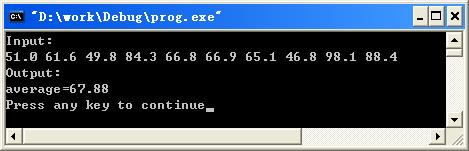
LB4106从键盘输入10个学生成绩（double），计算并输入这10个学生的平均成绩（保留小数点后2位）。



#include<stdio.h>

int main()

{

double a[10];

int i;

double average,sum=0;

printf("Input:\n");

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

{

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

}

printf("Output:\n");

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

{

sum=sum+a[i];

}

printf("average=%.2lf",sum/10);

}

XH4206题目描述

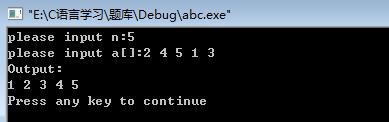
对输入的n个数进行排序并输出。

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;

}

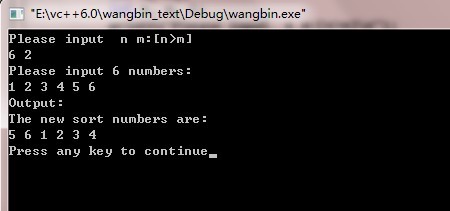
WB4109输入n个整数，使前面的n-m个数顺序向后移动m个位置，最后的m个数变成最前面的m个数。

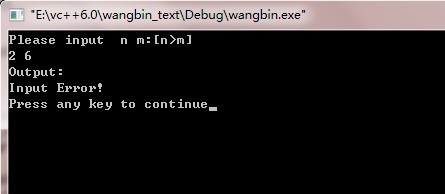
输入格式

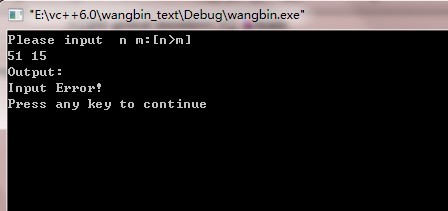
第一行包含两个用空格隔开的整数n和m，保证n>m且n不超过50，如果输入不符合以上要求，要有“Input Error！”提示。第二行包含n个用空格隔开的整数。

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--)

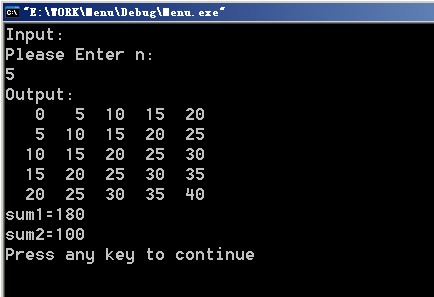
x[i + m] = x[i];

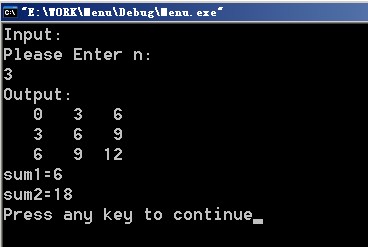
for (i = 0;i < m;i++)

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

}

D4203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;

}

XH4305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')。

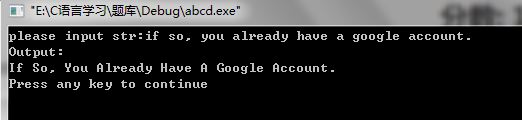
(注：程序中的字符串输入输出用gets和puts）。

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");

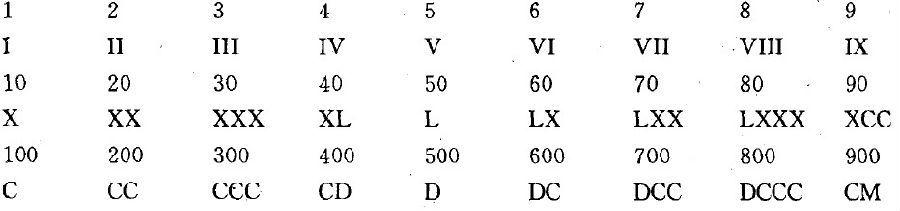
puts(str+1); // 注意这里输出也是从第一个位置开始，而非是字符串的起始位置

return 0;

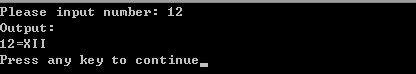
}

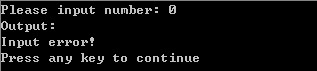
HE4205输入：阿拉伯数组：n  （0<n<1000）  
输出：翻译后的罗马数字（注：罗马数字规定用大写字母代替书写）

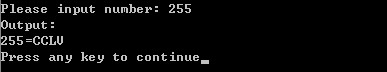
对应数字表：



程序效果如下图：







#include<stdio.h>

int main()

{

static char \*a[][10]={"","I","II","III","IV","V","VI","VII","VIII","IX",

"","X","XX","XXX","XL","L","LX","LXX","LXXX","XCC",

"","C","CC","CCC","CD","D","DC","DCC","DCCC","CM"

};

int n,t,i,m;

printf("Please input number: ");

scanf("%d",&n);

printf("Output:\n");

if(n<=0 || n>=1000)

{

printf("Input error!\n");

return 0;

}

printf("%d=",n);

for(m=0,i=1000;m<3;m++,i/=10)

{

t=(n%i)/(i/10);

printf("%s",a[2-m][t]);

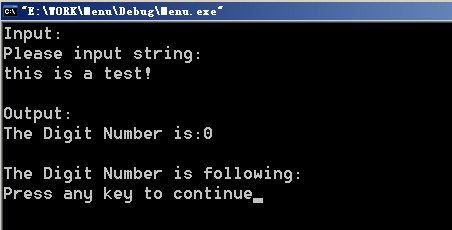
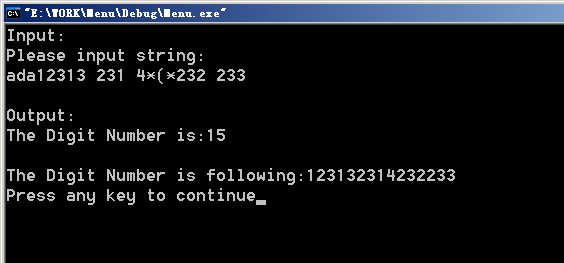
}

putchar(10);

return 0;

}

D4202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);

}

XH4306题目描述

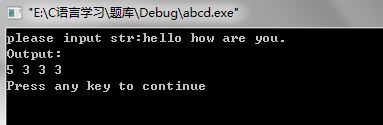
编一个程序，读入用户输入的，以“.”结尾的一行文字，统计一共有多少个单词，并分别输出每个单词含有多少个字符。  
（凡是以一个或多个空格隔开的部分就为一个单词，不判断输入是否以"."结尾，单词中包含的数字也要统计）

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;

}