编程题

### 1 题目描述

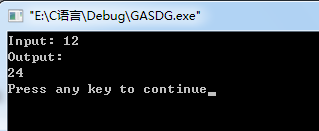
从键盘输入一个值给变量a，计算进行如下赋值表达式计算之后a的值：  
a+=a

### 输入格式

只有一行，包含一个整数a。

### 输出

输出题目描述中表达式运算后a的值。注意行尾输出换行。



#include<stdio.h>

int main(){

int a;

printf("Input: ");

scanf("%d",&a);

a+=a;

printf("Output:\n");

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

return 0;

}

### 2 题目描述

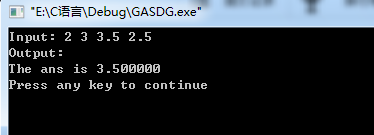
从键盘输入a,b,x,y的值，计算以下算术表达式的值：  
(double)(a+b)/2+(int)x%(int)y

### 输入格式

只有一行，包含四个数a，b，x，y，用空格隔开。其中x和y是浮点数，a和b是整数。

### 输出

输出题目描述中表达式的值并保留6位小数。注意行尾输出换行。



#include<stdio.h>

int main(){

int a,b;

double x,y,c;

printf("Input: ");

scanf("%d%d%lf%lf",&a,&b,&x,&y);

c=(double)(a+b)/2+(int)x%(int)y ;

printf("Output:\n");

printf("The ans is %lf\n",c);

return 0;

}

### 3 题目描述

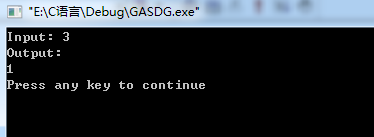
从键盘输入一个值给变量a，计算进行如下赋值表达式计算之后a的值：  
a-=2

### 输入格式

只有一行，包含一个整数a。

### 输出

输出题目描述中表达式运算后a的值。注意行尾输出换行。



#include<stdio.h>

int main(){

int a;

printf("Input: ");

scanf("%d",&a);

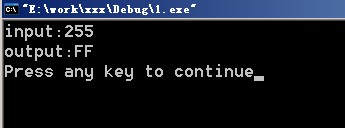
printf("Output:\n");

a-=2;

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

return 0;

}

4 从键盘输入一个10进制数，输出该数的16进制数。如图所示：其中255和20是键盘输入。

#include<stdio.h>

int main(){

int a;

printf("input:");

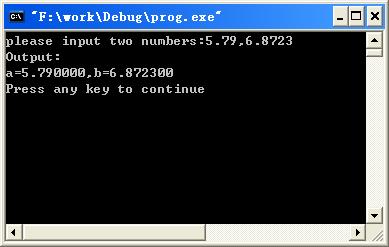
scanf("%d",&a);

printf("output:%X\n",a);

return 0;

}

5 输入实型数据a,b，然后输出a、b的值。  
　　程序的运行效果应类似地如图1所示，图1中的5.79,6.8723是从键盘输入的内容。



#include<stdio.h>

int main(){

double a,b;

printf("please input two numbers:");

scanf("%lf,%lf",&a,&b);

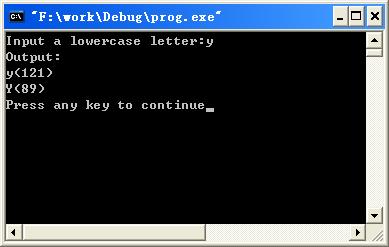
printf("Output:\n");

printf("a=%lf,b=%lf\n",a,b);

return 0;

}

输入一个小写英文字母，首先输出它及其ASCII码，然后输出其对应的大写字母及其ASCII码。编程可用素材：printf("Input a lowercase letter: ")...、printf("\n...(...)...。  
　　程序的运行效果应类似地如图1所示，图1中的y是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

char ch;

printf("Input a lowercase letter:");

ch = getchar();

printf("Output:\n");

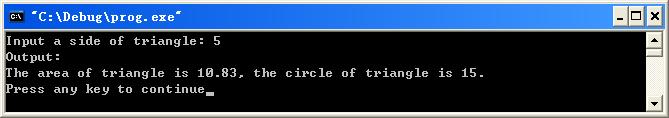
printf("%c(%d)\n", ch, ch);

printf("%c(%d)\n", ch - 32, ch - 32);

return 0;

}

从键盘输入一个正三角形的边长（整型），计算该三角形的面积和周长。注：根据海伦－秦九韶公式，area＝sqrt(p(p-a)(p-b)(p-c))，其中p＝(a+b+c)/2、a,b,c为三角形的3条边长。  
程序的运行效果应类似地如图1所示，图1中的5是从键盘输入的内容。



#include <stdio.h>

#include <math.h>

int main(void)

{

int side;

printf("Input a side of triangle: ");

scanf("%d", &side);

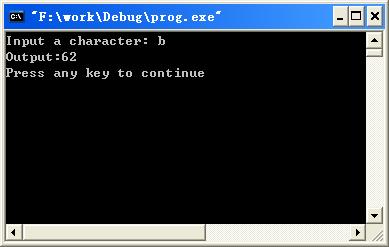
printf("Output:\n");

printf("The area of triangle is %.2f, the circle of triangle is %.0f.\n", 1.0 \* side \* side \* sqrt(3) / 4, 3.0 \* side);

return 0;

}

从键盘读入一个任意字符，输出该字符ASCII的十六进制值（小写）。  
程序的运行效果应类似地如图1所示，图1中的b是从键盘输入的内容



#include <stdio.h>

int main(void)

{

char ch;

printf("Input a character: ");

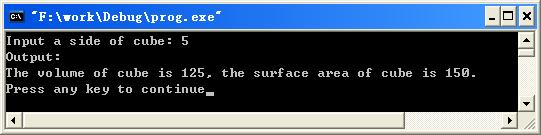
ch = getchar();

printf("Output:%x\n", ch);

return 0;

}

从键盘输入一个正方体的边长（整型），计算该正方体的体积和表面积。  
　　程序的运行效果应类似地如图1所示，图1中的5是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int side;

printf("Input a side of cube: ");

scanf("%d", &side);

printf("Output:\nThe volume of cube is %d, the surface area of cube is %d.\n", side \* side \* side, 6 \* side \* side);

return 0;

}

从键盘输入x,y,z的值，编写程序输出以下表达式的值：  
        x+z%3\*(int)(x+y)%2/4  
　　程序的运行效果应类似地如图1所示，图1中的2.5,4.7,7是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int float\_z;

double float\_x, float\_y;

printf("please input x,y,z:");

scanf("%lf,%lf,%d", &float\_x, &float\_y, &float\_z);

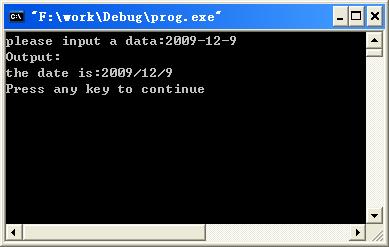
printf("Output:");

printf("%f\n", float\_x + float\_z % 3 \* (int)(float\_x + float\_y) % 2 / 4);

return 0;

}

从键盘输入一日期，年月日之间以“-”分隔，并以同样的形式但以“/”作分隔符输出。  
　　程序的运行效果应类似地如图1所示，图1中的2009-12-9是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int year, month, day;

printf("please input a data:");

scanf("%d-%d-%d", &year, &month, &day);

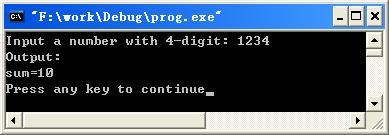
printf("Output:\n");

printf("the date is:%d/%d/%d\n", year, month, day);

return 0;

}

从键盘上输入一个四位整数，计算各个位上的数字之和。编程可用素材：printf("Input a number with 4-digit: ")...、printf("Output:\nsum=...\n"...。  
　　程序的运行效果应类似地如图1所示，图1中的1234是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int number;

printf("Input a number with 4-digit: ");

scanf("%d", &number);

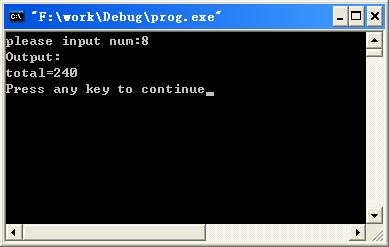
printf("Output:\n");

printf("sum=%d\n", number / 1000 + number / 100 % 10 + number / 10 % 10 + number % 10);

return 0;

}

已知某产品单价是30 ，输入其数量是num（整型），输出其总价（整型）。  
　　程序的运行效果应类似地如图1所示，图1中的8是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int num;

printf("please input num:");

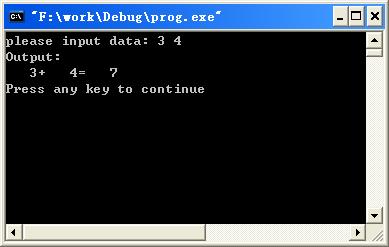
scanf("%d", &num);

printf("Output:\ntotal=%d\n", 30 \* num);

return 0;

}

输入两个整数，输出这两个整数的和（每个数占4位宽，+和=左右均无空格）。  
　　程序的运行效果应类似地如图1所示，图1中的3 4是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int number1, number2;

printf("please input data: ");

scanf("%d%d", &number1, &number2);

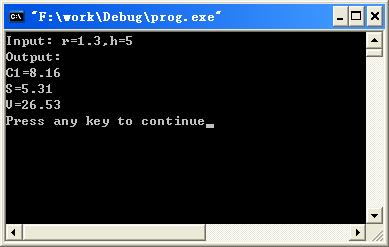
printf("Output:\n");

printf("%4d+%4d=%4d\n", number1, number2, number1 + number2);

return 0;

}

用scanf输入圆半径r，圆柱高h，求圆周长C1(＝2πr)、圆面积S(＝πr2)、圆柱体积V(＝πr2h)。(注意：本题中规定圆周率取值为3.14)  
　　程序的运行效果应类似地如图1所示，图1中的r=1.3,h=5是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

double rad, high, c1, s, ver;

printf("Input: ");

scanf("r=%lf,h=%lf", &rad, &high);

c1 = 2 \* rad \* 3.14;

s = 3.14 \* rad \* rad;

ver = s \* high;

printf("Output:\n");

printf("C1=%.2f\n", c1);

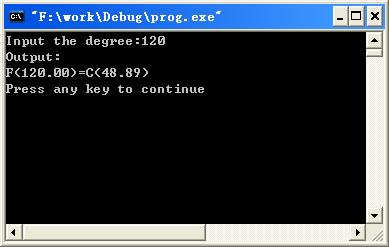
printf("S=%.2f\n", s);

printf("V=%.2f\n", ver);

return 0;

}

输入一个华氏温度，要求输出摄氏温度，计算公式为c=5(F-32)/9。编程可用素材：printf("Input the degree:")...、printf("\nF(...)=C(...)...。  
　　程序的运行效果应类似地如图1所示，图1中的120是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

double cel, fra;

printf("Input the degree:");

scanf("%lf", &fra);

cel = 5 \* (fra - 32) / 9;

printf("Output:\n");

printf("F(%.2f)=C(%.2f)\n", fra, cel);

return 0;

}

### 题目描述

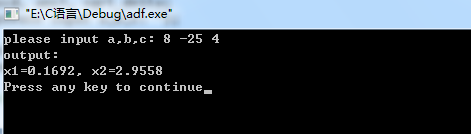
求方程式ax2+bx+c=0的根。

### 输入格式

三个用空格隔开的整数a、b、c，保证a不为0。

### 输出

方程式ax2+bx+c=0的根，若有两个不同的实根，则输出这两个实根，用空格隔开；若有一个实根，请直接输出；若没有实根，请输出“no answer”（请注意不需要输出引号）。所有的实数输出请使用C语言的默认舍入方式保留4位小数，注意行尾输出换行。



#include<stdio.h>

#include<math.h>

int main(){

int a,b,c,d;

double x1,x2;

printf("please input a,b,c: ");

scanf("%d %d %d",&a,&b,&c);

d=b\*b-4\*a\*c;

printf("output:\n");

if(d==0){

x1=x2=(-b)/(2.0\*a);

printf("x=%.4lf\n",x1);

}

else if(d<0)

printf("no answer\n");

else{

x1=((-b)-sqrt(d))/(2.0\*a);

x2=((-b)+sqrt(d))/(2.0\*a);

printf("x1=%.4lf, x2=%.4lf\n",x1,x2);

}

return 0;

}

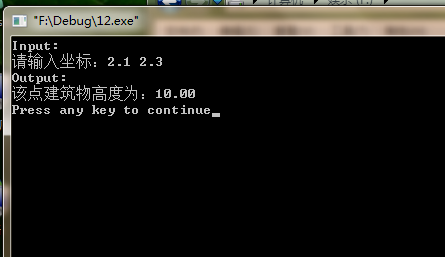
有4个圆塔，圆心分别为(2，2)，(-2，2)，(2，-2)，(-2，-2)，圆半径为1。这4个塔的高度都是10m，塔以外无建筑物。现在输入任一点的坐标，求该点的建筑高度（塔外的高度为0）。

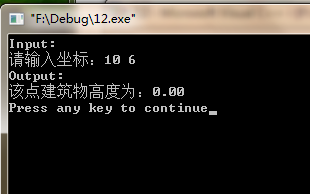
### 输入格式

两个用空格分隔的实数，表示输入的点坐标。

### 输出

输入坐标位置的建筑高度，保留2位小数。  
请注意行尾输出换行。





#include <stdio.h>

int main() {

int h = 10;

float x1=2,y1=2,x2=-2,y2=2,x3=-2,y3=-2,x4=2,y4=-2;

float x,y,d1,d2,d3,d4;

printf("Input:\n请输入坐标：");

scanf("%f %f", &x, &y); /\*求该点到各中心点的距离\*/

d1=(x-x1) \* (x-x1)+(y-y1) \* (y-y1);

d2=(x-x2) \* (x-x2)+(y-y2) \* (y-y2);

d3=(x-x3) \* (x-x3)+(y-y3) \* (y-y3);

d4=(x-x4) \* (x-x4)+(y-y4) \* (y-y4);

if (d1>1 && d2>1 && d3>1 && d4>1) h=0; /\*判断该点是否在塔外\*/

printf("Output:\n该点建筑物高度为：%.2f\n", h + .0);

return 0;

}

### 题目描述

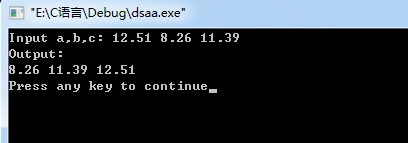
读入3个实数，按照代数值由小到大的顺序输出这3个数。

### 输入格式

3个用空格隔开的实数。

### 输出

按由小到大的顺序输出这3个实数，用空格隔开，并保留两位小数。  
请注意行尾输出换行。



#include <stdio.h>

int main() {

float a, b, c, t;

printf("Input a,b,c: ");

scanf("%f %f %f", &a, &b, &c);

printf("Output:\n");

if (a > b) {

t = a; a = b; b = t;

}

if (a > c) {

t = a; a = c; c = t;

}

if (b > c) {

t = b; b = c; c = t;

}

printf("%.2f %.2f %.2f\n", a, b, c);

return 0;

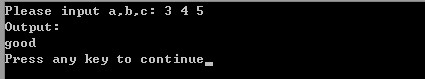
}

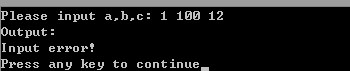
给你三角形的三条边，你能告诉我它是哪种三角形吗？  
如果是直角三角形，请输出“good”。如果是等腰三角形，请输出“perfect”。否则，请输出“just a triangle”。  
题目保证输入数据合法。

### 输入格式 每组样例包含了三个整数a,b,c，代表了三角形的三条边的长度。（0<a,b,c<300）

### 输出

对于每组样例，输出结果，每组结果占一行。





#include<stdio.h>

int main()

{

int t,a,b,c;

printf("Please input a,b,c: ");

scanf("%d%d%d",&a,&b,&c);

printf("Output:\n");

if(a+b>c && a+c>b && b+c>a)

{

if(a\*a+b\*b==c\*c||a\*a+c\*c==b\*b||b\*b+c\*c==a\*a)

printf("good\n");

else if(a==b||a==c||b==c)

printf("perfect\n");

else

printf("just a triangle\n");

}

else

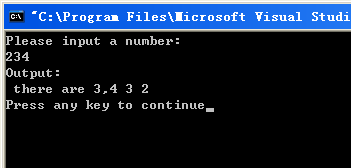
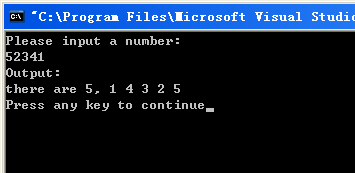
printf("Input error!\n");

return 0;

}

给一个不多于5位的正整数，要求：一、求它是几位数，二、逆序打印出各位数字。

**输出示例：**



#include <stdio.h>

int main()

{

long a, b, c, d, e, x;

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

scanf("%ld", &x);

a = x / 10000;

b = x % 10000 / 1000;

c = x % 1000 / 100;

d = x % 100 / 10;

e = x % 10;

printf("Output:\n");

if (a != 0)

printf("there are 5, %ld %ld %ld %ld %ld\n", e, d, c, b, a);

else if (b != 0)

printf("there are 4, %ld %ld %ld %ld\n", e, d, c, b);

else if (c != 0)

printf("there are 3, %ld %ld %ld\n", e, d, c);

else if (d != 0)

printf("there are 2, %ld %ld\n", e, d);

else if (e != 0)

printf("there are 1, %ld\n", e);

return 0;

}

### 题目描述

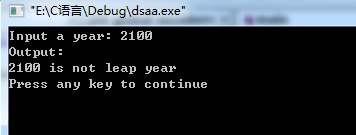
输入一个年份，判断是否是闰年。

### 输入格式

输入一个整数year，表示年份。

### 输出

如果输入的年份是闰年，请输出“leap year”，否则请输出“not leap year”。  
请注意不需要输出引号，行尾输出换行。



#include <stdio.h>

int main() {

int year;

printf("Input a year: ");

scanf("%d", &year);

printf("Output:\n");

if (year % 4 == 0) {

if (year % 100 == 0) {

if (year % 400 == 0) {

printf("%d isleap year\n",year);

} else {

printf("%d is not leap year\n",year);

}

} else {

printf("%d is leap year\n",year);

}

} else {

printf("%d is not leap year\n",year);

}

return 0;

}

### 题目描述

给定一个百分制的成绩，输出成绩等级’A’、 ’B’、 ’C’、 ’D’、 ’E’。90分以上为’A’，80至89分为’B’，70至79分为’C’，60至69分为’D’，60分一下为’E’。

### 输入格式

一个整数score，保证0<=score<=100.

### 输出

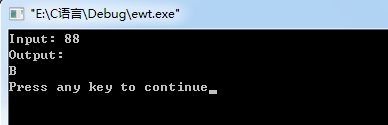
输出score对应的成绩等级。  
请注意行尾输出换行。

### 样例输入

88

### 样例输出

B



 #include <stdio.h>

int main() {

int score;

char level;

printf("Input: ");

scanf("%d", &score);

printf("Output:\n");

if (score >= 90)

level = 'A';

else if (score >= 80)

level = 'B';

else if (score >= 70)

level = 'C';

else if (score >= 60)

level = 'D';

else

level = 'E';

printf("%c\n", level);

return 0;

}

### 题目描述

运输公司对用户计算运费。路程s越远，每吨×千米运费越低。标准如下：

|  |  |
| --- | --- |
| s<250 | 没有折扣 |
| 250<=s<500 | 2%折扣 |
| 500<=s<1000 | 5%折扣 |
| 1000<=s<2000 | 8%折扣 |
| 2000<=s<3000 | 10%折扣 |
| 3000<=s | 15%折扣 |

设每吨每千米货物的基本运费为p（price的缩写），货物重为w（weight的缩写），距离为s，折扣为d（discount的缩写），则总运费f（freight的缩写）的计算公式为：

f=p×w×s×(1-d)

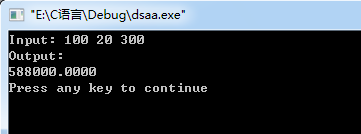
现在给定p、w、s，计算总运费f。

### 输入格式

三个用空格隔开的数p、w、s，其中p和w是实数，s是整数。

### 输出

输出总运费。  
请保留4位小数，注意行尾输出换行。



#include <stdio.h>

#include <math.h>

int main() {

int c, s;

float p, w, d, f;

printf("Input: ");

scanf("%f %f %d", &p, &w, &s);

printf("Output:\n");

if (s >= 3000) c = 12;

else c = s / 250;

switch (c) {

case 0: d = 0; break;

case 1: d = 2; break;

case 2:

case 3: d = 5; break;

case 4:

case 5:

case 6:

case 7: d = 8; break;

case 8:

case 9:

case 10:

case 11: d = 10; break;

case 12: d = 15; break;

}

f = p \* w \* s \* (1 - d / 100.0);

printf("%.4f\n", f);

return 0;

}

### 题目描述

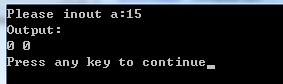
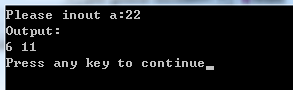
一个笼子里面关了鸡和兔子（鸡有2只脚，兔子有4只脚，没有例外）。已经知道了笼子里面脚的总数a，问笼子里面至少有多少只动物，至多有多少只动物。

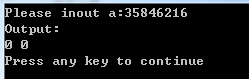
### 输入格式

测试数据占1行，每行一个正整数a (a < 32768)

### 输出

包含两个正整数，第一个是最少的动物数，第二个是最多的动物数，两个正整数用一个空格分开  
如果没有满足要求的答案，则输出两个0。





#include <stdio.h>

int main()

{

int a;

printf("Please inout a:");

scanf("%d",&a);

if(a%2!=0||a<=0||a>=32768)

{

printf("Output:\n0 0\n");

}

else

{

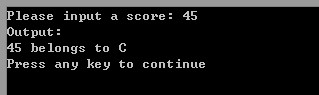
printf("Output:\n%d %d\n",(a%4)/2+a/4,a/2);

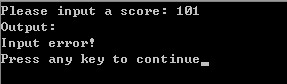
}

}

利用条件运算符的嵌套来完成此题:学习成绩>=90分的同学用A表示,60-89分之间的用B表示,60分以下的用C表示。提示：未嵌套语句为：Grade=score>=60?'B':'C';

示例：





#include<stdio.h>

int main()

{

int score;

char grade;

printf("Please input a score: ");

scanf("%d",&score);

printf("Output:\n");

if(score>100 || score<0)

{

printf("Input error!\n");

return 0;

}

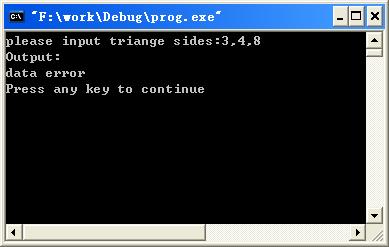
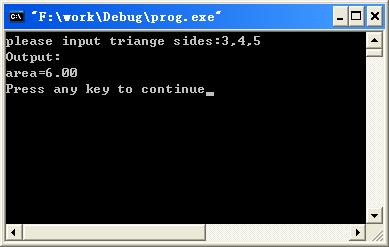
grade=score>=90?'A':(score>=60?'B':'C');

printf("%d belongs to %c\n",score,grade);

return 0;

}

输入三角形的三边长a、b、c(边长可以是小数)，求三角形面积area，并输出。如果输入的三边构不成三角形，应给出“data error”的信息提示。注：根据“海伦－秦九韶”公式，area＝√p(p-a)(p-b)(p-c)，其中p＝(a+b+c)/2。编程可用素材：printf("\nplease input triange sides:")...、printf("Output:\ndata error\n")...、printf("Output:\narea=...\n"...。  
　　程序的运行效果应类似地如图1和图2所示，图1中的3,4,5和图2中的3,4,8是从键盘输入的内容



#include <stdio.h>

#include <math.h>

int main(void)

{

double side1, side2, side3, circle, area;

printf("please input triange sides:");

scanf("%lf,%lf,%lf", &side1, &side2, &side3);

printf("Output:\n");

if (side1 + side2 > side3 && side1 + side3 > side2 && side2 + side3 > side1)

{

circle = (side1 + side2 + side3) / 2;

area = sqrt(circle \* (circle - side1) \* (circle - side2) \* (circle - side3));

printf("area=%.2f\n", area);

}

else

{

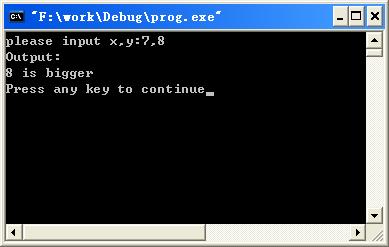
printf("data error\n");

}

return 0;

}

输入x,y两个整数，输出其中较大的数。  
　　程序的运行效果应类似地如图1所示，图1中的7,8是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int number1, number2, max;

printf("please input x,y:");

scanf("%d,%d", &number1, &number2);

if (number1 > number2)

{

max = number1;

}

else

{

max = number2;

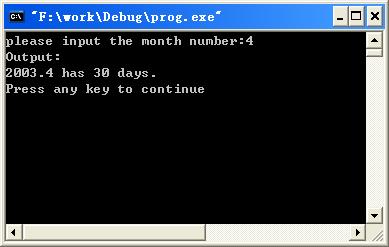
}

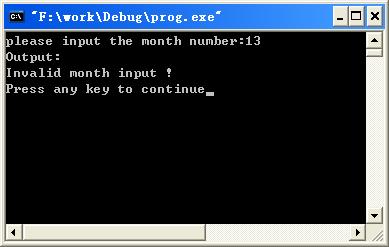
printf("Output:\n%d is bigger\n", max);

return 0;

}

输入月份，打印2003年该月有几天。当输入的月份超范围时，应显示“Invalid month input”。编程可用素材：printf("please input the month number:")...、printf("\nInvalid month input !\n")...、printf("\n 2003.... has ... days.\n"...。  
　　程序的运行效果应类似地如图1和图2所示，图1中的4和图2中的13是从键盘输入的内容





#include <stdio.h>

int main(void)

{

int month, day;

printf("please input the month number:");

scanf("%d", &month);

switch (month)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

day = 31;

break;

case 4:

case 6:

case 9:

case 11:

day = 30;

break;

case 2:

day = 28;

break;

default:

day = 0;

break;

}

printf("Output:\n");

if (day == 0)

{

printf("Invalid month input !\n");

}

else

{

printf("2003.%d has %d days.\n", month, day);

}

return 0;

}

编写程序实现以下功能  
　　从键盘输入3个整数，输出绝对值最大的数。编程可用素材：printf("Input 3 numbers: ")...、printf("\nThe number with maximum absolute value is ....\n"...。  
　　程序的运行效果应类似地如图1所示，图1中的1,-362,300是从键盘输入的内容。



#include <stdio.h>

#include <math.h>

int main(void)

{

int number1, number2, number3, max;

printf("Input 3 numbers: ");

scanf("%d,%d,%d", &number1, &number2, &number3);

if (abs(number1) > abs(number2))

{

max = number1;

}

else

{

max = number2;

}

if (abs(max) < abs(number3))

{

max = number3;

}

printf("Output:\nThe number with maximum absolute value is %d.\n", max);

return 0;

}

### 题目描述

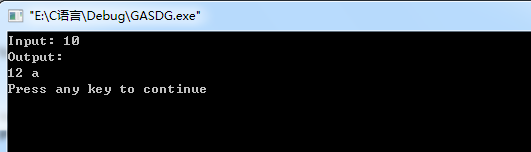
输入1个10进制整数，依次输出每一个数的八进制和十六进制表示。

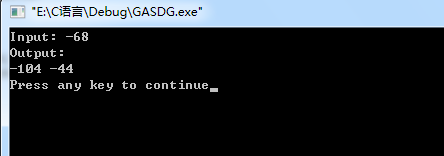
### 输入格式

每行包含一个整数。

### 输出

对于每一行输入，输出一行，包含相应数的八进制和十六进制表示，用空格隔开。





#include <stdio.h>

int main() {

int n;

printf("Input: ");

scanf("%d", &n);

printf("Output:\n");

if (n < 0) {

printf("-%o -%x\n", -n, -n);

} else {

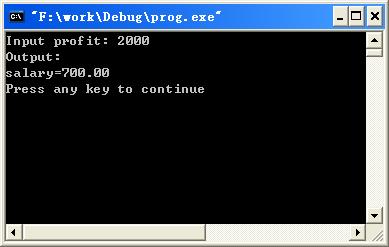
printf("%o %x\n", n, n);

}

return 0;

}

已知某公司员工的保底薪水为500，某月所接工程的利润profit（整数）与利润提成的关系如下（计量单位：元）：  
        profit<=1000　　　　　　　　没有提成；  
        1000＜profit<=2000　　　　　提成10%；  
        2000＜profit<=5000　　　　　提成15%；  
        5000＜profit<=10000　　　　　提成20%；  
        10000＜profit　　　　　　　　　提成25%。  
　　请根据输入的利润计算员工的薪水。编程可用素材：printf("Input  profit: ")...、printf("Output:\nsalary=...\n"...。  
　　程序的运行效果应类似地如图1所示，图1中的2000是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int profit;

double rate = 500;

printf("Input profit: ");

scanf("%d", &profit);

if (profit <= 1000)

{

;

}

else if (profit <= 2000)

{

rate += profit \* 0.1;

}

else if (profit <= 5000)

{

rate += profit \* 0.15;

}

else if (profit <= 10000)

{

rate += profit \* 0.2;

}

else

{

rate += profit \* 0.25;

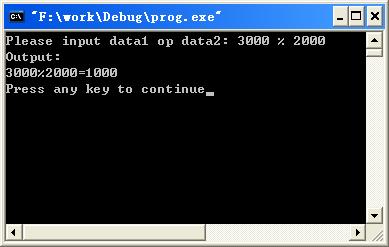
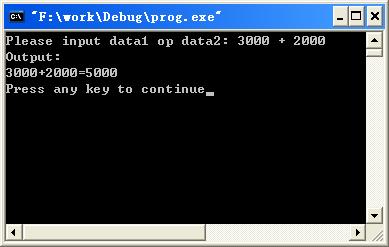
}

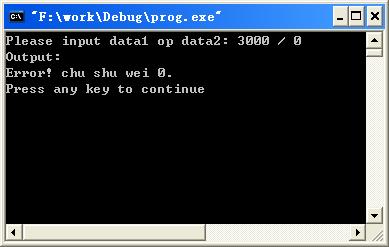
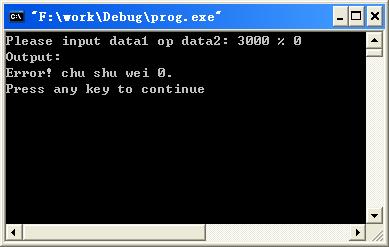
printf("Output:\nsalary=%.2f\n", rate);

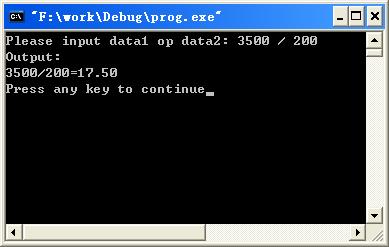
return 0;

}

编写一个简单计算器程序，输入格式为：data1 op data2。其中data1和data2是参加运算的两个数(data1、data2**必须定义为int**，但二者相加可能超出int能表示的范围)，op为运算符，它的取值只能是+、-、\*、/、%。输出：计算除法输出小数点后两位，其他运算均输出整数。编程可用素材：printf("Please input data1 op data2: ")...、printf("\nError! chu shu wei 0.\n")...。  
　　程序的运行效果应类似地如图1、图2、图3、图4、图5所示，图1中的3000 + 2000、图2中的3000 % 2000、图3中的3000 % 0、图4中的3000 / 0和图5中的3500 / 200是从键盘输入的内容。







#include <stdio.h>

int main(void)

{

int data1, data2, flag = 0;

char op;

double result;

printf("Please input data1 op data2: ");

scanf("%d %c %d", &data1, &op, &data2);

switch (op)

{

case '+':

result = (double)data1 + data2;

break;

case '-':

result = (double)data1 - data2;

break;

case '\*':

result = (double)data1 \* data2;

break;

case '/':

if (data2 == 0)

{

flag = 1;

break;

}

result = (double)data1 / data2;

break;

case '%':

if (data2 == 0)

{

flag = 1;

break;

}

result = data1 % data2;

break;

default:

break;

}

printf("Output:");

if (flag == 0)

{

if (op == '/')

{

printf("\n%d%c%d=%.2f\n", data1, op, data2, result);

}

else

{

printf("\n%d%c%d=%.0f\n", data1, op, data2, result);

}

}

else

{

printf("\nError! chu shu wei 0.\n");

}

return 0;

}

　x(x只考虑整数int且**必须定义为int**、但F(x)完全可能超过int的表示范围)通过键盘输入(输入前给出提示Please input x:)，然后计算并在屏幕上输出函数值。编程可用素材：printf("Please input x: ")...、printf("\nF(...) = ..."...。  
　　　　　┌　-5x+27   (x<0)  
　　F(x)= ├　7909     (x=0)  
　　　　　└　2x-1     (x>0)  
　　程序的运行效果应类似地如图1所示，图1中的3是从键盘输入的内容。输出F(3) = 5中使用的是小括号。



#include <stdio.h>

int main(void)

{

int x;

double y;

printf("Please input x:");

scanf("%d", &x);

if (x < 0)

{

y = -5.0\* x + 27;

}

else if (x == 0)

{

y = 7909;

}

else

{

y = 2.0 \* x - 1;

}

printf("Output:\n");

printf("F(%d) = %.0f\n", x, y);

return 0;

}

### 题目描述

输出介于a,b（含a,b，且a<=b）的所有“水仙花数”。所谓“水仙花数”是指一个数，其各位数字立方和等于该数本身。例如，153是一个“水仙花数”，因为153=13+53+33。

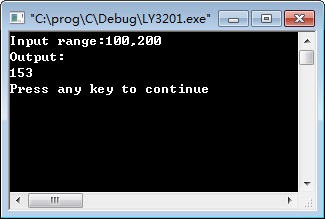
### 输入格式

100,200

### 输出

“水仙花数”，在每一个数之后输出一个空格。

请注意行尾输出换行。



#include <stdio.h>

int main() {

int i, a, b;

int n, k, sum;

printf("Input range:");

scanf("%d,%d", &a, &b);

printf("Output:\n");

for (i = a;i <= b;i++)

{

n = i;

sum = 0;

while (n != 0)

{

k = n % 10;

sum += k \* k \* k;

n = n /10;

}

if (sum == i)

{

printf("%d ", i);

}

}

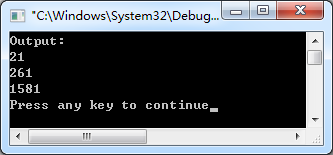
printf("\n");

return 0;

}

在10万以内判断，一个整数，它加上100后是一个完全平方数，再加上168又是一个完全平方数，请问该数是多少？

编程可用素材：printf("Output:\n");



#include<stdio.h>

#include<math.h>

int main()

{

long int i,x,y,z;

printf("Output:\n");

for (i=1;i<100000;i++)

{

x=sqrt(i+100);/\*x为加上100后开方后的结果\*/

y=sqrt(i+268);/\*y为再加上168后开方后的结果\*/

if(x\*x==i+100&&y\*y==i+268)/\*如果一个数的平方根的平方等于该数，这说明此数是完全平方数\*/

printf("%ld\n",i);

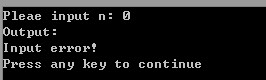
}

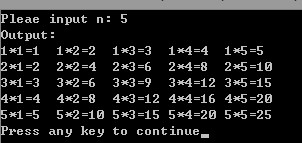
return 0;

}

输入：n(0<n<=9),输入越界时，提示"Input error!\n"并结束。

输出9\*9乘法表，它的积左对齐，占3位





#include "stdio.h"

int main()

{

int i,j,result,n;

printf("Pleae input n: ");

scanf("%d",&n);

printf("Output:\n");

if(n<=0 || n>9)

{

printf("Input error!\n");

return 0;

}

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

{

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

{

result=i\*j;

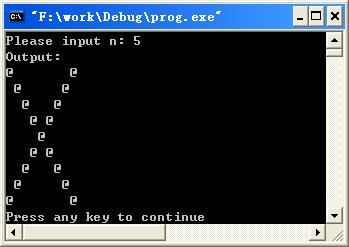
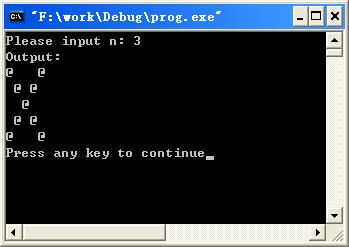
printf("%d\*%d=%-3d",i,j,result);/\*-3d表示左对齐，占3位\*/

} printf("\n");/\*每一行后换行\*/

}

return 0;

}

根据输入的n(约定n>0)在屏幕上显示对应的图案。程序的运行效果应类似地如图1和图2所示，图1中的3和图2中的5是从键盘输入的内容。

#include <stdio.h>

int main(void){

int i, j, n;

printf("Please input n: ");

scanf("%d", &n);

printf("Output:\n");

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

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

{

printf(" ");

}

if (i == n)

{

printf("@\n");

continue;

}

printf("@");

for (j = 1; j <= 2 \* (n - i) - 1; j++)

{

printf(" ");

}

printf("@");

printf("\n");

}

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

{

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

{

printf(" ");

}

printf("@");

for (j = 1; j <= 2 \* i - 1; j++)

{

printf(" ");

}

printf("@");

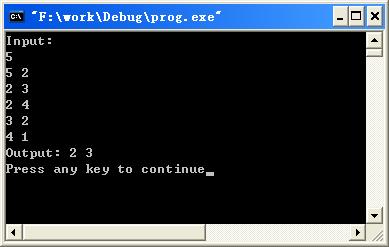
printf("\n");

}

return 0;

}

程序输入：第一行是一个正整数n(约定1<=n<=1000)，表示一共有n个点。接下来的n行，每行2个整数，分别表示每个点的横坐标和纵坐标。  
　　程序输出：输出横坐标最小的那个点的横坐标和纵坐标，如果横坐标最小的点有多个，则只输出其中纵坐标最小的那个点。  
　　程序的运行效果应类似地如图1所示，图1中的5、5 2、2 3、2 4、3 2和4 1是从键盘输入内容。



#include <stdio.h>

int main(void)

{

int i, n;

int x, y;

int minx, miny;

printf("Input:\n");

scanf("%d", &n);

scanf("%d%d", &minx, &miny);

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

{

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

if (minx > x)

{

minx = x;

miny = y;

}

else if (minx == x && miny > y)

{

minx = x;

miny = y;

}

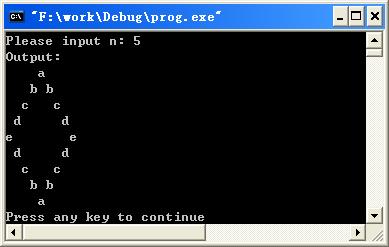
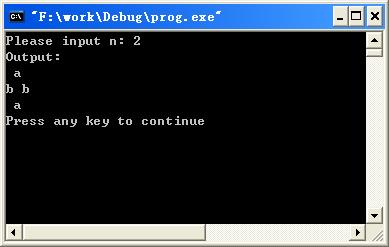
}

printf("Output: %d %d\n", minx, miny);

return 0;

}

根据输入的n(约定n>1)在屏幕上显示对应的图案。编程可用素材：printf("Please input n: ")...printf("Output:\n")...。  
　　程序的运行效果应类似地如图1和图2所示，图1中的2和图2中的5是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int i, j, n;

printf("Please input n: ");

scanf("%d", &n);

printf("Output:\n");

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

{

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

{

printf(" ");

}

if (i == 1)

{

printf("a\n");

continue;

}

printf("%c", 'a' + i - 1);

for (j = 1; j <= 2 \* i - 3; j++)

{

printf(" ");

}

printf("%c", 'a' + i - 1);

printf("\n");

}

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

{

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

{

printf(" ");

}

if (i == n - 1)

{

printf("a\n");

continue;

}

printf("%c", 'a' + n - i - 1);

for (j = 1; j <= 2 \* (n - i) - 3; j++)

{

printf(" ");

}

printf("%c", 'a' + n - i - 1);

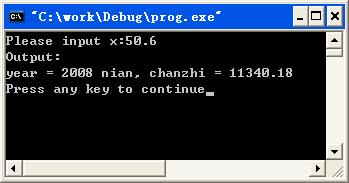
printf("\n");

}

return 0;

}

编写一程序实现以下功能  
　　设某企业2006年的产值为5000万，计划以后每年的增长率为x(x从键盘输入，例如输入8.75表示8.75%)，计算该企业的产值在哪年实现翻番以及翻番时的产值，然后输出(输出时以万为单位，应考虑有小数)。编程可用素材：printf("Please input x: ")...、printf("\nyear = ... nian, chanzhi = ...\n"...。  
　　程序的运行效果应类似地如图1所示，图1中的50.6是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int i, n = 2006;

double x;

double chanzhi = 5000;

printf("Please input x:");

scanf("%lf", &x);

for (i = 0; chanzhi < 10000; i++)

{

chanzhi = chanzhi \* (1 + x / 100);

}

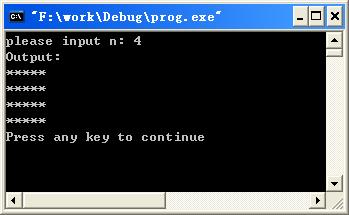
printf("Output:");

printf("\nyear = %d nian, chanzhi = %.2f\n", n + i, chanzhi);

return 0;

}

输出n行星号，每行5个\*星号。程序的运行效果应类似地如图所示，图中的4是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int n, i;

printf("please input n: ");

scanf("%d", &n);

printf("Output:\n");

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

{

printf("\*\*\*\*\*\n");

}

return 0;

}

### 题目描述

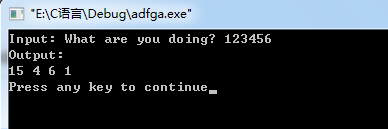
输入一行字符，分别统计出其中英文字母、空格、数字和其他字符的个数。

### 输入格式

一行字符。

### 输出

分别输出这行字符中的英文字母、空格、数字和其他字符的个数，用空格隔开。  
请注意行尾输出换行



#include <stdio.h>

#include <math.h>

int main() {

char c;

int letter, space, digit, other;

printf("Input: ");

letter = 0; space = 0; digit = 0; other = 0;

while ((c = getchar()) != '\n') {

if (('A' <= c && c <= 'Z') || ('a' <= c && c <= 'z'))

letter++;

else if (c == ' ')

space++;

else if ('0' <= c && c <= '9')

digit++;

else

other++;

}

printf("Output:\n");

printf("%d %d %d %d\n", letter, space, digit, other);

return 0;

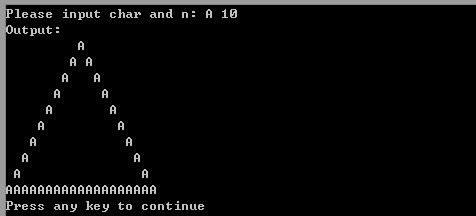
}

把一个n \* (2n-1) 的区域里除掏空成为一个镂空的三角形，减轻重量，但关键是为了追求另一种视觉效果。在设计的过程中，需要给出各种花纹的材料和大小尺寸的三角形样板，通过电脑临时做出来，以便看看效果。

### 输入格式

每行包含一个字符和一个整数n(3<=n<=35)，不同的字符表示不同的花纹，整数n表示等腰三角形的高。显然其底边长为2n-1。

输出如图：(在其它镂空的地方以空格代替)



#include<stdio.h>

int main()

{

int n,i,j;

char c;

printf("Please input char and n: ");

c=getchar();

scanf("%d",&n);

printf("Output:\n");;

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

{

for(j=0;j<2\*n-1;j++)

{

if(i+j==n-1)

{

printf("%c",c);

}

else if(j-i==n-1)

{

printf("%c",c);

}

else

{

printf(" ");

}

}

putchar(10);

}

for(j=0;j<2\*n-1;j++)

{

printf("%c",c);

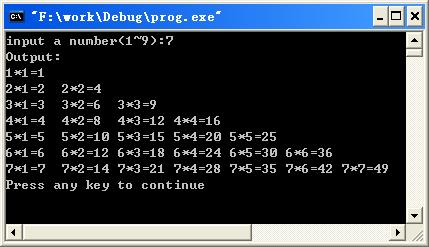
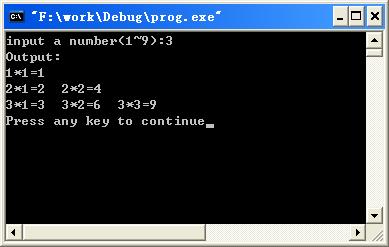
}

putchar(10);

return 0;

}

从键盘输入1～9之间的一个数，根据输入的数，打印下三角乘法口诀表。要求积的输出占3个宽度，且左对齐。编程可用素材：printf("input a number(1~9): ")...。  
　　程序的运行效果应类似地如图1和图2所示，图1中的3和图2中的7是从键盘输入的内容



#include <stdio.h>

int main(void){

int i, j, n;

printf("input a number(1~9):");

scanf("%d", &n);

printf("Output:\n");

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

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

{

printf("%d\*%d=%-3d", i, j, i \* j);

}

printf("\n");

}

return 0;

}

### 题目描述

给定两个正整数，计算这两个数的最小公倍数。

### 输入格式

输入测试数据，包括两个不大于1000的正整数。

### 输出

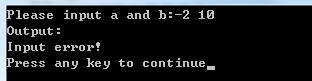
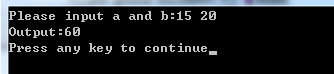
对于每个测试用例，给出这两个数的最小公倍数，每个实例输出一行。

### 样例输入

20 15

### 样例输出

60



#include<stdio.h>

int main()

{

int a,b,k,t,i;

printf("Please input a and b:");

scanf("%d %d",&a,&b);

if(a>1000||a<=0||b>1000||b<=0)

{

printf("Output:\nInput error!\n");

return 0;

}

else

{

if(a<b)

{

k=a;

a=b;

b=k;

}

t=a\*b;

while(b!=0)

{

i=a%b;

a=b;

b=i;

}

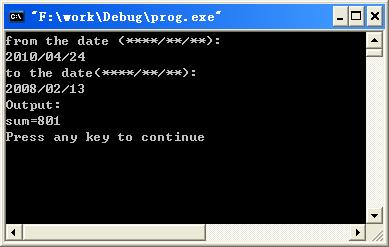
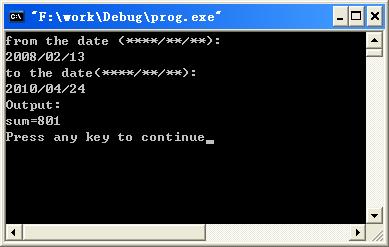
printf("Output:");

printf("%d\n",t/a);

return 0;}

}

计算两个日期之间的天数。注：判断年份是否为闰年的方法&mdash;&mdash;为400的倍数为闰年，如2000年；若非100的倍数，而是4的倍数，为闰年，如1996年。编程可用素材：printf("from the date (\*\*\*\*/\*\*/\*\*):\n")...、printf("to the date(\*\*\*\*/\*\*/\*\*):\n")、printf("Output:\nsum=%d\n"...。  
　　程序的运行效果应类似地如图1和图2所示，图1中的2008/02/13、2010/04/24和图2中的2010/04/24、2008/02/13是从键盘输入的内容。



#include <stdio.h>

int runyear(int year);

int days(int year, int month, int day);

int main(void)

{

int i, sum = 0;

int year2, month2, day2;

int year1, month1, day1;

int temp;

printf("from the date (\*\*\*\*/\*\*/\*\*):\n");

scanf("%d/%d/%d", &year2, &month2, &day2);

printf("to the date(\*\*\*\*/\*\*/\*\*):\n");

scanf("%d/%d/%d", &year1, &month1, &day1);

if (year1 > year2 || (year1 == year2 && month1 > month2)

|| (year1 == year2 && month1 == month2 && day1 > day2))

{

temp = year1;

year1 = year2;

year2 = temp;

temp = month1;

month1 = month2;

month2 = temp;

temp = day1;

day1 = day2;

day2 = temp;

}

for (i = year1 ; i < year2; i++)

{

if (i == year1)

{

if (month1 <=2 && runyear(i) == 1)

{

sum = sum + 366;

}

else

{

sum = sum + 365;

}

continue;

}

if (runyear(i) == 1)

{

sum = sum + 366;

continue;

}

sum = sum + 365;

}

sum = sum - days(year1, month1, day1) + days(year2, month2, day2);

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

return 0;

}

int runyear(int year)

{

if (year % 400 == 0 || (year % 100 != 0 && year % 4 == 0))

{

return (1);

}

else

{

return (0);

}

}

int days(int year, int month, int day)

{

int j, m = 0;

int array[13] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

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

{

m = m + array[j];

}

m = m + day;

if (runyear(year) == 1 && month > 2)

{

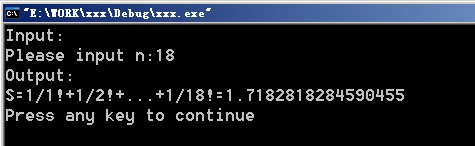
m++;

}

return (m);

}

求S=1/1！+1/2！+1/3！+…+1/N！并输出结果(显示时小数部分占16位，计算时要求从第1项开始往后累加)。N为任意自然数(只考虑int型)，从键盘读入。编程可用素材：printf("Input:\nPlease input n: ")...、printf("Output:\nS=1/1!+1/2!+...+1/...!=...。  
程序的运行效果应类似地如图1所示，图1中的18是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int n;

int i;

double result = 0.0, s = 1.0;

printf("Input:\nPlease input n:");

scanf("%d", &n);

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

{

s = s \* i;

result += 1 / s;

}

printf("Output:\nS=1/1!+1/2!+...+1/%d!=%.16f\n", n, result);

return 0;

}

根据输入的n在屏幕上显示对应的以#组成的菱形图案。编程可用素材：printf("Please input n: ")...。  
　　程序的运行效果应类似地如图1和图2所示，图1中的4和图2中的1是从键盘输入的内容。

图1：

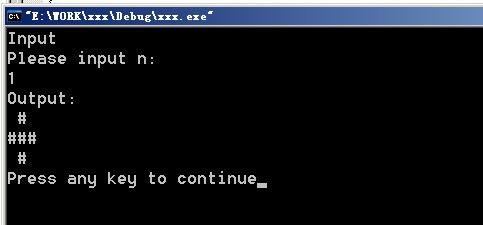
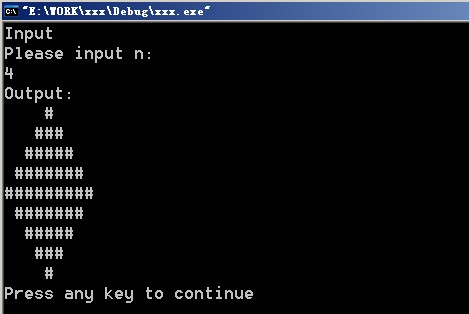


图2：

#include <stdio.h>

int main(void)

{

int i, j, n;

printf("Input\nPlease input n: \n");

scanf("%d", &n);

printf("Output:\n");

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

{

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

{

printf(" ");

}

for (j = 1; j <= 2 \* i - 1; j++)

{

printf("#");

}

printf("\n");

}

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

{

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

{

printf(" ");

}

for (j = 1; j <= 2 \* (n - i + 1) - 1; j++)

{

printf("#");

}

printf("\n");

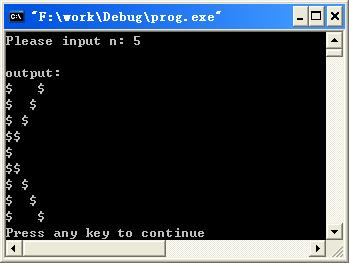
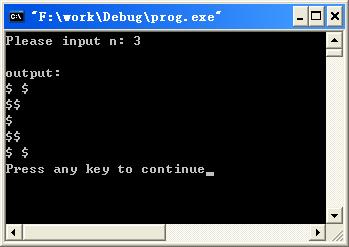
}

return 0;

}

编写一程序P214.C实现以下功能  
　　根据输入的n(约定n>0)在屏幕上显示对应的图案。

　　程序的运行效果应类似地如图1和图2所示，图1中的3和图2中的5是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int i, j, n;

printf("Please input n: ");

scanf("%d", &n);

printf("\noutput:\n");

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

{

if (i == n)

{

printf("$\n");

continue;

}

printf("$");

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

{

printf(" ");

}

printf("$");

printf("\n");

}

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

{

printf("$");

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

{

printf(" ");

}

printf("$");

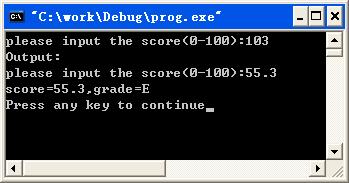
printf("\n");

}

return 0;

}

编写一程序实现以下功能  
　　从键盘输入一个一百分制成绩，如果不在0～100范围内，则要求重新输入数据，直到输入的数据在0～100范围内。将输入的数据转换成等级‘A’，‘B’，‘C’，‘D’，‘E’。90分以上为‘A’，80～89分为‘B’，70～79分为‘C’，60～69分为‘D’，60分以下为‘E’，要求使用switch、case、default语句，结果赋值给变量grade，并将变量grade的值输出到屏幕上，注意输出是Output:只显示一次。编程可用素材：printf("please input the score(0-100): ")...、printf("score=...,grade=...\n",...。  
　　程序的运行效果应类似地如图1所示，图1中的103和55.3是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

float score;

char grade;

printf("please input the score(0-100):");

scanf("%f", &score);

printf("Output:\n");

while (score < 0 || score > 100)

{

printf("please input the score(0-100):");

scanf("%f", &score);

}

switch ((int)score / 10)

{

case 10:

case 9:

grade = 'A';

break;

case 8:

grade = 'B';

break;

case 7:

grade = 'C';

break;

case 6:

grade = 'D';

break;

default:

grade = 'E';

break;

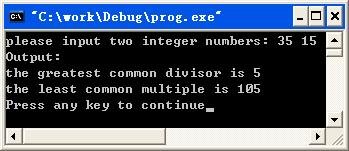
}

printf("score=%.1f,grade=%c\n", score, grade);

return 0;

}

　编写程序实现以下功能  
　　输入两个正整数m和n，求其最大公约数和最小公倍数。注：最大公约数也称最大公因子，指某几个整数共有因子中最大的一个；两个整数公有的倍数称为它们的公倍数，其中最小的一个正整数称为它们两个的最小公倍数。编程可用素材：printf("please input two integer numbers: ")...、printf("Output:\nthe greatest common divisor is ...\n"...、printf("the least common multiple is ...\n"...。  
　　程序的运行效果应类似地如图1所示，图1中的35 15是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int number1, number2, rem, i, j;

printf("please input two integer numbers: ");

scanf("%d%d", &number1, &number2);

i = number1;

j = number2;

rem = number1 % number2;

while (rem != 0)

{

i = number2;

j = rem;

rem = i % j;

}

printf("Output:\n");

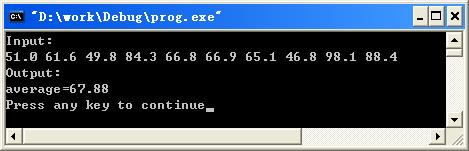
printf("the greatest common divisor is %d\n", j);

printf("the least common multiple is %d\n", number1 \* number2 / j);

return 0;

}

从键盘输入10个学生成绩（double），计算并输入这10个学生的平均成绩（保留小数点后2位）。



#include <stdio.h>

int main(void)

{

int i;

double arr[10];

double sum = 0.0, aver;

printf("Input:\n");

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

{

scanf("%lf", &arr[i]);

sum += arr[i];

}

aver = sum / 10;

printf("Output:\n");

printf("average=%.2f\n", aver);

return 0;

}

### 题目描述

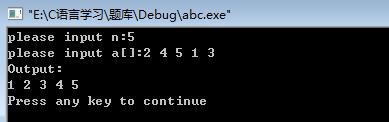
对输入的n个数进行排序并输出。

### 输入格式

输入的第一行包括一个整数n(1<=n<=100)。 接下来的一行包括n个整数。

### 输出

将排序后的n个整数输出，每个数后面都有一个空格。



#include<stdio.h>

int main()

{

int n,i,j,t,a[100];

printf("please input n:");

scanf("%d",&n);

printf("please input a[]:");

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

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

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

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

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

{

t=a[i];

a[i]=a[j];

a[j]=t;

}

printf("Output:\n");

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

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

printf("\n");

return 0;

}

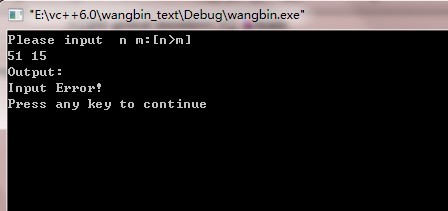
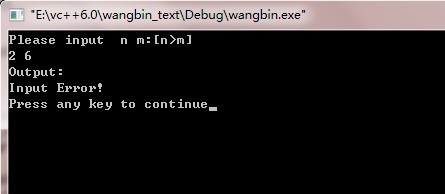
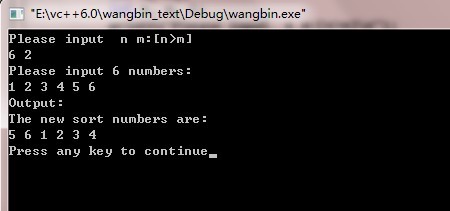
输入n个整数，使前面的n-m个数顺序向后移动m个位置，最后的m个数变成最前面的m个数。

### 输入格式

第一行包含两个用空格隔开的整数n和m，保证n>m且n不超过50，如果输入不符合以上要求，要有“Input Error！”提示。第二行包含n个用空格隔开的整数。

### 输出

输出进行题目描述操作之后的n个整数，每个整数之后输出一个空格。  
请注意行尾输出换行。



#include <stdio.h>

int main() {

void update(int x[], int n, int m);

int a[50];

int n, m, i;

printf("Please input n m:[n>m]\n");

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

if((n<=m)||(n>50))

{

printf("Output:\nInput Error!\n");

}

else

{

printf("Please input %d numbers:\n",n);

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

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

update(a, n, m);

printf("Output:\nThe new sort numbers are:\n");

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

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

printf("\n");

}

return 0;

}

void update(int x[], int n, int m) {

int tmp[50];

int i;

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

tmp[i] = x[i];

for (i = n - m - 1;i >= 0;i--)

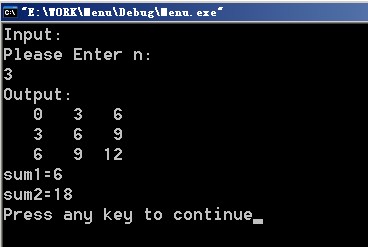
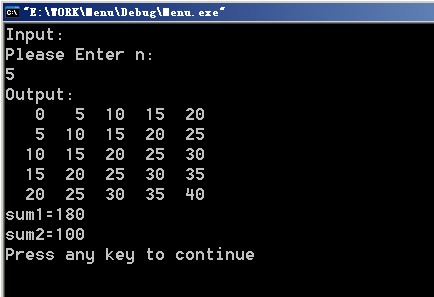
x[i + m] = x[i];

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

x[i] = tmp[i + (n - m)];

}

从键盘上输入矩阵的阶数n(n<=14), 矩阵中元素的值等于其位置的行数和列数之和的n倍（行列的值从0开始计数），如n=3时，矩阵为  
        0   3   6  
        3   6   9  
        6   9   12  
　　先输出该矩阵(显示时每个数宽度为4、右对齐)，然后计算输出sum1和sum2的值：sum1为矩阵中所有不靠边元素之和、sum2为矩阵的一条对角线元素之和。编程可用素材：printf("Enter n: ")...、printf("sum1=...\nsum2=...\n"....



#include <stdio.h>

int main(void)

{

int i, j, n, sum1 = 0, sum2 = 0;

int array[15][15];

printf("Input:\nPlease Enter n:\n");

scanf("%d", &n);

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

{

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

{

array[i][j] = (i + j) \* n;

}

}

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

{

for (j = 1; j < n - 1; j++)

{

sum1 += array[i][j];

}

}

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

{

sum2 += array[i][i];

}

printf("Output:\n");

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

{

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

{

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

}

printf("\n");

}

printf("sum1=%d\nsum2=%d\n", sum1, sum2);

return 0;

}

### 题目描述

对一个字符串中的所有**单词**，如果单词的首字母不是大写字母，则把单词的首字母变成大写字母。  
在字符串中，单词之间通过空白符分隔，空白符包括：空格(' ')、制表符('\t')。

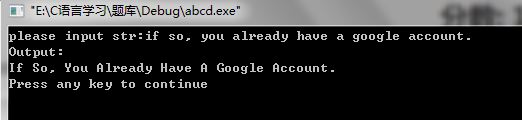
(注：程序中的字符串输入输出用gets和puts）。

### 输入格式

输入一行：待处理的字符串（长度小于100，逗号后要先输入一个空格再输入单词）。

### 输出

转换后的字符串。



#include <stdio.h>

int main()

{

int i=1;

char str[110] = " "; // 注意这里先将字符串初始化为一个空格，而非空串

printf("please input str:");

gets(str+1);

// 注意这里读入字符时，是从第一个字符开始，而不是字符串的开头，及第一个字符存放在str[1]里面

// 这样是为了方便判断，因为程序判断这个字母是否该大写是根据该字母前时候有空格来的。

while(str[i])// 注意这里下标也从 1 开始

{

if((str[i-1]==' ' || str[i-1]=='\t') && 'a'<=str[i] && str[i]<='z')

{

str[i] -= 32;

}

i++;

}

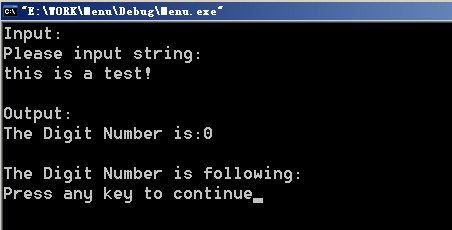
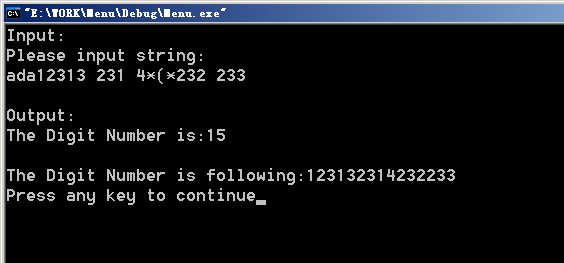
printf("Output:\n");

puts(str+1); // 注意这里输出也是从第一个位置开始，而非是字符串的起始位置

return 0;

}

从键盘读入一行字符(约定：字符数≤127字节)，将其中的数字字符以及这些数字字符的数量在屏幕上显示，注意：要求先显示这些数字字符的数量。编程可用素材：printf("Input:\nPlease input string:\n")...、printf("\nOutput:\nThe Digit Number is:...、printf("\nThe Digit Number is following:...。  
　　程序的运行效果应类似地如图1所示，图1中的this is a test!,ada12313 231 4\*(\*232 233是从键盘输入的内容。



#include <stdio.h>

void main()

{

int i, j;

char string[128];

printf("Input:\nPlease input string:\n");

gets(string);

for (i = 0, j = 0; string[i] != '\0'; i++)

{

if (string[i] >= '0' && string[i] <= '9')

{

string[j] = string[i];

j++;

}

}

string[j] = '\0';

printf("\nOutput:\nThe Digit Number is:%d\n", j);

printf("\nThe Digit Number is following:%s\n", string);

}

### 题目描述

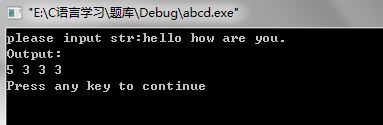
编一个程序，读入用户输入的，以“.”结尾的一行文字，统计一共有多少个单词，并分别输出每个单词含有多少个字符。（凡是以一个或多个空格隔开的部分就为一个单词，不判断输入是否以"."结尾，单词中包含的数字也要统计）

### 输入格式

输入包括1行字符串，以“.”结束，字符串中包含多个单词，单词之间以一个或多个空格隔开。

### 输出

输出字符串中每个单词包含的字母的个数。



#include <stdio.h>

#include <string.h>

int main()

{

char str[1000]; // 用来读取整个字符串

char \*pStr; // 一个字符指针，用来存储以空格和点号分割的字符串

int iCount = 0; // 记录已经有多少数字输出了，由于每个数字间用一个空格分开，因而需要记录数字的输出数目

printf("please input str:");

gets(str);

pStr = strtok(str, " ."); // 先用空格和点号作一次切割

printf("Output:\n");

while(pStr)

{

if(iCount++)

{

putchar(' '); // 数字间使用一个空格分割

}

printf("%d", strlen(pStr)); // 输出单词的长度

pStr = strtok(NULL, " ."); // 做下一次分割

}

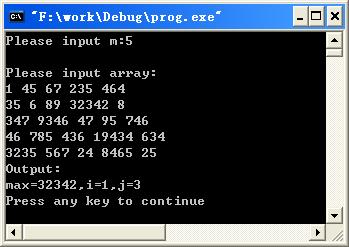
putchar('\n'); // 别忘了换行

return 0;

}

求任意的一个m×m矩阵的最大数及其所在的行列数，m(2<=m<=20)及矩阵元素从键盘输入(只考虑int型)。

程序的运行效果应类似地如图所示，图中的5和  
1 45 67 235 464  
35 6 89 32342 8   
347 9346 47 95 746  
46 785 436 19434 634  
3235 567 24 8465 25  
是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int i, j, m, max, maxi, maxj;

int array[20][20];

printf("Please input m:");

scanf("%d", &m);

printf("\nPlease input array:\n");

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

{

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

{

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

}

}

max = array[0][0];

maxi=0;maxj=0;

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

{

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

{

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

{

max = array[i][j];

maxi = i;

maxj = j;

}

}

}

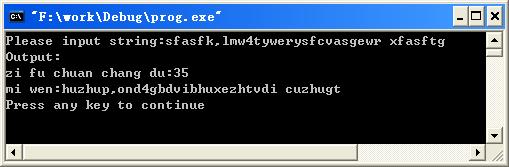
printf("Output:");

printf("\nmax=%d,i=%d,j=%d\n", max, maxi, maxj);

return 0;

}

从键盘上读入一行字符(约定：字符数≤127字节)，按以下方法将其加密变换：  
　　　　A->Z   　a->z  
　　　　B->Y   　b->y  
　　　　C->X   　c->x  
　　　　……   　……  
　　　　Z->A   　z->a  
　　即字母A变成Z、字母B变成Y...，非字母字符不变。最后在屏幕上先显示这一行字符的长度，再显示生成的密文。  
　　编程可用素材：printf("Please input string:")...、printf("Output:\nzi fu chuan chang du:...、printf("\nmi wen:...。  
　　程序的运行效果应类似地如图1所示，图1中的sfasfk,lmw4tywerysfcvasgewr xfasftg是从键盘输入的内容



#include <stdio.h>

int main(void)

{

int i;

char string[128];

printf("Please input string:");

gets(string);

for (i = 0; string[i] != '\0'; i++)

{

if (string[i] >= 'a' && string[i] <= 'z')

{

string[i] = 'a' + 25 - (string[i] - 'a');

}

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

{

string[i] = 'A' + 25 - (string[i] - 'A');

}

}

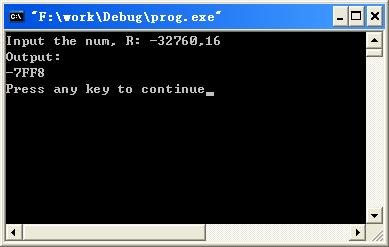
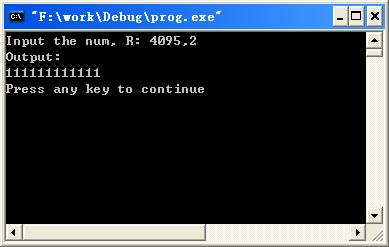
printf("Output:\nzi fu chuan chang du:%d", i);

printf("\nmi wen:%s\n", string);

return 0;

}

实现十进制转num换成R进制数，num和R从键盘读入(假定num为int且-32767<=num<=32767，R为int且2<=R<=16)。编程可用素材：printf("Input the num, R: ")...、printf("Output:\n")...。  
　　程序的运行效果应类似地如图1和图2所示，图1中的4095,2和图2中的-32760,16是从键盘输入的内容。



#include <stdio.h>

#include <string.h>

void fun(char str[40], int num, int rad);

int main(void)

{

int num, rad;

char str[40];

printf("Input the num, R: ");

scanf("%d,%d", &num, &rad);

fun(str, num, rad);

printf("Output:\n");

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

return 0;

}

void fun(char str[40], int num, int rad)

{

int i = 0, j;

int flag = 0;

char str2[40];

char number[17] = "0123456789ABCDEF";

if (num == 0)

{

strcpy(str, "0");

return ;

}

if (num < 0)

{

flag = 1;

num = -num;

}

while (num > 0)

{

str2[i] = number[num % rad];

i++;

num = num / rad;

}

str2[i] = '\0';

j = strlen(str2) - 1;

if (flag == 0)

{

i = 0;

}

else

{

str[0] = '-';

i = 1;

}

for (; j >= 0; i++, j--)

{

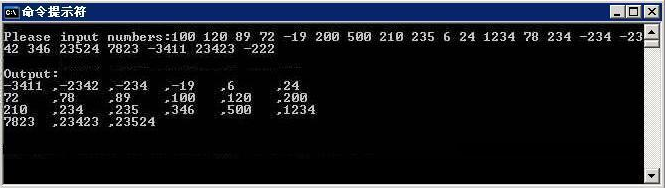
str[i] = str2[j];

}

str[i] = '\0';

}

①程序运行时先显示Please input numbers:，再从键盘上读入一组整数(只考虑int型)，数与数之间只使用空格或回车作分隔。数可正可负，最多10000个，但若读入的数为-222时，则表示输入结束且-222不算在该组数内。  
　　②对这一组数按从小到大的顺序进行排序。  
　　③将排序后的这一组数输出到屏幕上，输出格式为每行6个数，数与数之间使用逗号(,)分隔，两个逗号之间的宽度(不算逗号)为6且使用左对齐格式。注意，行尾没有逗号。



#include <stdio.h>

int main(void)

{

int i, j, cnt, temp;

int array[10000];

printf("Please input numbers:");

for (i = 0; i < 10000; i++)

{

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

if (array[i] == -222)

{

break;

}

}

cnt = i;

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

{

for (j = 0; j < cnt - i - 1; j++)

{

if (array[j] > array[j + 1])

{

temp = array[j];

array[j] = array[j + 1];

array[j + 1] = temp;

}

}

}

printf("\nOutput:\n");

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

{

if (i == cnt - 1)

{

printf("%-6d\n", array[i]);

break;

}

if ((i + 1) % 6 != 0)

{

printf("%-6d,", array[i]);

}

else if ((i + 1) % 6 == 0)

{

printf("%-6d", array[i]);

printf("\n");

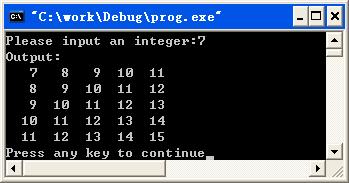
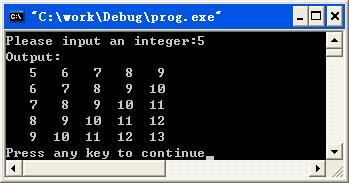
}

}

return 0;

}

编写程序实现以下功能  
　　用键盘输入的整数产生5\*5矩阵N，并按行输出该矩阵，每个元素占4个数位、右对齐。  
　　程序的运行效果应类似地如图1和图2所示，图1中的5和图2中的7是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int n, i, j;

int array[5][5];

printf("Please input an integer:");

scanf("%d", &n);

printf("Output:\n");

for (i = 0; i < 5; i++)

{

for (j = 0; j < 5; j++)

{

array[i][j] = n + i + j;

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

}

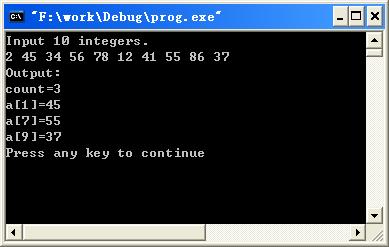
printf("\n");

}

return 0;

}

输入10个整型数存入一维数组，输出值和下标都为奇数(数组第1个元素的下标为0)的元素及其个数，要求先输出个数。编程可用素材：printf("Input 10 integers.\n")...、printf("Output:\ncount=...\n"...、printf("a[...]=...\n"...。  
　　程序的运行效果应类似地如图1所示，图1中的2 45 34 56 78 12 41 55 86 37是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int i, count = 0;

int array[10], barray[10];

printf("Input 10 integers.\n");

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

{

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

}

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

{

if (array[i] % 2 == 1 && i % 2 == 1)

{

barray[count] = i;

count++;

}

}

printf("Output:\ncount=%d\n", count);

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

{

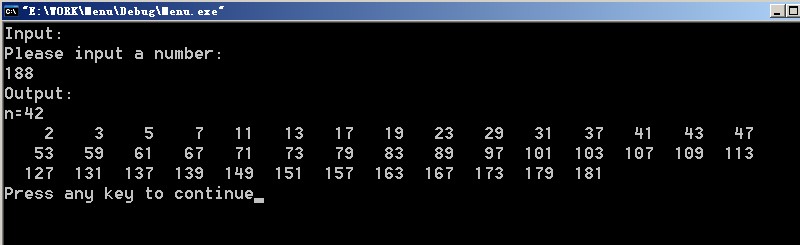
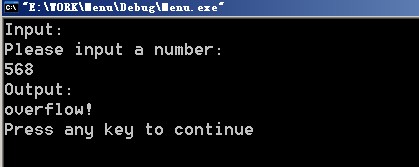
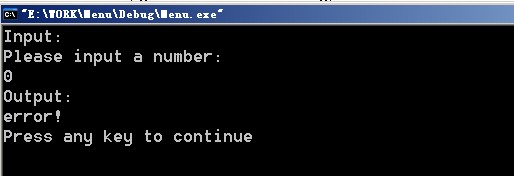
printf("a[%d]=%d\n", barray[i], array[barray[i] ]);

}

return 0;

}

输入整数m，将所有大于1小于整数m的素数存入所指定的数组中(数组最多只存放100个素数，超过则提示“overflow”)，输出素数的个数n及各素数——素数的输出格式为每个素数5列宽、右对齐、每行显示15个。若输入的m≤0，则提示“error”，程序终止。注：素数(Prime Number)，亦称质数，指在一个大于1的自然数中，除了1和此整数自身外，没法被其他自然数整除的数。  
　　编程可用素材：printf("input a number: ")...、printf("error")...、printf("overflow")...、printf("n=...\n"...。  
　　程序的运行效果应类似地如图1、图2和图3所示，图1中的188、图2中的0和图3中的568是从键盘输入的内容



#include <stdio.h>

int main(void)

{

int m, cnt = 0, i, j;

int array[101];

printf("Input:\nPlease input a number: \n");

scanf("%d", &m);

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

{

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

{

if (i % j == 0)

{

break;

}

}

if (j == i)

{

array[cnt++] = i;

}

if (cnt > 100)

{

break;

}

}

printf("Output:\n");

if (m <= 0)

{

printf("error!\n");

}

else if (cnt > 100)

{

printf("overflow!\n");

}

else

{

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

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

{

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

if ((i + 1) % 15 == 0)

{

printf("\n");

}

}

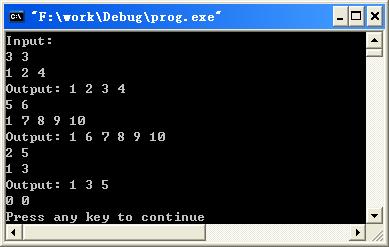
printf("\n");

}

return 0;

}

保持数列有序：有n(约定n<=100)个整数，已经按照从小到大顺序排列好，现在另外给一个整数x，请将该数插入到序列中，并使新的序列仍然有序。  
　　程序输入：输入数据包含多组测试实例，每组数据由两行组成，第一行是n和x，第二行是已经有序的n个数的数列。n和x同时为0表示输入数据的结束，本行不做处理。  
　　程序输出：对于每个测试实例，输出插入新的元素后的数列。  
　　程序的运行效果应类似地如图1所示，图1中的3 3、1 2 4和0 0是从键盘输入内容。



#include <stdio.h>

int main(void){

int i, j, n, x;

int arr[101];

printf("Input:\n");

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

while (n != 0 || x != 0){

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

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

}

for (j = n - 1; j >= 0 && x < arr[j]; j--){

arr[j + 1] = arr[j];

}

arr[j + 1] = x;

printf("Output: ");

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

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

}

printf("\n");

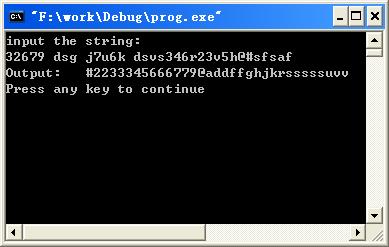
scanf("%d%d", &n, &x);

}

return 0;

}

对从键盘输入的一行字符(约定：字符数≤127字节)进行排序(按每个字符的ASCII码由小到大)并输出。编程可用素材：printf("input the string: \n")...、printf("Output:")...。  
程序的运行效果应类似地如图1所示，图1中的32679 dsg j7u6k dsvs346r23v5h@#sfsaf是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int i = 0, j, ch, len;

char te;

char str[128];

printf("input the string: \n");

ch = getchar();

while (ch != '\n' && ch != EOF)

{

str[i] = (char)ch;

i++;

ch = getchar();

}

str[i] = '\0';

len = i;

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

{

for (j = 0; j < len - i - 1; j++)

{

if (str[j] > str[j + 1])

{

te = str[j];

str[j] = str[j + 1];

str[j + 1] = te;

}

}

}

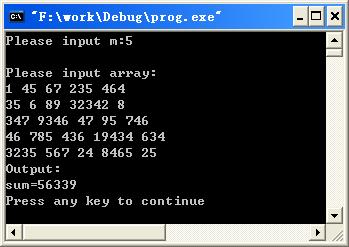
printf("Output:%s\n", str);

return 0;

}

求任意的一个m×m矩阵的主副对角线上元素之和（注：每个元素只计一次），m(2<=m<=20)及矩阵元素从键盘输入(只考虑int型，且不需考虑求和的结果可能超出int型能表示的范围)。

程序的运行效果应类似地如图所示，图中的5和  
1 45 67 235 464  
35 6 89 32342 8   
347 9346 47 95 746  
46 785 436 19434 634  
3235 567 24 8465 25  
是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int i, j, m, sum = 0;

int array[20][20];

printf("Please input m:");

scanf("%d", &m);

printf("\nPlease input array:\n");

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

{

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

{

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

}

}

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

{

sum = sum + array[i][i] + array[i][m - i - 1];

}

if (m % 2 == 1)

{

sum = sum - array[m / 2][m /2];

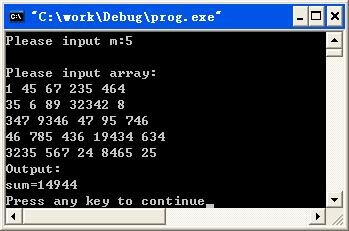
}

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

return 0;

}

编写一程序实现以下功能  
　　求任意的一个m×m矩阵的周边元素之和，m(2<=m<=20)及矩阵元素从键盘输入(只考虑int型，且不需考虑求和的结果可能超出int型能表示的范围)。编程可用素材：printf("\nPlease input m: ")...、printf("\nPlease input array:\n")...、printf("Output:\nsum=...\n"...。  
　　程序的运行效果应类似地如图1所示，图1中的5和  
1 45 67 235 464  
35 6 89 32342 8   
347 9346 47 95 746  
46 785 436 19434 634  
3235 567 24 8465 25  
　　是从键盘输入的内容。



/\*

5

1 45 67 235 464

35 6 89 32342 8

347 9346 47 95 746

46 785 436 19434 634

3235 567 24 8465 25

\*/

#include <stdio.h>

int main(void)

{

int i, j, m, sum = 0;

int array[20][20];

printf("Please input m:");

scanf("%d", &m);

printf("\nPlease input array:\n");

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

{

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

{

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

}

}

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

{

if (i == 0 || i == m - 1)

{

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

{

sum += array[i][j];

}

continue;

}

sum = sum + array[i][0] + array[i][m - 1];

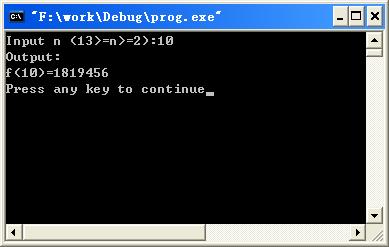
}

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

return 0;

}

有一递推数列，满足f(0)=0,f(1)=1,f(2)=2,  f(n+1)=2f(n)+f(n-1)f(n-2) (n>=2), 编写程序求f(n)的值（n由键盘输入，13>=n>=2）。编程可用素材：printf("Input n (13>=n>=2): ")...、printf("Output:\nf(...)=...\n"...。  
　　程序的运行效果应类似地如图1所示，图1中的10是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int i, n;

double array[21];

printf("Input n (13>=n>=2):");

scanf("%d", &n);

array[0] = 0;

array[1] = 1;

array[2] = 2;

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

{

array[i] = 2 \* array[i - 1] + array[i - 2] \* array[i - 3];

}

printf("Output:\nf(%d)=%.0f\n", n, array[n]);

return 0;

}

求斐波拉契数列的前n（n的范围为：3～20）项，并输出到屏幕上（数和数之间用字符'\t'隔开）。斐波拉契公式：f(1)=1,f(2)=1,f(n)=f(n-1)+f(n-2) (n≥3)。编程可用素材：printf("input a data(3--20):")...printf("Output:\n")...。  
　　程序的运行效果应类似地如图1所示，图1中的11是从键盘输入的内容。



#include <stdio.h>

int main(void)

{

int i, n;

int array[21] = {1, 1};

printf("input a data(3--20):");

scanf("%d", &n);

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

{

array[i] = array[i - 1] + array[i - 2];

}

printf("Output:\n");

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

{

printf("%d\t", array[i]);

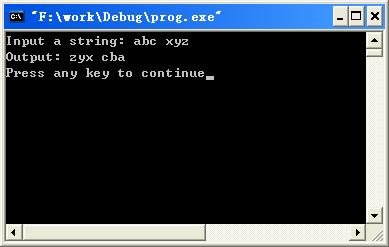
}

printf("%d\n", array[i]);

return 0;

}

编写程序实现以下功能  
　　从键盘输入一行可带空格的字符串(约定：字符数≤127字节)，按逆序输出该字符串。编程可用素材：printf("Input a string: ")...、printf("Output: ")...。  
　　程序的运行效果应类似地如图1所示，图1中的abc xyz是从键盘输入的内容。



#include <stdio.h>

#include <string.h>

int main(void)

{

int i;

char string[128];

printf("Input a string: ");

gets(string);

printf("Output: ");

for (i = strlen(string) - 1; i >= 0; i--)

{

printf("%c", string[i]);

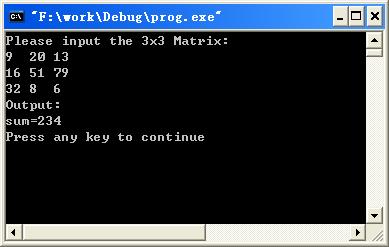
}

printf("\n");

return 0;

}

输入3行3列的矩阵，输出所有元素的累加和（整型）。编程可用素材：printf("Please input the 3x3 Matrix:\n"...、printf("Output:\nsum=...\n"...。  
　　程序的运行效果应类似地如图1所示，图1中的  
        9  20 13  
        16 51 79  
        32 8  6  
　　是从键盘输入的内容。



#include <stdio.h>

int main(void){

int i, j, sum = 0;

int array[3][3];

printf("Please input the 3x3 Matrix:\n");

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

{

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

{

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

sum += array[i][j];

}

}

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

return 0;

}

函数

程序运行时，先从键盘输入二个不相同的整数值，依次调用GetMax,GetMin,Getaverage函数获取二个数中的最大值，最小最，平均值，并保存到Result数组中。然后输出。注意：只允许在/\*\*\*\*\*\*start\*\*\*\*\*\*/和/\*\*\*\*\*\*end\*\*\*\*\*\*/之间添加代码。



#include <stdio.h>

#include <stdlib.h>

/\*\*\*\*\*\*start\*\*\*\*\*\*/

int GetMax(int x, int y);

int GetMin(int x, int y);

int GetAverage(int x, int y);

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

int main()

{

int Number1 = 0,Number2 = 0;

int Result[3], i = 0; //Result数组用来存存储的最大值，最小值和平均值

printf("output:\ninput two numbers:\n");

scanf("%d%d", &Number1, &Number2);

if(Number1 == Number2)

{

printf("输入二个不同的数字：\n");

return 0;

}

/\*在start和end之间完成函数的调用和参数的传递以及返回值的存储\*/

/\*\*\*\*\*\*start\*\*\*\*\*\*/

Result[0] = GetMax(Number1,Number2);

Result[1] = GetMin(Number1,Number2);

Result[2] = GetAverage(Number1,Number2);

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

printf("\n最大数 最小数 平均值\n");

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

{

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

}

printf("\n");

return 0;

}

//取最大值函数

int GetMax(int x, int y)

{

return (x > y) ? x : y;

}

//取最小值函数

int GetMin(int x, int y)

{

return (x > y) ? y : x;

}

//取平均值函数

int GetAverage(int x, int y)

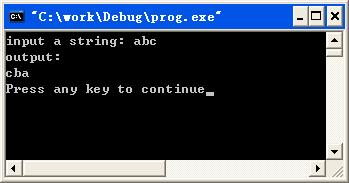
{

return (x + y) / 2;

}

完成以下代码中的函数声明和定义。该函数用于将字符串反序，如"abc"反序为"cba"。

注意：只允许在/\*\*\*\*\*\*start\*\*\*\*\*\*/和/\*\*\*\*\*\*end\*\*\*\*\*\*/之间添加代码。



#include <stdio.h>

#include <string.h>

/\*\*\*\*\*\*start\*\*\*\*\*\*/

void ReverseSTring(char str[]);

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

int main()

{

char str[200]={0};

printf("input a string: ");

gets(str);

ReverseString(str);

puts("output:");

puts(str);

return 0;

}

/\*\*\*\*\*\*start\*\*\*\*\*\*/

void ReverseSTring(char str[])

{

char ch;

int i, j;

j = strlen(str) - 1;

for (i = 0; i < j; i++, j--)

{

ch = str[i];

str[i] = str[j];

str[j] = ch;

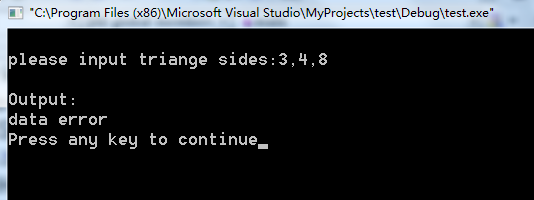
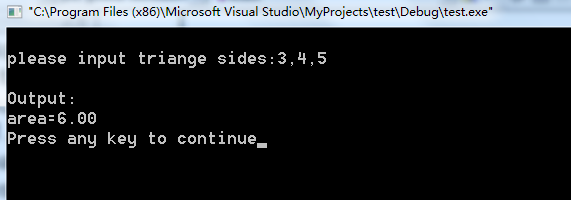
}

}

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

根据要求编写程序D5201的指定部分：

　　程序D5201.C已编写部分代码，请根据程序中的要求完善程序——注意，只能在/\*\*\*\*\*\*start\*\*\*\*\*\*/和/\*\*\*\*\*\*end\*\*\*\*\*\*/之间添加代码，不能对程序中已有部分作任何修改或重新编写一个程序，否则作0分处理。  
　　程序的功能是：输入三角形的三边长a、b、c(约定为整数)，求三角形面积area，并输出。如果输入的三边构不成三角形，应给出“data error”的信息提示。注：根据“海伦－秦九韶”公式，area＝√p(p-a)(p-b)(p-c)，其中p＝(a+b+c)/2。  
　　程序的运行效果应类似地如图1和图2所示，图1中的3,4,5和图2中的3,4,8是从键盘输入的内容。



#include <math.h>

#include <stdio.h>

/\* 本部分代码功能建议：函数原型声明 \*/

/\*\*\*\*\*\*start\*\*\*\*\*\*/

double fun(int a, int b, int c);

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

int main(void)

{

int aBian, bBian, cBian;

double mJi;

printf("\nplease input triange sides:");

scanf("%d,%d,%d", &aBian, &bBian, &cBian);

if (aBian<0 || bBian<0 || cBian<0

|| (aBian+bBian <= cBian) || (aBian+cBian <= bBian) || (bBian+cBian <= aBian))

{

printf("\nOutput:\ndata error\n");

}

else

{

mJi = fun(aBian, bBian, cBian);

printf("\nOutput:\narea=%.2f\n", mJi);

}

return 0;

}

/\*\*\*\*\*\*start\*\*\*\*\*\*/

double fun(int a, int b, int c)

{

double p;

p = (a+b+c)/2.0;

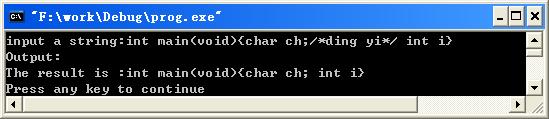
return sqrt(p\*(p-a)\*(p-b)\*(p-c));

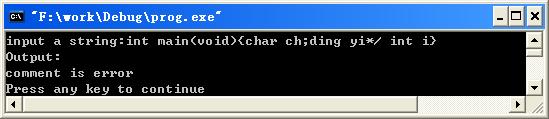
}

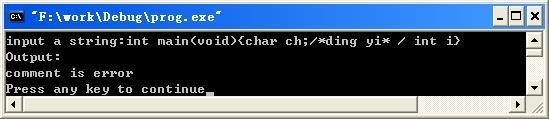
/\*\*\*\*\*\*end\*\*\*\*\*\*/

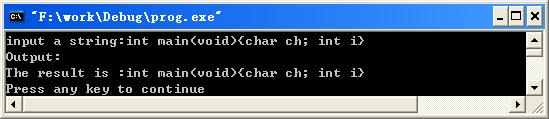
从键盘上读入一行字符(约定：字符数≤127字节)，判断其中的注释是否合法，不合法则报错，合法时则删除注释后再输出。合法注释是指“/\*”标记注释开始、“\*/”标记注释结束，通常表现为/\* ……\*/。  
　　注意事项：  
(1)只考虑行内最多只包含一个注释的情况。  
(2)不合法的注释情况有很多种，例如 ……\*/缺注释开始标记、/  \* ……\*/缺注释开始标记、/\* ……缺注释结束标记、/\* ……\*  /缺结束标记。  
(3)编程可用素材：printf("input a string:")...、printf("Output:\nThe result is :")...、printf("Output:\ncomment is error\n")...。  
　　程序的运行效果应类似地如图1、图2、图3和图4所示，图1中的int main(void){char ch;/\*ding yi\*/ int i}、图2中的int main(void){char ch;ding yi\*/ int i}、图3中的int main(void){char ch;/\*ding yi\* / int i}和图4中的int main(void){char ch; int i}是从键盘输入的内容。

程序运行效果示例：









#include <stdio.h>

#include <string.h>

int findfirst(char str[128]);

int findlast(char str[128]);

int main(void)

{

int first, last;

char string[128];

char result[128];

printf("input a string:");

gets(string);

first = findfirst(string);

last = findlast(string);

strcpy(result, string);

if (first >= 0 && last >= 0)

{

strcpy(result + first, string + last);

}

if (first >= 0 && last >= 0 && last >= first + 4 || first == -1 && last == -1)

{

printf("Output:\nThe result is :");

puts(result);

}

else

{

printf("Output:\ncomment is error\n");

}

return 0;

}

int findfirst(char str[128])

{

int i;

for (i = 0; str[i] != '\0'; i++)

{

if (str[i] == '/' && str[i + 1] == '\*')

{

return i;

}

}

return (-1);

}

int findlast(char str[128])

{

int i;

for (i = 0; str[i] != '\0'; i++)

{

if (str[i] == '\*' && str[i + 1] == '/')

{

return i + 2;

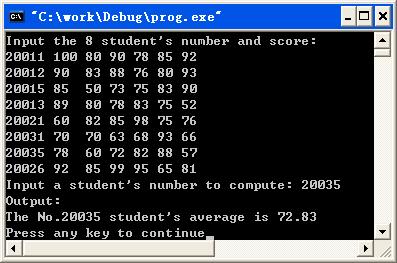
}

}

return (-1);

}

根据要求编写程序的指定部分：  
　　程序已编写部分代码，请根据程序中的要求完善程序——注意，只能在**/\*\*\*\*\*\*start\*\*\*\*\*\*/和/\*\*\*\*\*\*end\*\*\*\*\*\*/之间填加代码，不能对程序中已有其它部分作任何修改或重新编写一个程序，否则作0分处理**。  
　　程序的功能是：先从键盘读入的8个学生的6门课成绩存放在二维数组student中(每一行存储一个学生的数据，第0列为学号，第1～6列依次为6门课的成绩)，再从键盘读入一个学号，在student查找该学生是否存在，若存在，则输出其平均成绩，若不存在，则显示“not Find!”。  
　　程序的运行效果应类似地如图1所示，图1中的：  
20011 100 80 90 78 85 92  
20012 90  83 88 76 80 93  
20015 85  50 73 75 83 90  
20013 89  80 78 83 75 52  
20021 60  82 85 98 75 76  
20031 70  70 63 68 93 66  
20035 78  60 72 82 88 57  
20026 92  85 99 95 65 81  
　　和20035是从键盘输入的内容。



#include<stdio.h>

#define NOT\_FIND -1

#define TOTAL\_STU 8

/\* 函数aver的功能为：求学号为stuNo的学生的6门课课程之平均成绩，并通过函数值返回

若未找到学生stuNo，则返回NOT\_FIND \*/

float aver(int pStu[TOTAL\_STU][7], int stuNo);

int main(void)

{

int student[TOTAL\_STU][7]; /\* the first column save student's number \*/

float averScore;

int i, j, stuNumber;

printf("Input the %d student's number and score: \n", TOTAL\_STU);

/\* 本部分代码功能建议：从键盘读入的8个学生的6门课成绩存放在二维数组student中 \*/

/\*\*\*\*\*\*start\*\*\*\*\*\*/

for (i = 0; i < TOTAL\_STU; i++)

{

for (j = 0; j < 7; j++)

{

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

}

}

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

printf("Input a student's number to compute: ");

scanf("%d", &stuNumber);

averScore = aver(student, stuNumber);

printf("Output:");

if (NOT\_FIND == averScore)

{

printf("\nstudent of No.%d not Find!\n", stuNumber);

}

else

{

printf("\nThe No.%d student's average is %.2f\n", stuNumber, averScore);

}

return 0;

}

/\*\*\*\*\*\*start\*\*\*\*\*\*/

float aver(int pStu[TOTAL\_STU][7], int stuNo)

{

int i, j;

double sum = 0;

for (i = 0; i < TOTAL\_STU; i++)

{

if (pStu[i][0] == stuNo)

{

for (j = 1; j < 7; j++)

{

sum += pStu[i][j];

}

return (float)(sum/6);

}

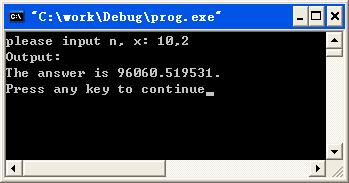
}

return NOT\_FIND;

}

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

　程序的功能是：从键盘输入一个整数n(n≥0)和x，计算对应的n阶勒让德多项式Pn(x)的值，并按示例格式输出相应信息。n阶勒让德多项式Pn(x)的定义如下：  
　　　　 　　　┌　1                                  (n=0)  
　　　　Pn(x)=    ├　x                                  (n=1)  
　　　　 　　　 └((2n-1)xPn-1(x) - (n-1)Pn-2(x)) / n      (n>1)  
　　程序的运行效果应类似地如图1所示，图1中的：10, 2是从键盘输入的内容。



#include <stdio.h>

/\* 本部分代码功能建议：调用相应的函数计算S \*/

/\*\*\*\*\*\*start\*\*\*\*\*\*/

double func(int x, int n);

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

/\* 考生添加代码结束 \*/

int main(void)

{

double Pnx;

int n, x;

printf("please input n, x: ");

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

/\* 本部分代码功能建议：调用相应的函数计算S \*/

/\*\*\*\*\*\*start\*\*\*\*\*\*/

Pnx = func(x, n);

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

/\* 考生添加代码结束 \*/

printf("Output:\nThe answer is %.6f.\n", Pnx);

return 0;

}

/\*\*\*\*\*\*start\*\*\*\*\*\*/

double func(int x, int n)

{

if (n == 0)

{

return 1;

}

else if (n == 1)

{

return x;

}

else

{

return ((2\*n-1)\*x\*func(x,n-1)-(n-1)\*func(x,n-2))/n;

}

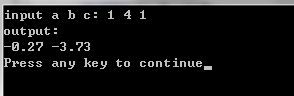
}

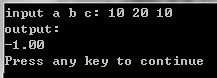
/\*\*\*\*\*\*end\*\*\*\*\*\*/

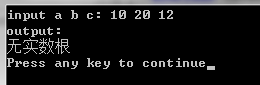
写一个求二次方程ax2+bx+c根的函数。在主函数中首先由键盘输入a、b、c，然后调用求根的函数resolve()求方程的根并输出。需注意b2-4ac大于、等于或小于0的情况。

程序代码中只允许在/\*\*\*\*\*start\*\*\*\*\*/和/\*\*\*\*\*end\*\*\*\*\*/之间填写代码，其余部分代码不能修改。输入输出使用：printf("input a b c: "), printf("output:\n"),printf("%.2f %.2f\n", ...)，等。

程序运行的结果如下图：







#include <stdio.h>

#include <math.h>

/\*\*\*\*\*\*start\*\*\*\*\*\*/

void resolve(double a,double b,double c);

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

int main()

{

double a, b, c=0;

printf("input a b c: ");

scanf("%lf%lf%lf", &a, &b, &c);

resolve(a, b, c);

return 0;

}

/\*\*\*\*\*\*start\*\*\*\*\*\*/

void resolve(double a,double b,double c)

{

double d;

d=pow(b,2)-4\*a\*c;

printf("output:\n");

if(a==0)

{

if(b==0)

printf("无实数根\n");

else

printf("%.2f\n",c/b);

}

else if (d<0)

printf("无实数根\n");

else if (d==0)

printf("%.2f\n",-b/(2\*a));

else

printf("%.2f %.2f\n",(sqrt(d)-b)/(2\*a),-(b+sqrt(d))/(2\*a));

}

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

请在以下代码中完成判断素数函数的代码。素数是只能被1或自身整除的正整数。

可用输入输出：puts("是素数"), puts("不是素数")。

注意：只允许在/\*\*\*\*\*\*start\*\*\*\*\*\*/和/\*\*\*\*\*\*end\*\*\*\*\*\*/之间添加代码。



#include <stdio.h>

/\*\*\*\*\*\*start\*\*\*\*\*\*/

int IsShuShu(int x);

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

int main()

{

int x;

printf("input a integer: ");

scanf("%d", &x);

printf("output:\n");

if(IsShuShu(x))

puts("是素数");

else

puts("不是素数");

}

/\*\*\*\*\*\*start\*\*\*\*\*\*/

int IsShuShu(int x)

{

int i;

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

{

if (x % i == 0)

{

return 0;

}

}

return 1;

}

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

从键盘读入一行字符(约定：字符数≤127字节，其中的空格不固定、有多有少)和加密间隔(假定只输入正整数)，将加密后的文字输出。具体加密方法示例为：对于输入“1 2 3 4 5 6 7   8 9abcd ef ghiA  BCD  EFG HI XYZ”、加密间隔为9，先去掉输入中的空格后输出为：123456789abcdefghiABCDEFGHIXYZ，然后按间隔9分组：  
123456789  
abcdefghi  
ABCDEFGHI  
XYZ  
　　输出密码的方法为：从第1组开始，依次从每组各取1字符输出，每一轮取完后输出一个空格，则以上输入的密码输出为：1aAX 2bBY 3cCZ 4dD 5eE 6fF 7gG 8hH 9iI。  
　　编程可用素材：printf("Input a string:")...、printf("\nInput jiange:")...、printf("Output:\nThe string of deleted space:")...、printf("\nThe result is:")...。  
　　程序的运行效果应类似地如图1所示，图1中的1 2 3 4 5 6 7   8 9abcd ef ghiA  BCD  EFG HI XYZ和9是从键盘输入的内容。



#include <stdio.h>

#include <string.h>

void clearspace(char str[128]);

void prencode(char str[128], int encode);

int main(void)

{

int encode;

char str[128];

printf("Input a string:");

gets(str);

printf("Input jiange:");

scanf("%d", &encode);

clearspace(str);

printf("Output:\nThe string of deleted space:");

puts(str);

prencode(str, encode);

return 0;

}

void clearspace(char str[128])

{

int i, j;

for (i = 0, j = 0; str[i] !='\0'; i++)

{

if (str[i] != ' ')

{

str[j] = str[i];

j++;

}

}

str[j] = '\0';

}

void prencode(char str[128], int encode)

{

int i, j;

int length;

length = strlen(str);

printf("The result is:");

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

{

for (j = i; j < length; j += encode)

{

printf("%c", str[j]);

}

printf(" ");

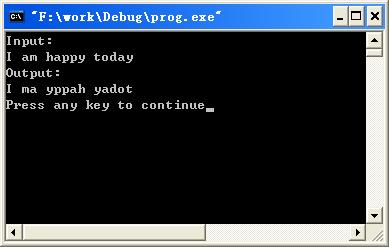
}

printf("\n");

}

要求：不改变单词的顺序的情况下，将每一个单词翻转（单词倒序写）。输入中只有英文单词和空格（不含其它字符）。

如图：其中I am happy today是键盘输入的。



#include <stdio.h>

#include <stdlib.h>

#include <string.h>

void result(char str1[], char str2[]);

void revert(char st1[], char st2[]);

int main(void)

{

char str[80], str1[80];

printf("Input:\n");

gets(str);

result(str, str1);

printf("Output:\n");

puts(str1);

return 0;

}

void result(char str1[], char str2[])

{

int i = 0, j = 0, flag = 0;

char st[80], st2[80];

strcpy(str2, "");

for (i = 0; str1[i] != '\0'; i++)

{

if (str1[i] != ' ')

{

flag = 1;

st[j] = str1[i];

j++;

}

if (str1[i] == ' ' && flag == 1)

{

flag = 0;

st[j] = '\0';

revert(st, st2);

strcat(str2, st2);

strcat(str2, " ");

j = 0;

continue;

}

if (str1[i] == ' ' && flag == 0)

{

strcat(str2, " ");

}

}

st[j] = '\0';

revert(st, st2);

strcat(str2, st2);

}

void revert(char st1[], char st2[])

{

int i;

int n;

n = strlen(st1);

st2[n] = '\0';

for (i = 0; st1[i] != '\0'; i++)

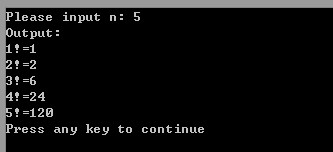
{

st2[n - 1 - i] = st1[i];

}

}

输出 1到n(12>=n>=1)的阶乘的值(要求利用静态局部变量相关的知识)。注意：只允许在/\*\*\*\*\*\*start\*\*\*\*\*\*/和/\*\*\*\*\*\*end\*\*\*\*\*\*/之间添加代码。



#include<stdio.h>

/\*\*\*\*\*\*start\*\*\*\*\*\*/

int fun(int n);

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

int main()

{

int i,n;

printf("Please input n: ");

scanf("%d",&n);

printf("Output:\n");

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

{

printf("%d!=%d\n",i,fun(i));

}

return 0;

}

/\*\*\*\*\*\*start\*\*\*\*\*\*/

int fun(int n)

{

static int ans=1;

ans=ans\*n;

return ans;

}

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