## 

## 源代码：

#include <stdio.h>

struct student

{

double average;

char name[10];

int IDnumber;

char gender[10];

int course\_1;

int course\_2;

int course\_3;

int course\_4;

int course\_5;

};

int main()

{

struct student student\_1; //具体化结构体变量

printf("Please input the student's name: \t"); //输入信息

scanf("%s", &student\_1.name);

printf("Please input the student's gender: \t");

scanf("%s", &student\_1.gender);

printf("Please input the student's ID number: \t");

scanf("%d", &student\_1.IDnumber);

printf("Please input the student's course\_1: \t");

scanf("%d", &student\_1.course\_1);

printf("Please input the student's course\_2: \t");

scanf("%d", &student\_1.course\_2);

printf("Please input the student's course\_3: \t");

scanf("%d", &student\_1.course\_3);

printf("Please input the student's course\_4: \t");

scanf("%d", &student\_1.course\_4);

printf("Please input the student's course\_5: \t");

scanf("%d", &student\_1.course\_5);

student\_1.average = (student\_1.course\_1 + student\_1.course\_2 + student\_1.course\_3 + student\_1.course\_4 + student\_1.course\_5) / 5; //计算平均值

printf("\n\nThe student's name is %s \n", student\_1.name); //输出名字

printf("The student's gender is %s \n", student\_1.gender); //输出性别

printf("The student's ID number is %d \n", student\_1.IDnumber); //输出学号

printf("The student's course\_1 score is %d \n", student\_1.course\_1); //输出科目1

printf("The student's course\_2 score is %d \n", student\_1.course\_2); //输出科目2

printf("The student's course\_3 score is %d \n", student\_1.course\_3); //输出科目3

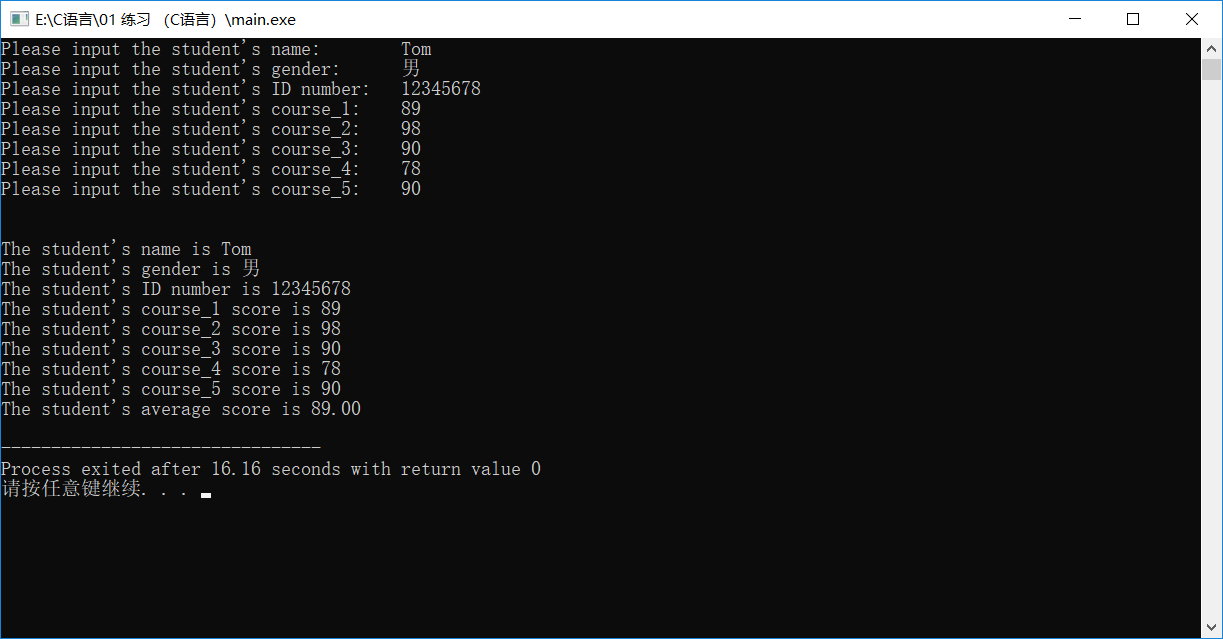
printf("The student's course\_4 score is %d \n", student\_1.course\_4); //输出科目4

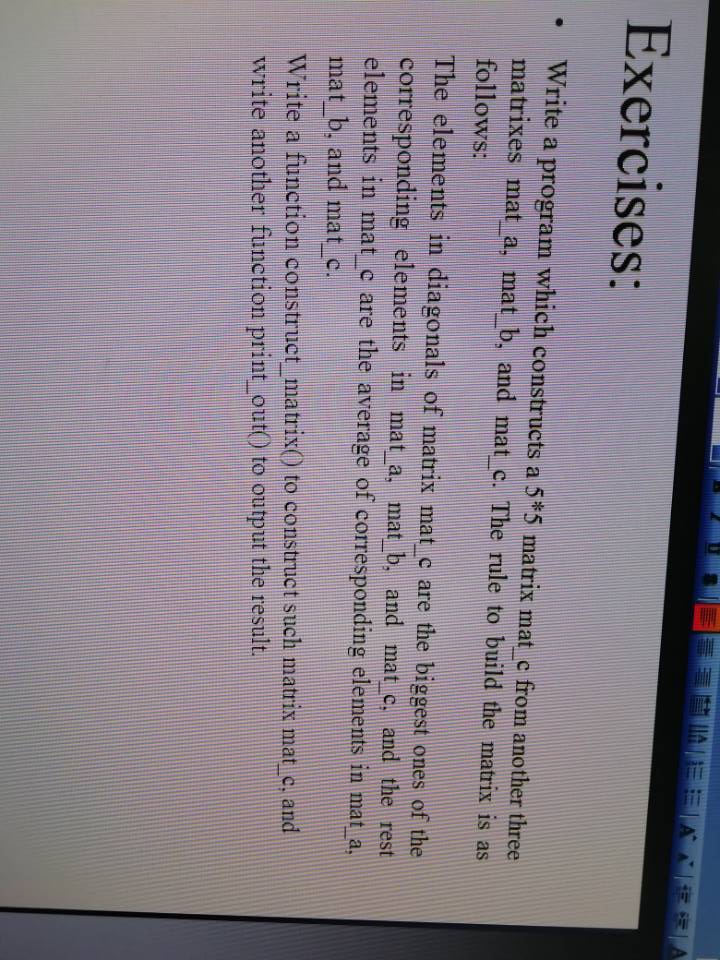
printf("The student's course\_5 score is %d \n", student\_1.course\_5); //输出科目5

printf("The student's average score is %.2lf \n", student\_1.average); //输出平均分

return 0;

## }输出结果：





## 源代码：

#include <stdio.h>

void construct\_matrix(double mat\_a[][5], double mat\_b[][5], double mat\_c[][5]);

void print\_out(double mat\_c[][5]);

int main()

{

//int mat\_a[5][5] = { 0 }, mat\_b[5][5] = { 0 }, mat\_c[5][5] = { 0 };

int i, j;

double mat\_a[5][5] = { 0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4 }; //设置多组实验数据方便调试

double mat\_b[5][5] = { 0,9,8,7,6,5,4,3,2,1,0,9,8,7,6,5,4,3,2,1,0,9,8,7,6 };

double mat\_c[5][5] = { 1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5 };

printf("\nmat\_a矩阵为：\n"); //展示刚才输入的矩阵mat\_a

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

{

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

{

printf("%3.0lf", mat\_a[i][j]);

}

printf("\n");

}

printf("\nmat\_b矩阵为：\n"); //展示刚才输入的矩阵mat\_b

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

{

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

{

printf("%3.0lf", mat\_b[i][j]);

}

printf("\n");

}

printf("\nmat\_c矩阵为：\n"); //展示刚才输入的矩阵mat\_c

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

{

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

{

printf("%3.0lf", mat\_c[i][j]);

}

printf("\n");

}

construct\_matrix(mat\_a, mat\_b, mat\_c); //调用操作函数

print\_out(mat\_c); //调用输出函数

return 0;

}

void construct\_matrix(double mat\_a[][5], double mat\_b[][5], double mat\_c[][5])

{

int i, j;

double max;

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

{

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

{

if (i == j) //如果在对角线上，i==j，找最大的

{

if (mat\_a[i][i] > mat\_b[i][i])

max = mat\_a[i][i];

else

max = mat\_b[i][i];

if (mat\_c[i][i] < max)

mat\_c[i][i] = max;

}

else //不是对角线上的，求平均数

{

mat\_c[i][j] = (mat\_a[i][j] + mat\_b[i][j] + mat\_c[i][j]) / 3;

}

}

}

}

void print\_out(double mat\_c[][5]) //输出新矩阵mat\_c

{

int i, j;

printf("\n\n新的mat\_c矩阵为：\n");

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

{

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

{

printf("%3.0lf", mat\_c[i][j]);

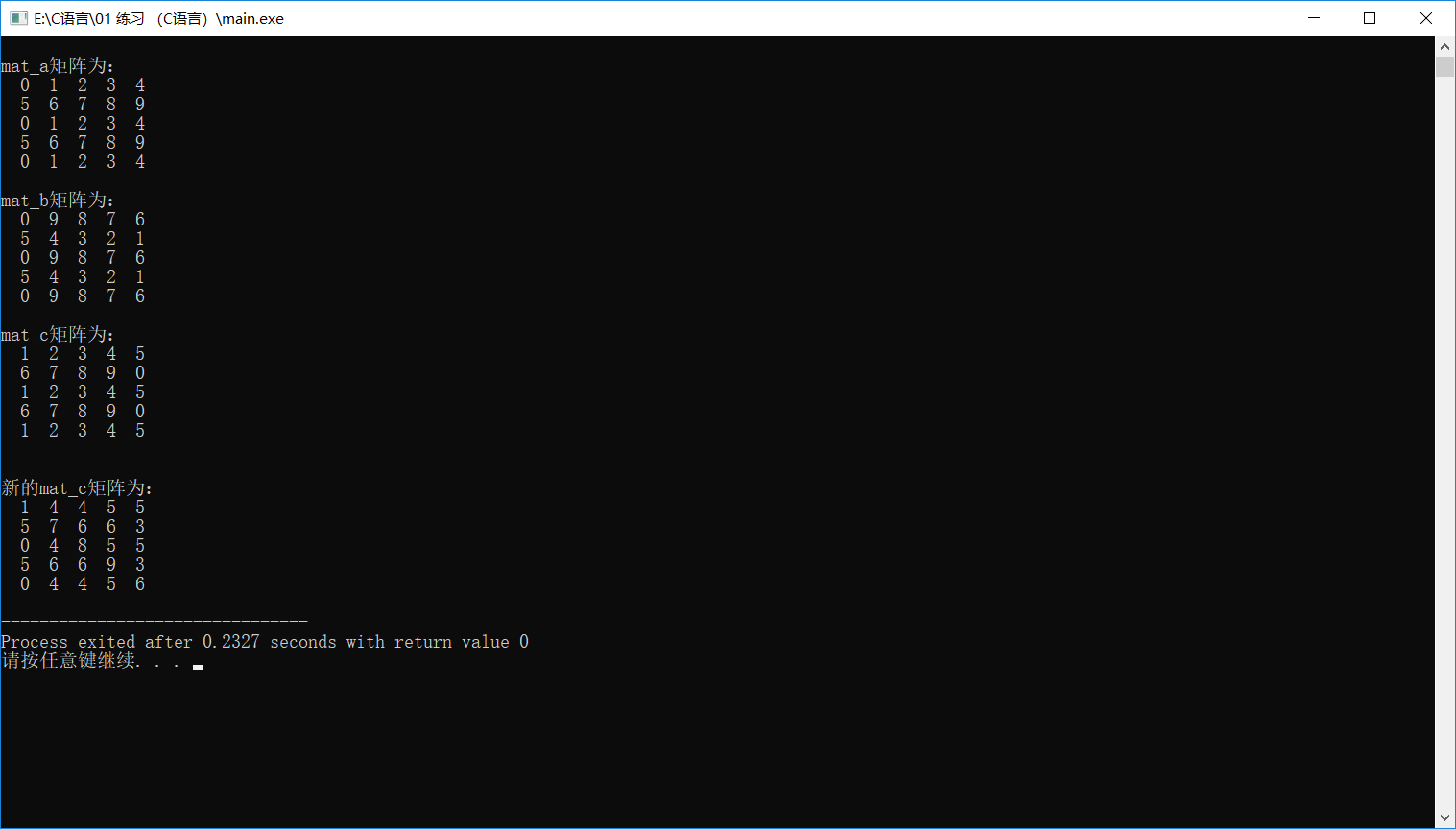
}

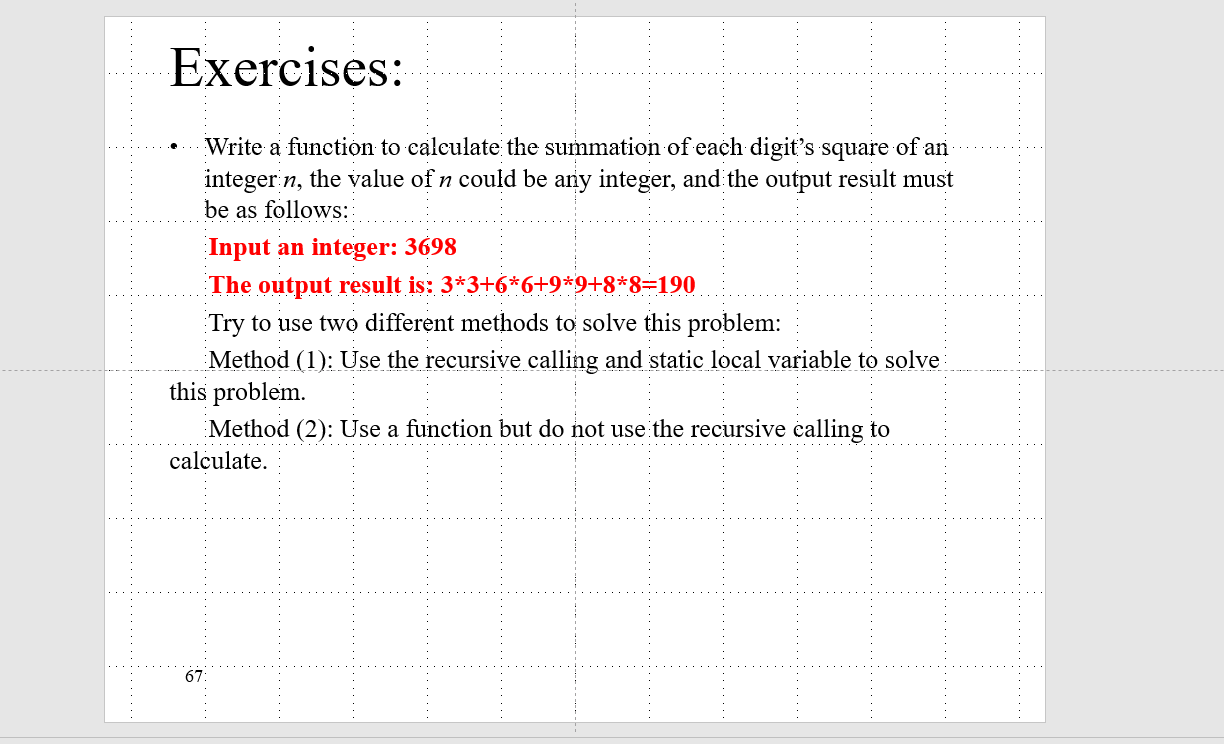
printf("\n");

}

}

## 输出结果：





## 源代码：

#include <stdio.h>

#include <math.h>

int separate(int n,int length);

int main()

{

int OriginalNumber, number, length; //定义

printf("Input an integer: ");

scanf("%d", &number); //输入数字，并存在number中

OriginalNumber = number; //复制数字，并存在OriginalNumber中，在主函数中进行数据处理

while (number>0) //用while循环，算数值的位数

{

number /= 10;

length++;

}

printf("%d", separate(OriginalNumber, length)); //调用函数，并输出sum

return 0;

}

int separate(int n , int length) //分离数字，并逐位平方

{

int i,j=0;

static int sum; //固定变量sum，计算平方和

if ((i = n / 10) != 0) //递归，分离数字

{

separate(i,length);

}

if(n>pow(10,length-1)) //如果是最后一位，输出“=”号

printf("%d\*%d=", n % 10, n % 10);

else

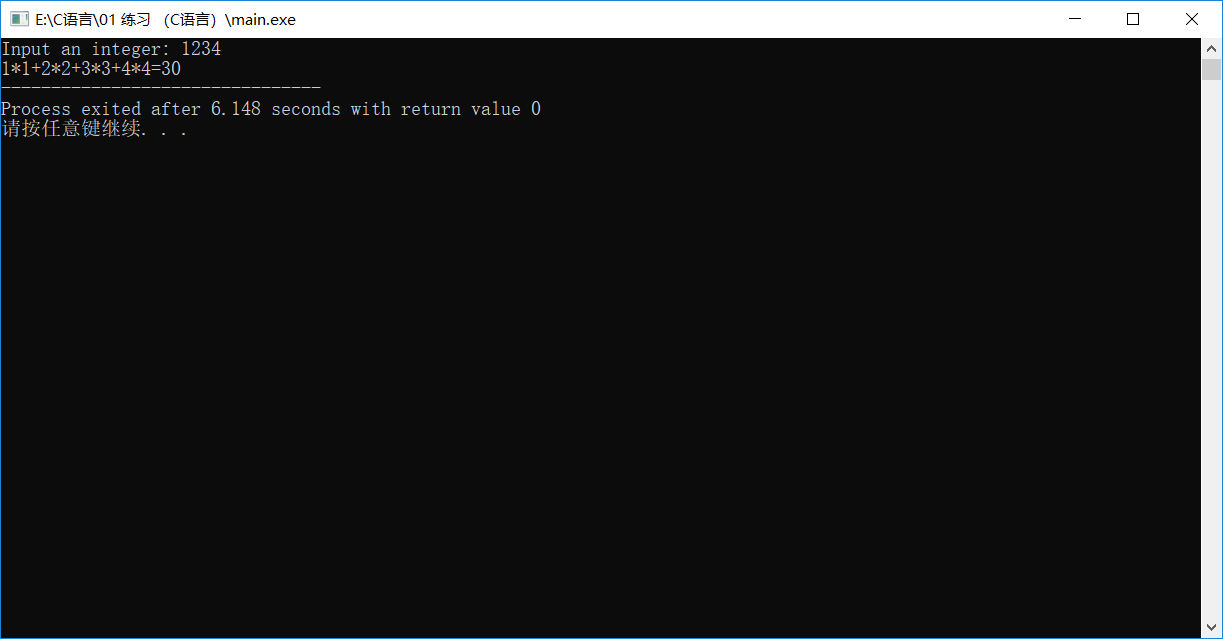
printf("%d\*%d+", n % 10, n % 10); //前几位，输出“+”号

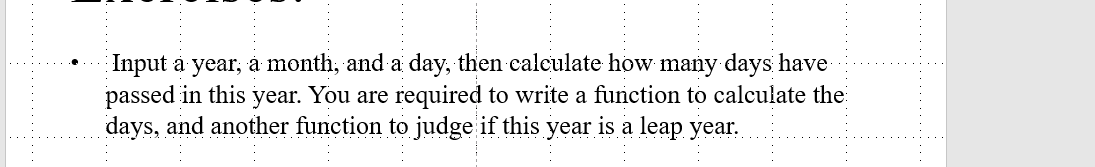
sum += (n % 10)\*(n % 10); //计算sum，计算平方和

return sum; //返回sum

}

## 输出结果：





## 源代码：

#include <stdio.h>

int main()

{

int year, month, day, leap, days\_number = 0, a[12] = { 31,28,30,30,31,30,31,31,30,31,30,31 }, i;

//输入数据

printf("请输入年份：");

scanf("%d", &year);

printf("请输入月份：");

scanf("%d", &month);

printf("请输入日期：");

scanf("%d", &day);

leap = leap\_year(year); //判断闰年

if (leap == 1)

{

a[1] = 29; //如果是闰年，二月改为29天

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

{

days\_number += a[i];

}

days\_number += day;

}

else

{

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

{

days\_number += a[i];

}

days\_number += day;

}

printf("今年过了%d天。", days\_number); //输出语句

return 0;

}

int leap\_year(int year) //判断闰年

{

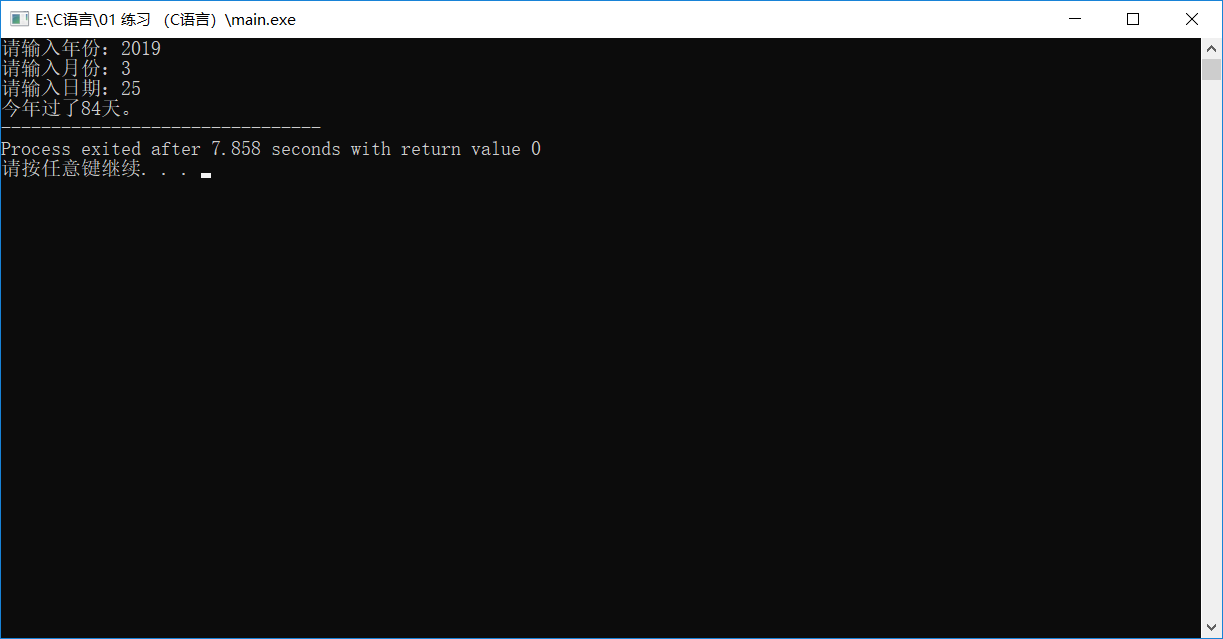
if (year % 400 == 0 || year % 4 == 0 && year % 100 != 0)

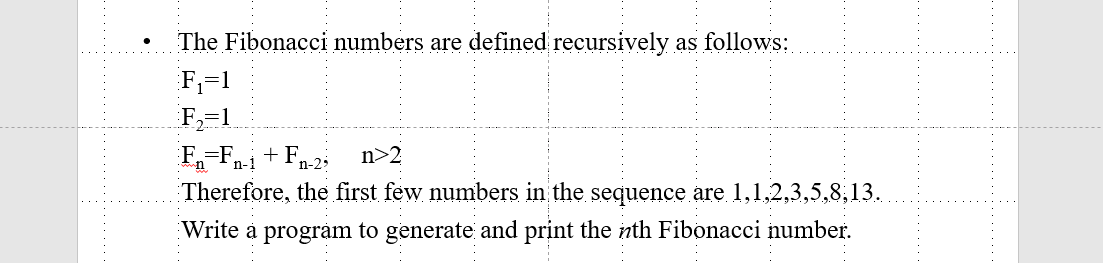
return 1;

else

return 0;

## }输出结果：





## 源代码：

#include <stdio.h>

int Fibonacci(int i);

int main()

{

int n, i;

printf("请输入需要几位斐波那契数列：");

scanf("%d", &n);

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

{

printf("Fibonacci of %d = %d\n", i + 1, Fibonacci(i));

}

return 0;

}

int Fibonacci(int i)

{

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

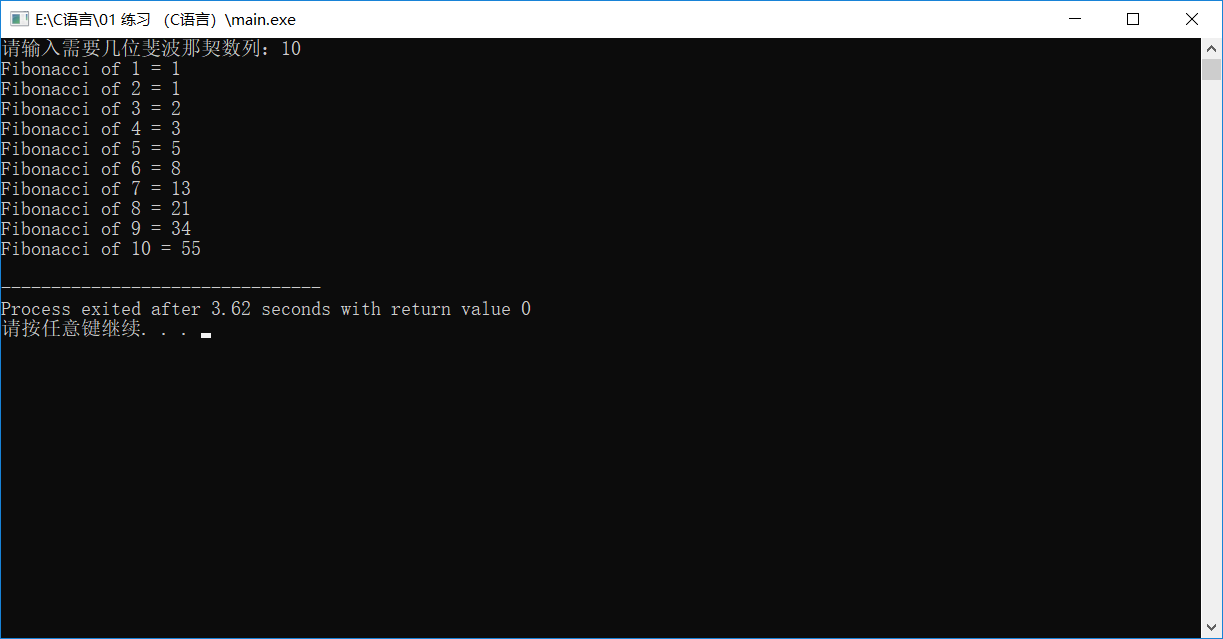
return 1;

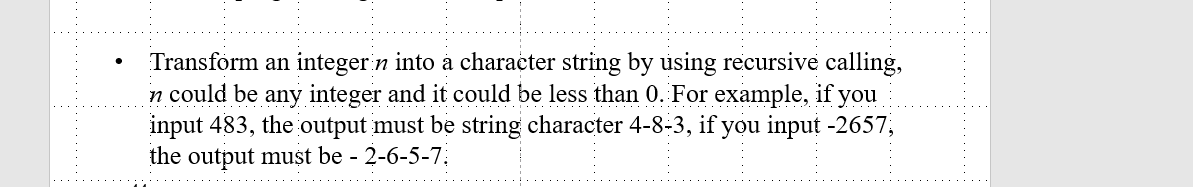
else

return Fibonacci(i - 1) + Fibonacci(i - 2);

}

## 输出结果：





## 源代码：

#include <stdio.h>

#include <string.h>

void transform(int number);

int main()

{

int number; //定义要分离的数字

printf("Please input an integer:\t");

scanf("%d", &number);

transform(number); //函数用-分离数字

return 0;

}

void transform(int number) //递归

{

int i;

if ((i=number/10) != 0) //如果这个数不是一位数，除10取商

{

transform(i); //递归引用

}

if(number<0)

printf("%d", number % 10); //如果是一个负数，前面自带“-”号。

else

{

if(number<10) //如果是正数，第一位不加“—”号，其他位加“—”号

printf("%d", number % 10); //第一位小于10

else

printf("-%d", number % 10);

}

}

## 输出结果：

