*\*\*\*\*/

}

int main()

{

char str1[80];

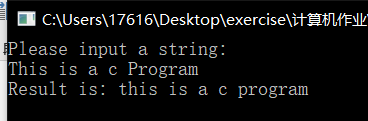
printf("Please input a string:\n");

gets(str1);

fun(str1);

printf("Result is: %s\n",str1);

}



1. 求[1，1000]之间既不能被 7 整除也不能被 5 整除的整数之和，将结果存入变量 s 中。

#include "stdio.h"

#include "math.h"

#include "stdlib.h"

int main()

{

int s;

int i;

/\*\*\*\*\*\*\*\*\*\*\*begin\*\*\*\*\*\*\*\*\*\*\*/

s = 0;

for(i = 1; i <= 1000; i++)

{

if(i % 5 != 0 && i % 7 != 0)

{

s += i;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*end\*\*\*\*\*\*\*\*\*\*\*\*/

printf("s=%d\n",s);

}

