改错题

以下程序的功能是：输入圆的半径r和运算标志m后，按照运算标志进行如下指定的计算。

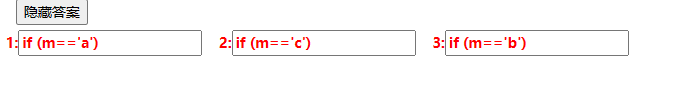
运算标志m计算内容

a面积

c周长

b二者均计算（面积和周长）

#include <stdio.h>  
#define pi 3.14159  
int main()  
{  
 char m;  
 float r, c, a;  
 printf("input mark a c or b && r\n");  
 scanf("%c%f", &m, &r);  
 /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
 if (\_\_\_\_\_)  
 {  
  a = pi \* r \* r;  
  printf("area is %f\n", a);  
 }  
 /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
 if (\_\_\_\_\_)  
 {  
  c = 2 \* pi \* r;  
  printf("circle is %f\n", c);  
 }  
 /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
 if (\_\_\_\_\_)  
 {  
  a = pi \* r \* r;  
  c = 2 \* pi \* r;  
  printf("area && circle are %f %f\n", a, c);  
 }  
 return 0;  
}



计算一个长整型数(最大不能超过2147483647、最小不能小于999)的最大三位数约数。

#include <stdio.h>

#define MAX 2147483647

int main(void)  
{  
 long i;  
 int j;  
  
 printf("请输入一个长整型数:");  
 do  
 {  
  /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
  scanf("%1f", &i);   
                /\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*/

  if (i>MAX || i<999)  
  {  
   printf("数据输入错误，请重新输入:");  
  }  
 } while (i>MAX || i<999);

 for (j=999; j>=100; j--)  
 {  
  if (i%j == 0)  
  {  
   printf("%ld的最大的三位数约数是:%d\n", i, j);  
   /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
   continue;   
                        /\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*/

  }  
 }  
  
 return 0;  
}



搜索神秘的三位数——有这样一个3位数，组成它的3个数字阶乘之和正好等于它本身，即：abc = a! + b! + c!，例如145 = 1! + 4! + 5!。

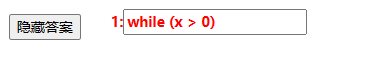
#include <stdio.h>

int main(void)  
{  
 int JC[] = {1, 1, 2, 6, 24, 120, 720, 5040, 40320, 362880};  
 int i, sum, x;

 for (i=100; i<1000; i++)  
 {  
  sum = 0;  
  x = i;  
  /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
  while (\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

  {  
   sum += JC[x%10];  
   x /= 10;  
  }  
  if (i == sum)  
  {  
   printf("%d\n",i);  
  }  
 }

 return 0;  
}



程序功能：计算1/1 + 1/2 + … + 1/N之和并显示。注意：只修改/\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/下面一行。

#include<stdio.h>

#define N 10

int main(void)  
{  
        /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
        long jieguo = 0;  
        int fm;

        for (fm = 1; fm <= N; fm++)  
        {  
                /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
                jieguo += 1 / fm;  
        }  
        printf("%8.6f\n", jieguo);

 return 0;  
}



从键盘读入a和b的值(输入格式为a,b)，在屏幕上显示b行a。

#include<stdio.h>

int main(void)  
{  
        int a, b;

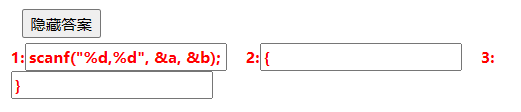
        /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
        scanf("%d%d", &a, &b);  
        /\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*/

        while (b > 0)  
        /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
        \_\_\_\_\_\_\_\_\_\_\_  
        /\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*/

                printf("%d\n", a);  
                b--;  
        /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
        \_\_\_\_\_\_\_\_\_\_\_  
        /\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*/

        printf("bye bye!\n");

 return 0;  
}



显示用1、2、3、4个数字组成的互不相同且无重复数字的三位数。

#include<stdio.h>

int main(void)  
{  
 int i, j, k;

 /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
 for (i=1;\_\_\_\_ ; i++)  
        /\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*/

 {  
  for (j=1; j<5; j++)  
  {  
   for (k=1; k<5; k++)  
   {  
    /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
    if (i!=k && i!=j && \_\_\_\_\_\_)  
                                /\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*/

    {  
     printf("%d%d%d ", i, j, k);  
    }  
   }  
  }  
 }

 return 0;  
}



从输入数据中统计正整数和负整数的个数。用输入0来结束输入，变量i存放正整数个数，变量j存放负整数的个数。

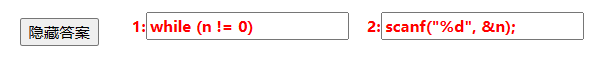
#include<stdio.h>

int main(void)  
{  
 int i = 0, j = 0, n;

 printf("\ninput numbers:");  
 scanf("%d", &n);  
 /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
 while (\_\_\_\_\_\_\_\_\_\_\_\_\_\_)  
        /\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*/

 {  
  if (n > 0)  
  {  
   i++;  
  }  
  if (n < 0)  
  {  
   j++;  
  }  
  /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
                /\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*/  
 }  
 printf("i=%d,j=%d", i, j);

 return 0;  
}



将任意十进制正小数分别转换成2,3,4,5,6,7,8,9进制正小数，小数点后保留8位，并输出。例如：若十进制小数为0.795，则输出：  
　　　　十进制正小数 0.795000 转换成 2 进制数为: 0.11001011  
　　　　十进制正小数 0.795000 转换成 3 进制数为: 0.21011011  
　　　　十进制正小数 0.795000 转换成 4 进制数为: 0.30232011  
　　　　十进制正小数 0.795000 转换成 5 进制数为: 0.34414141  
　　　　十进制正小数 0.795000 转换成 6 进制数为: 0.44341530  
　　　　十进制正小数 0.795000 转换成 7 进制数为: 0.53645364  
　　　　十进制正小数 0.795000 转换成 8 进制数为: 0.62702436  
　　　　十进制正小数 0.795000 转换成 9 进制数为: 0.71348853

#include <stdio.h>

void fun(double dTestNo, int iBase);

int main(void)  
{    
 double dTestNo = 0.795;  
 int iBase;  
  
 for (iBase=2; iBase<=9; iBase++)  
 {  
  fun(dTestNo, iBase);  
 }  
 printf("\n");  
  
 return 0;  
}

void fun(double dTestNo, int iBase)  
{  
 int iT[8], iNo;  
  
 printf("十进制正小数 %f 转换成 %d 进制数为:", dTestNo, iBase);  
 for (iNo=0; iNo<8; iNo++)  
 {  
  dTestNo \*= iBase;  
  /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
  iT[iNo] = \_\_\_\_\_\_\_\_\_\_\_\_\_\_;

  /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
  dTestNo -= \_\_\_\_\_\_\_\_\_\_\_\_\_\_;

 }  
  
 printf("0.");  
 for (iNo=0; iNo<8; iNo++)  
 {  
  printf("%d", iT[iNo]);  
 }  
 printf("\n");  
}

函数



iT[iNo] = (int)dTestNo; dTestNo -= iT[iNo];

定义一个整型数组，采用子函数实现数据输入。

#include<stdio.h>

void input(int a[], int n);

int main(void)  
{  
 int a[10], i;

 /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
 input(a[10], 10);  
        /\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*/

 for (i=0; i<10; i++)  
 {  
  printf("%d ", a[i]);  
 }  
 return 0;  
}

void input(int a[], int n)  
{  
 int i;

 for (i=0; i<n; )  
 {  
  scanf("%d", &a[i]);  
  /\*\*\*\*\*\*\*\*\*Found\*\*\*\*\*\*\*\*\*\*\*\*/  
  i--;  
                /\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*/  
 }  
}

