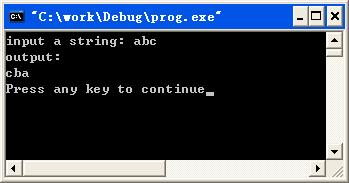
Wy5203完成以下代码中的函数声明和定义。该函数用于将字符串反序，如"abc"反序为"cba"。

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



#include <stdio.h>

#include <string.h>

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

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

int main()

{

char str[200]={0};

printf("input a string: ");

gets(str);

ReverseString(str);

puts("output:");

puts(str);

return 0;

}

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

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

void ReverseSTring(char str[]);

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];

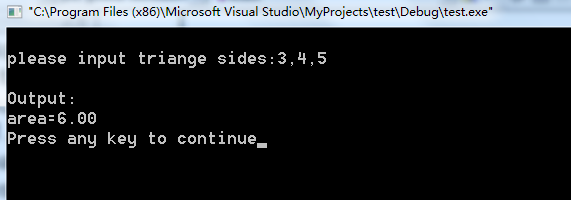
str[j] = ch;

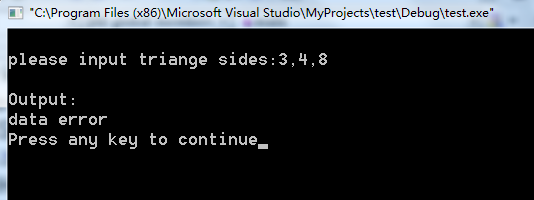
}

}

D5201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\*\*\*\*\*\*/

/\*\*\*\*\*\*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\*\*\*\*\*\*/

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

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

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

{

double p;

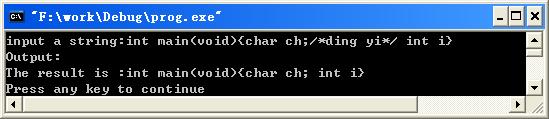
p = (a+b+c)/2.0;

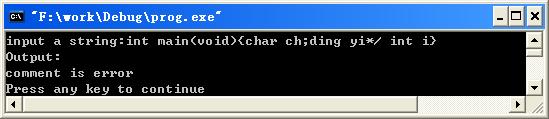
return sqrt(p\*(p-a)\*(p-b)\*(p-c));

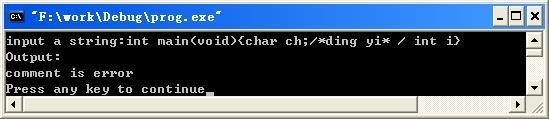
}

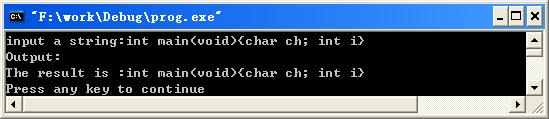
LB5303从键盘上读入一行字符(约定：字符数≤127字节)，判断其中的注释是否合法，不合法则报错，合法时则删除注释后再输出。合法注释是指“/\*”标记注释开始、“\*/”标记注释结束，通常表现为/\* ……\*/。  
　　注意事项：  
(1)只考虑行内最多只包含一个注释的情况。  
(2)不合法的注释情况有很多种，例如 ……\*/缺注释开始标记、/  \* ……\*/缺注释开始标记、/\* ……缺注释结束标记、/\* ……\*  /缺结束标记。  
(3)编程可用素材：printf("input a string:")...、printf("Output:\nThe result is :")...、printf("Output:\ncomment 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;

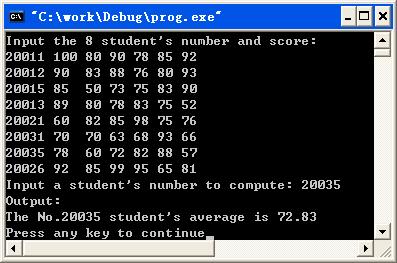
}

}

return (-1);

}

LB5201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\*\*\*\*\*\*/

/\*\*\*\*\*\*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\*\*\*\*\*\*/

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

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

{

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

{

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

}

}

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;

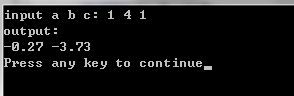
}

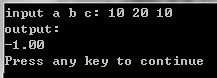
Wy5201

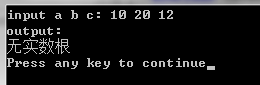
写一个求二次方程ax2+bx+c根的函数。在主函数中首先由键盘输入a、b、c，然后调用求根的函数resolve()求方程的根并输出。需注意b2-4ac大于、等于或小于0的情况。

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\*\*\*\*\*\*/

/\*\*\*\*\*\*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\*\*\*\*\*\*/

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

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

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

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

/\*\*\*\*\*\*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\*\*\*\*\*\*/

LB5302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(" ");

}

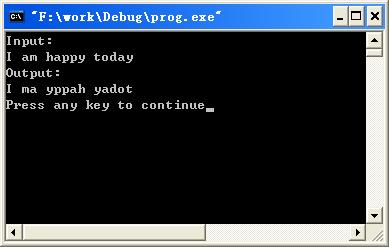
printf("\n");

}

LB5305

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];

}

}