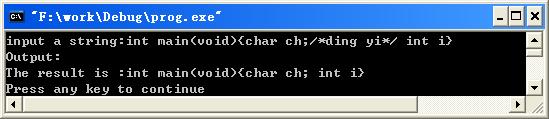
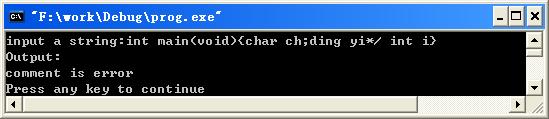
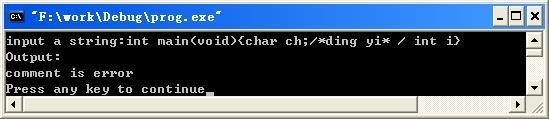
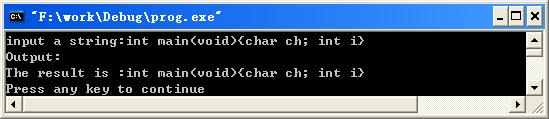
LB5303从键盘上读入一行字符(约定：字符数≤127字节)，判断其中的注释是否合法，不合法则报错，合法时则删除注释后再输出。合法注释是指“/\*”标记注释开始、“\*/”标记注释结束，通常表现为/\* ……\*/。  
　　注意事项：  
(1)只考虑行内最多只包含一个注释的情况。  
(2)不合法的注释情况有很多种，例如 ……\*/缺注释开始标记、/  \* ……\*/缺注释开始标记、/\* ……缺注释结束标记、/\* ……\*  /缺结束标记。  
(3)编程可用素材：printf("input a string:")...、printf("Output:\nThe result is :")...、printf("Output:\ncomment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

}

}

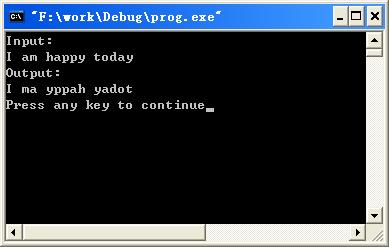
return (-1);

}

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];

}

}

XH4306题目描述

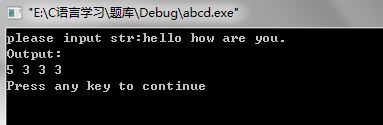
编一个程序，读入用户输入的，以“.”结尾的一行文字，统计一共有多少个单词，并分别输出每个单词含有多少个字符。  
（凡是以一个或多个空格隔开的部分就为一个单词，不判断输入是否以"."结尾，单词中包含的数字也要统计）

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

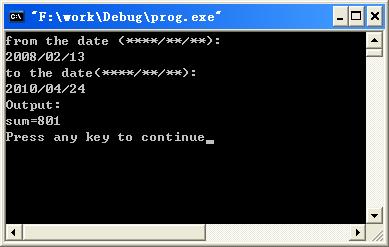
}

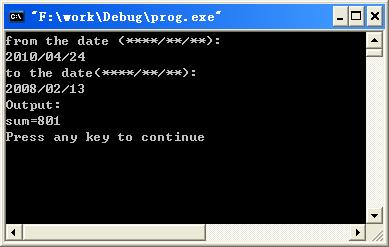
putchar('\n'); // 别忘了换行

return 0;

}

LB3301计算两个日期之间的天数。注：判断年份是否为闰年的方法——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(year2, 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);

}