1. **递归程序设计**
2. 题目分析

用递归方法设计下列各题，并给出每道题目的递归出口（递归结束的条件）和递归表达式。同时考虑题目可否设计为非递归方法，如果可以，设计出非递归的算法。

1.一个人赶着鸭子去每个村庄卖，每经过一个村子卖去所赶鸭子的一半又一只。这样他经过了七个村子后还剩两只鸭子，问他出发时共赶多少只鸭子？经过每个村子卖出多少只鸭子？

2.角谷定理。输入一个自然数，若为偶数，则把它除以2，若为奇数，则把它乘以3加1。经过如此有限次运算后，总可以得到自然数值1。求经过多少次可得到自然数1。

如：输入22，

输出 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

STEP=16

1. 算法构造
   1. 赶鸭子问题

2.1.1题目分析：

设在经过个n村子时有m只鸭子，根据题意可以得到如下递推公式：

则在第n+1个村子时有M(n)-(M(n)/2+1)=M(n)/2-1只鸭子，即M(n+1)=M(n)/2-1,

所以M(n-1)=(M(n)+1)\*2

在经过第7个村子书还剩下2只鸭子，即M(7)=2

2.1.2算法构造：

根据上述公式可以看出：

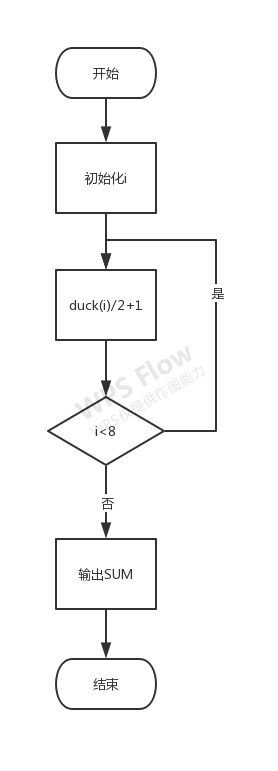
函数出口为： M(7)=2

函数体为：M(n-1)=(M(n)+1)\*2

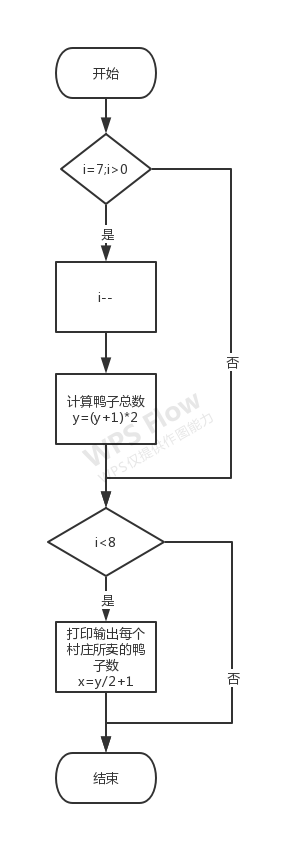
用最后的鸭子数倒推上一步的鸭子数，再以此类推，在经过最后一个村子时只剩下2两只鸭子，每递归一次经过的村次数就减少一个，直到最后为0即跳出函数。

2.1.3流程图设计

1. 递归流程图



1. 非递归流程图



2.2 角谷定理

2.2.1题目分析：

输入一个自然数，若为偶数，则把它除以2，若为奇数，则把它乘以3加1，并求出得到自然数为1的时候的步数，构造函数fun(x,y),其中x是输入的自然数，y为累积步数的计数器

2.2.2算法构造：

函数出口为：x=1

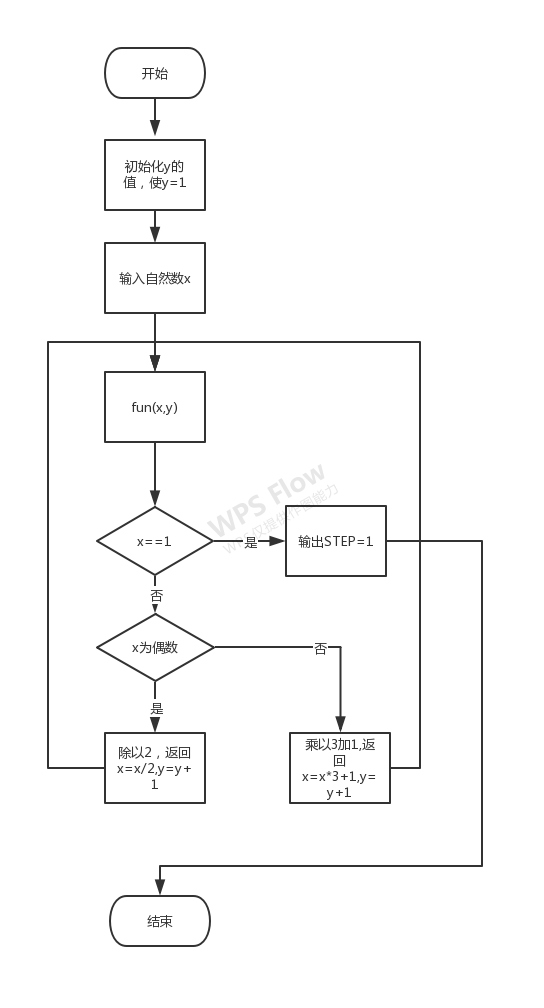
函数体为：

偶数：fun(i/2,j+1)

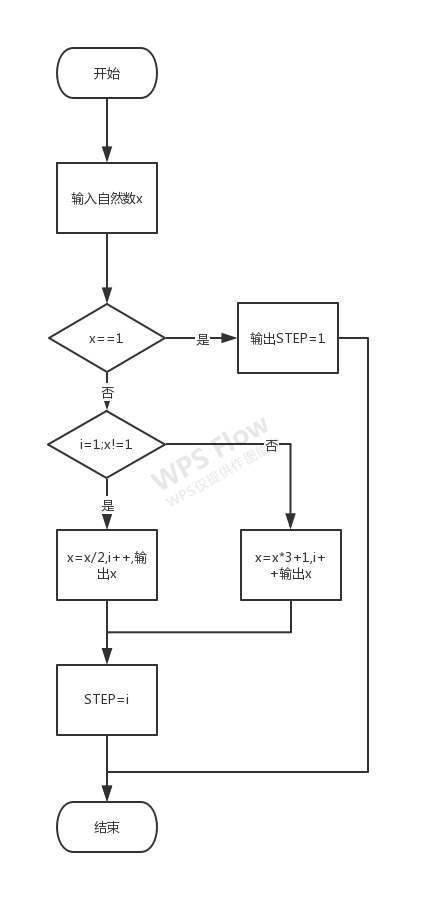
奇数：fun(i\*3+1,j+1)

2.2.3流程图设计

1. 递归流程图



1. 非递归流程图



1. 算法实现

3.1赶鸭子问题

3.1.1递归算法

#include<stdio.h>

#include<stdlib.h>

int duck(int i)

{

if(i==8)

return 2;//根据题意，当这个人到第8个村子时，只有2只鸭子，也是递归出口

else

return 2\*(duck(i+1)+1);//递归体

}

int main()

{

int i=1;

printf("出发时共有%d只鸭子\n",duck(1));

for(i=1;i<8;i++)//循环到每个村子时有多少鸭子，在每个村子卖出了多少鸭子

{

printf("他在第%d个村子时有%d只鸭子，卖出了%d只鸭子\n",i,duck(i),duck(i)/2+1);

}

system("pause");

}

3.1.2非递归算法

#include<stdio.h>

#include<stdlib.h>

int main()

{

int x,j;

int y=2,i=7;

for(i=7;i>0;i--)

{

y=(y+1)\*2;//根据题意可得公式

}

for(j=1;j<8;j++)//循环算到每个村子时有多少鸭子，在每个村子卖出了多少鸭子

{

printf("他在第%d村子有%d只鸭子,",j,y);

x=y/2+1;

y=y-x;

printf("卖了%d只鸭子\n",x);

}

system("pause");

}

3.2角谷定理

3.2.1递归算法

#include<stdio.h>

#include<stdlib.h>

int fun(int i,int j)

{

if(i==1)//递归出口

{

printf("STEP=%d\n",j);

return 1;

}

else//递归体

{

if(i%2==0)

{

printf("%d ",i/2);

return fun(i/2,j+1);

}

else

{

printf("%d ",i\*3+1);

return fun(i\*3+1,j+1);

}

}

}

int main()

{

int x;

int y=1;

scanf("%d",&x);

printf("%d ",x);

fun(x,y);

system("pause");

}

3.2.2非递归算法

#include<stdio.h>

#include<stdlib.h>

int main()

{

int x,y,i,j;

scanf("%d",&x);

if(x==1)

printf("STEP=1");

else

{

for(i=1;x!=1;i++)

{

if(x%2==0)

{

x=x/2;

printf("%d ",x);

}

else

{

x=x\*3+1;

printf("%d ",x);

}

}

printf("STEP=%d\n",i);

system("pause");

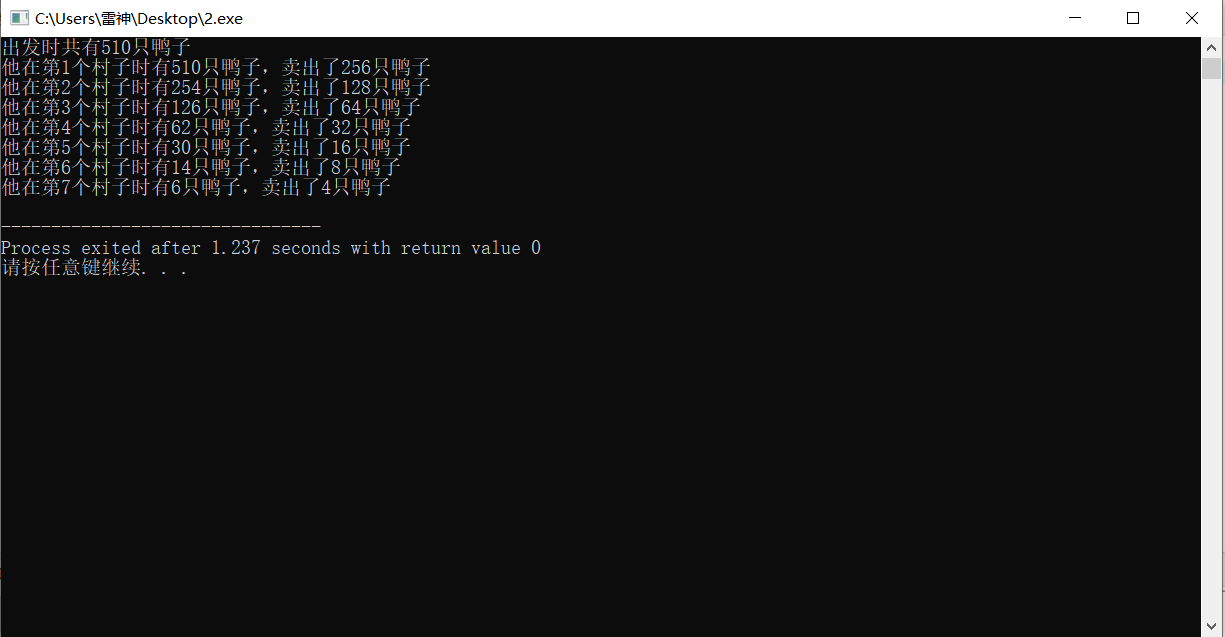
}

}

1. 调试、测试及运行结果

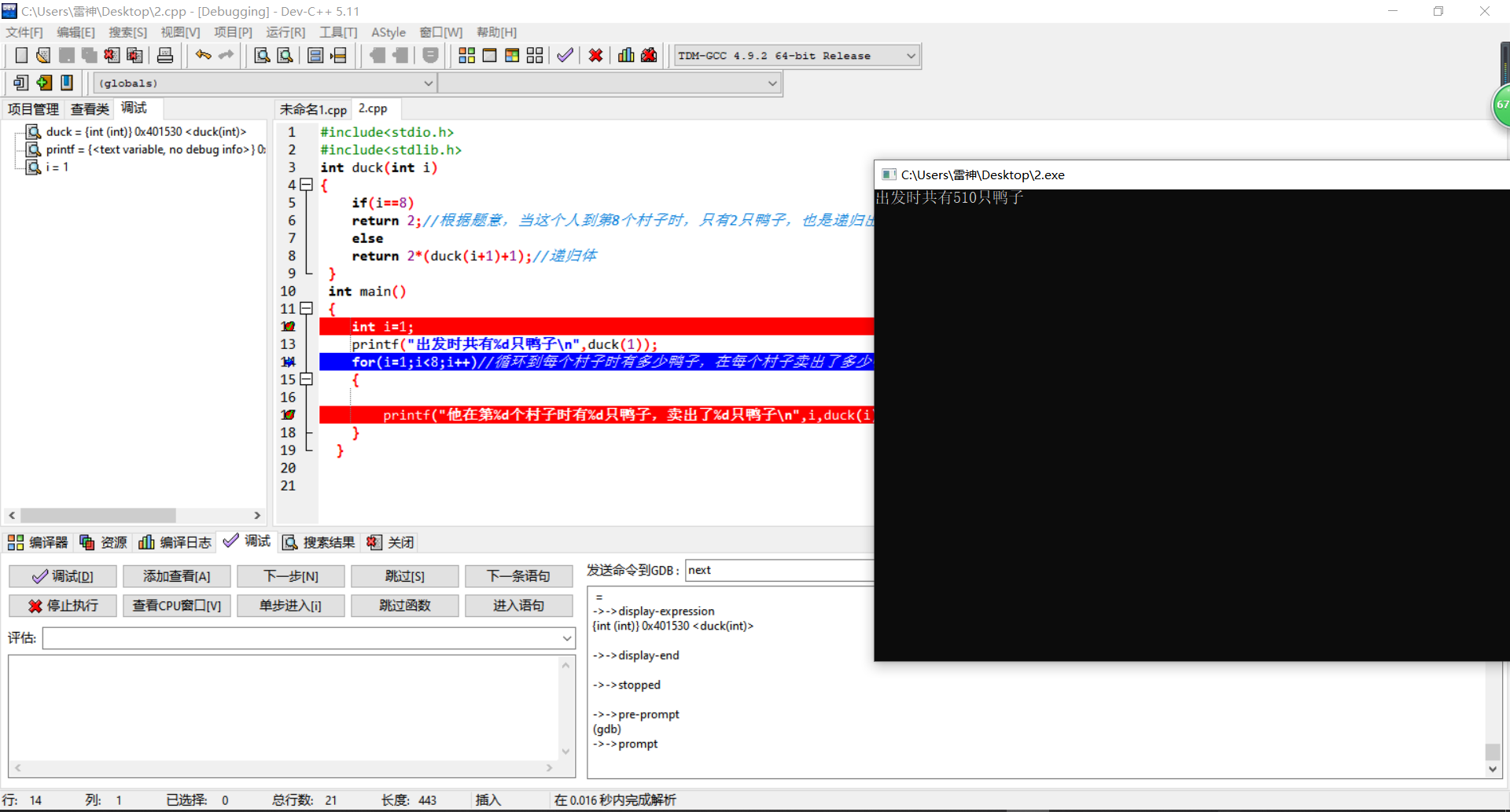
4.1赶鸭子问题

4.1.1运行结果

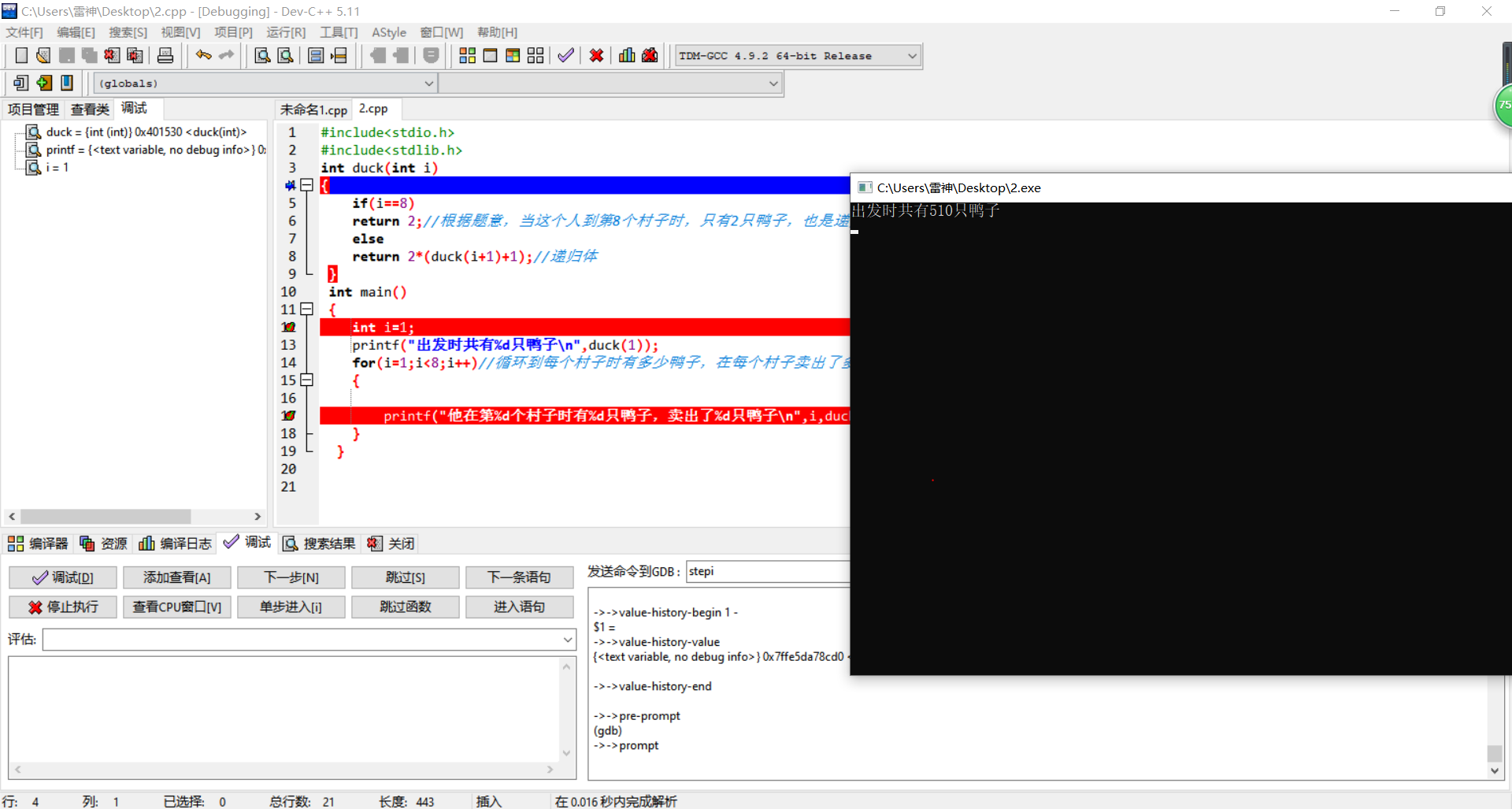


4.1.2递归调试截图

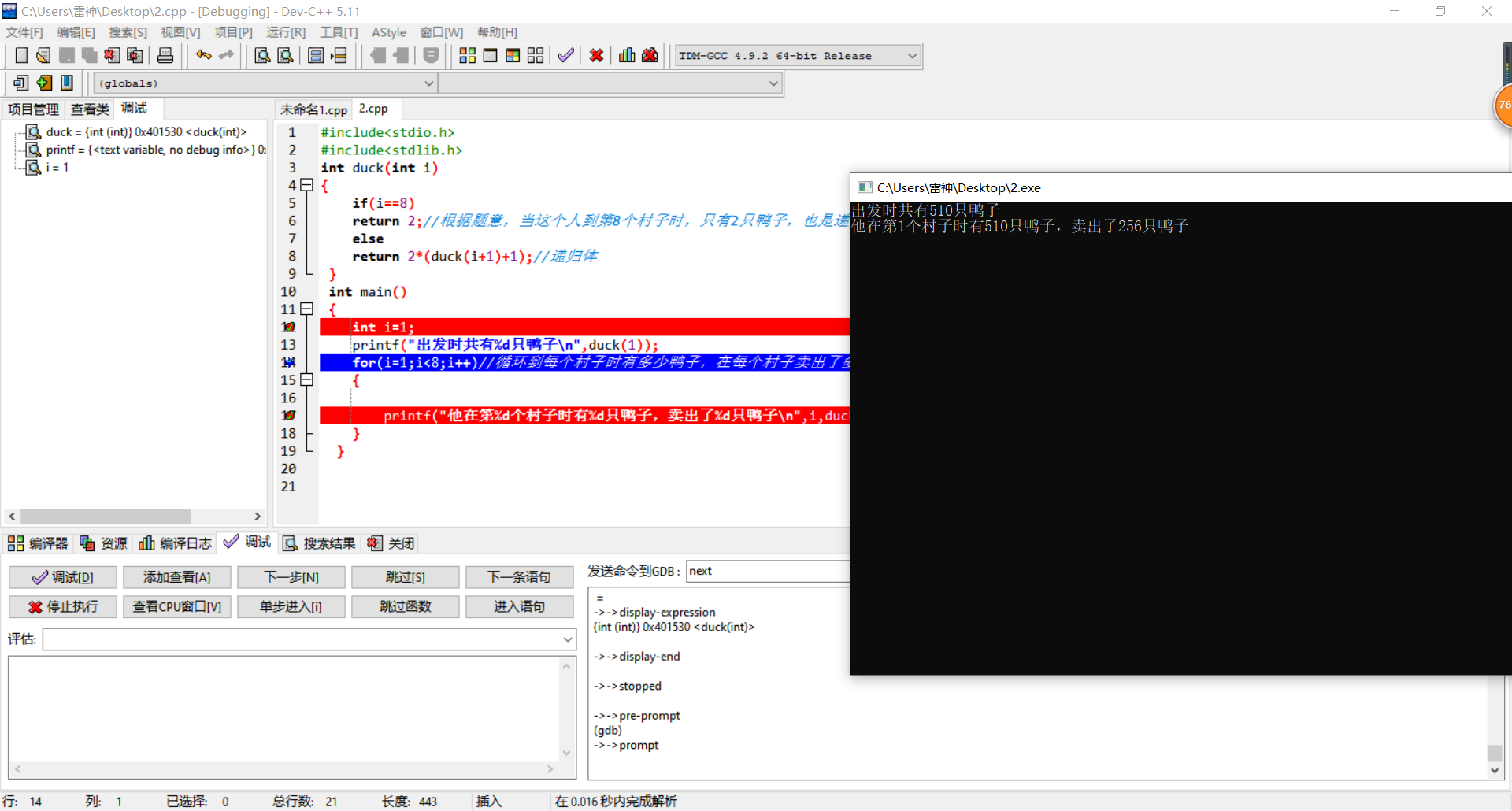
1. 计算出鸭子的总数



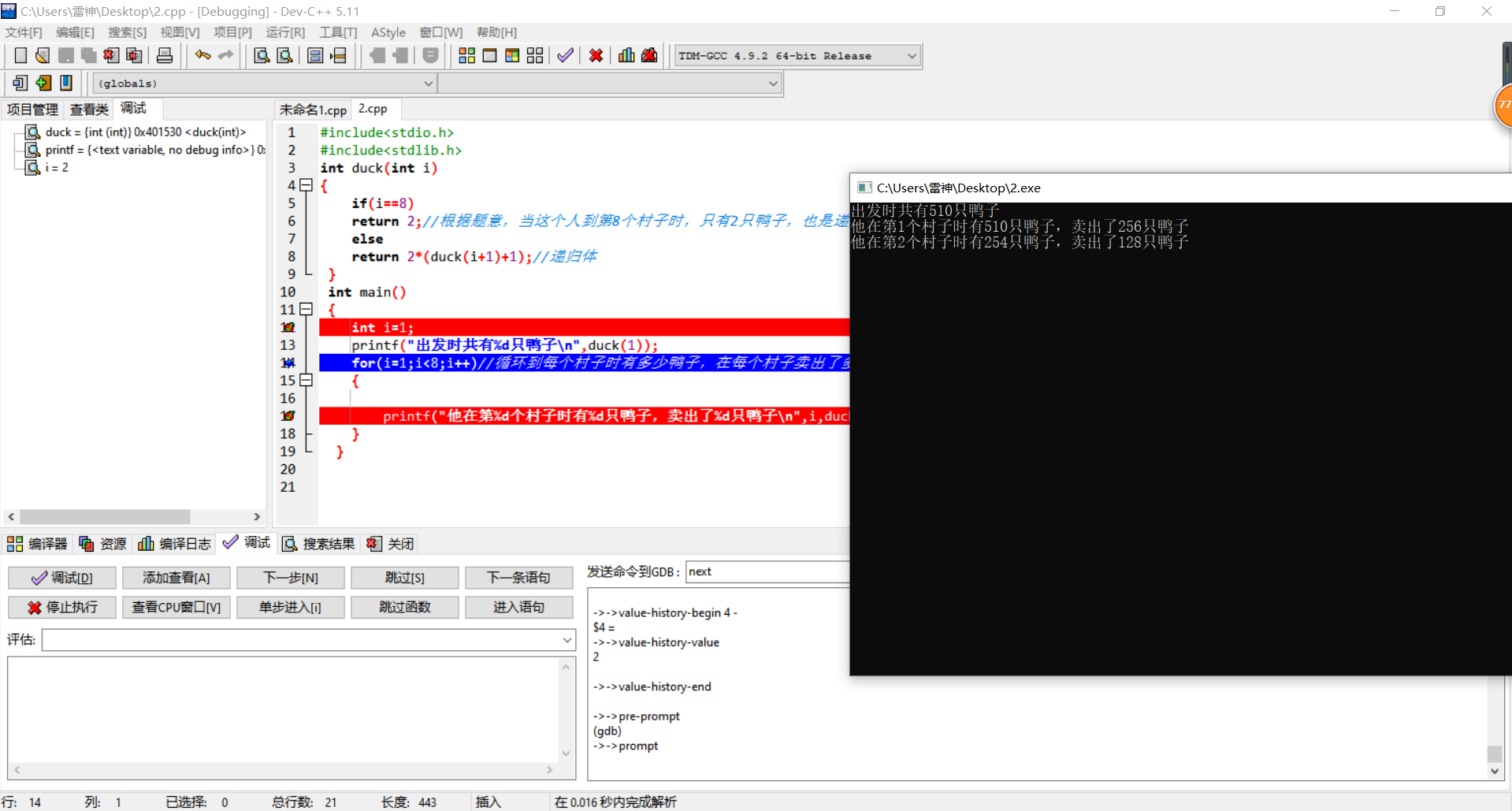
1. 在第一个村庄时
2. 进入函数体语句



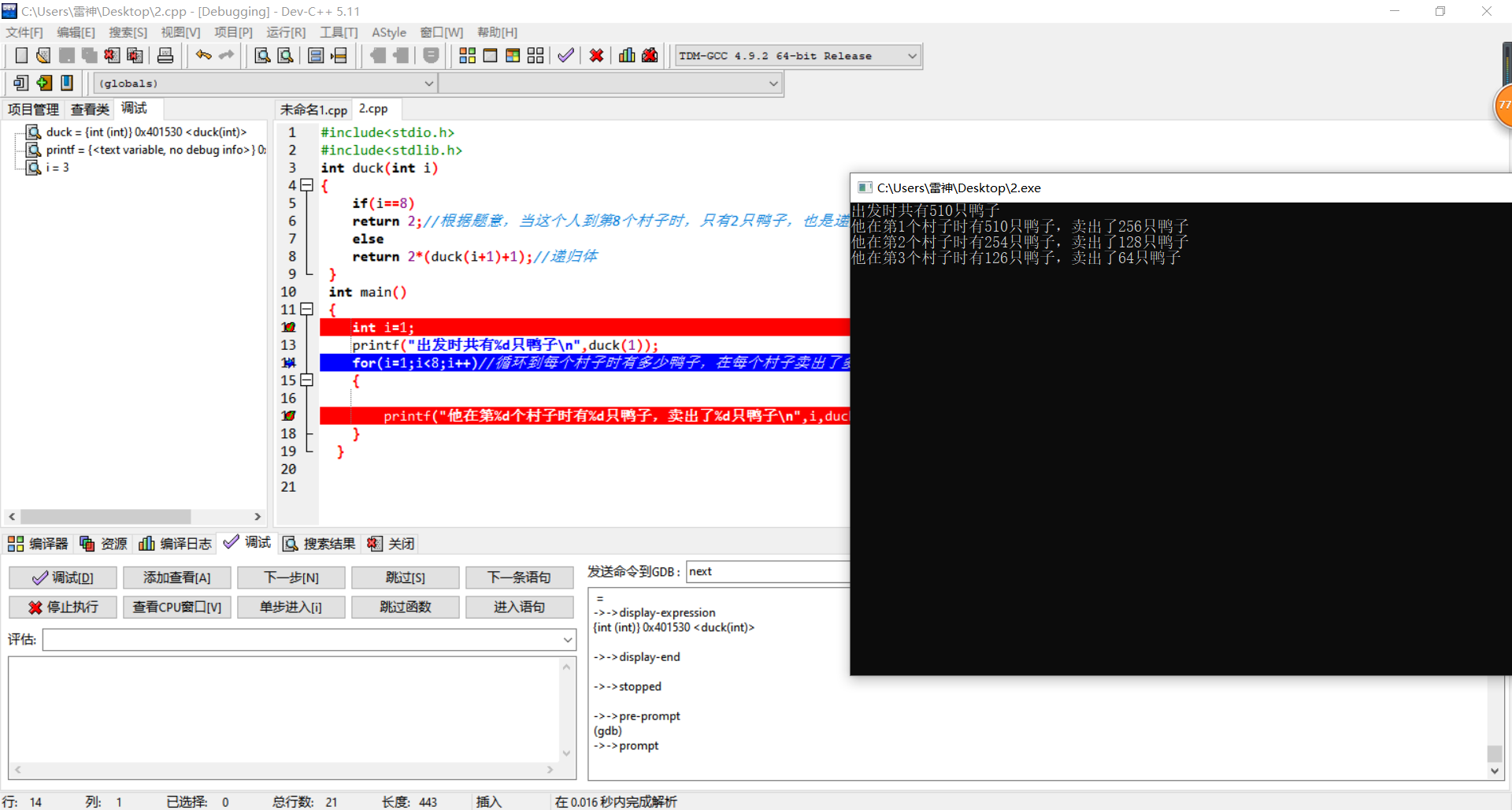
1. 运行完递归体后再次进入循环，并打印输出第一个村庄信息



1. 第二个村庄信息的打印输出
2. 第二个村庄信息的打印输出

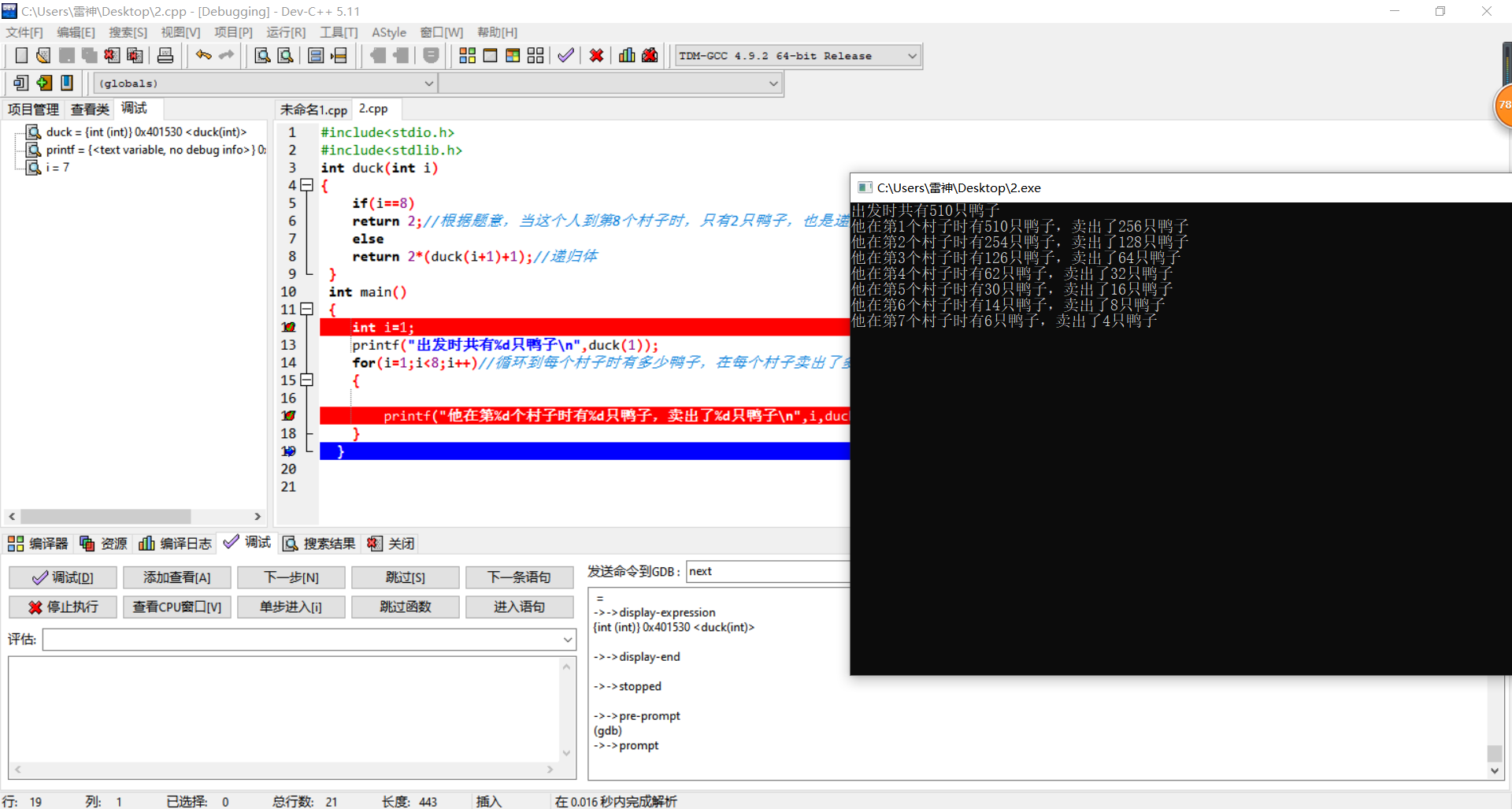


1. 第三个村庄信息的打印输出



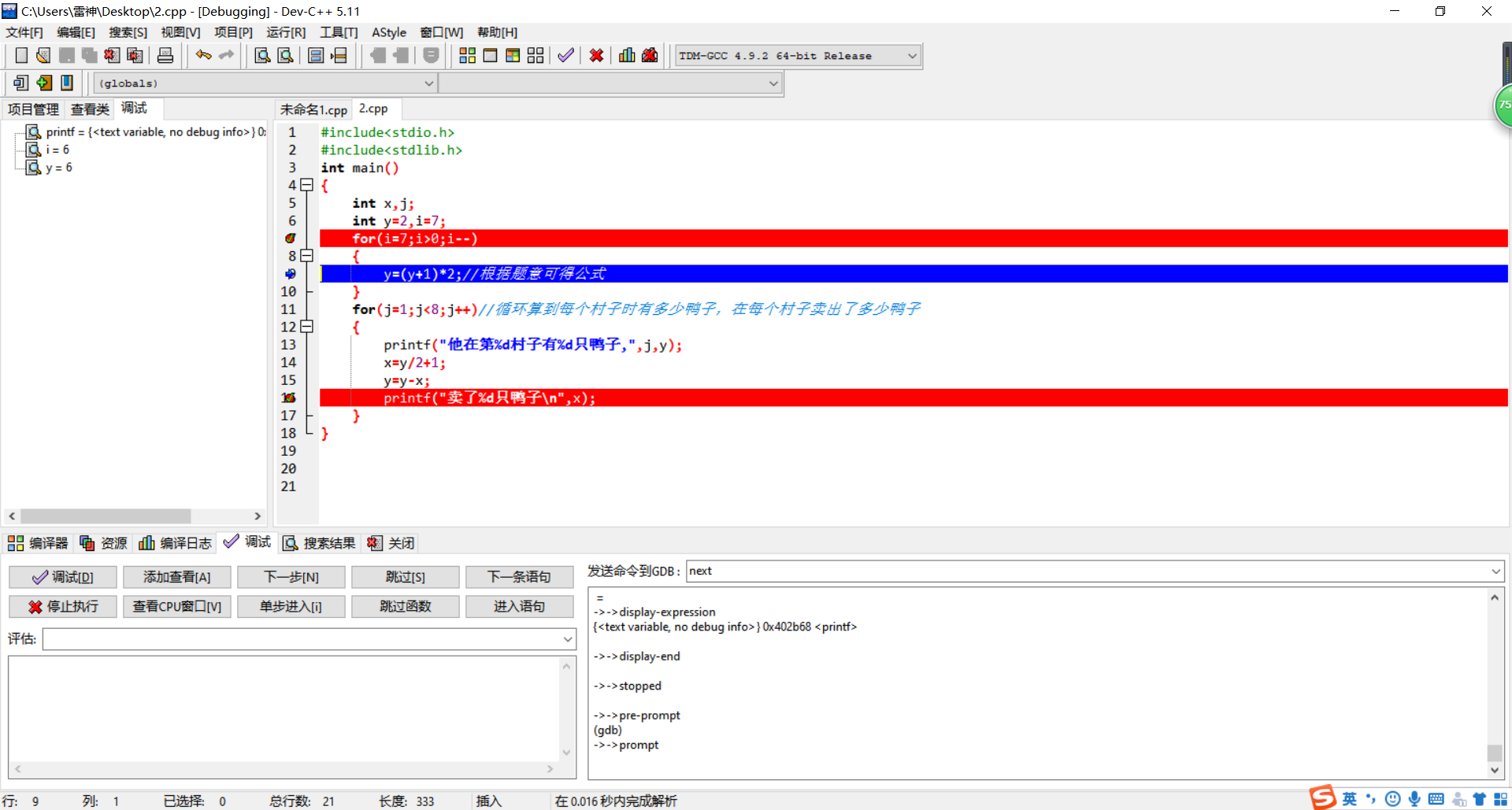
.........

1. 所有的信息打印输出



4.1.3非递归调试截图

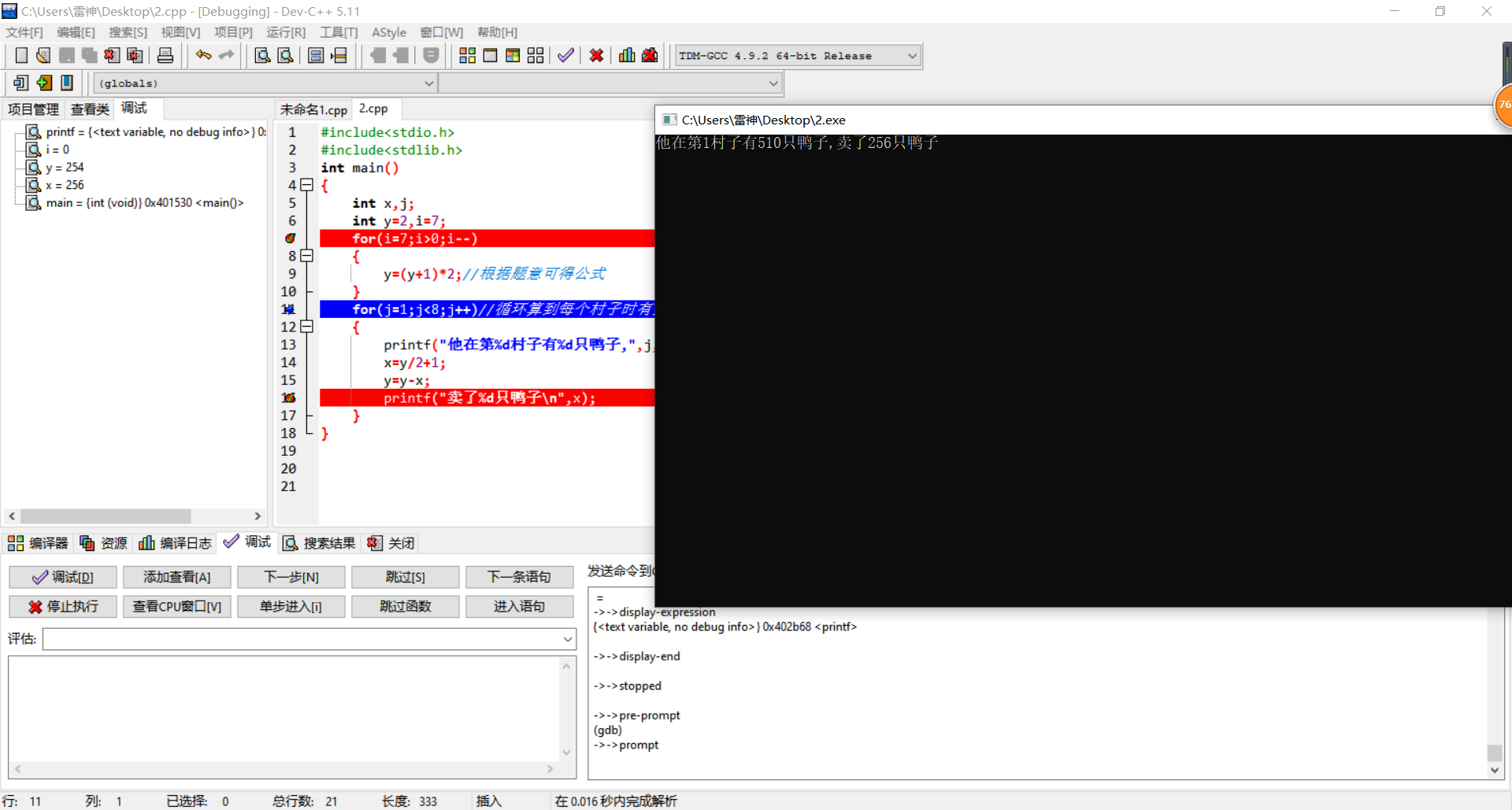
1. 进行计算每一个村庄的鸭子数目



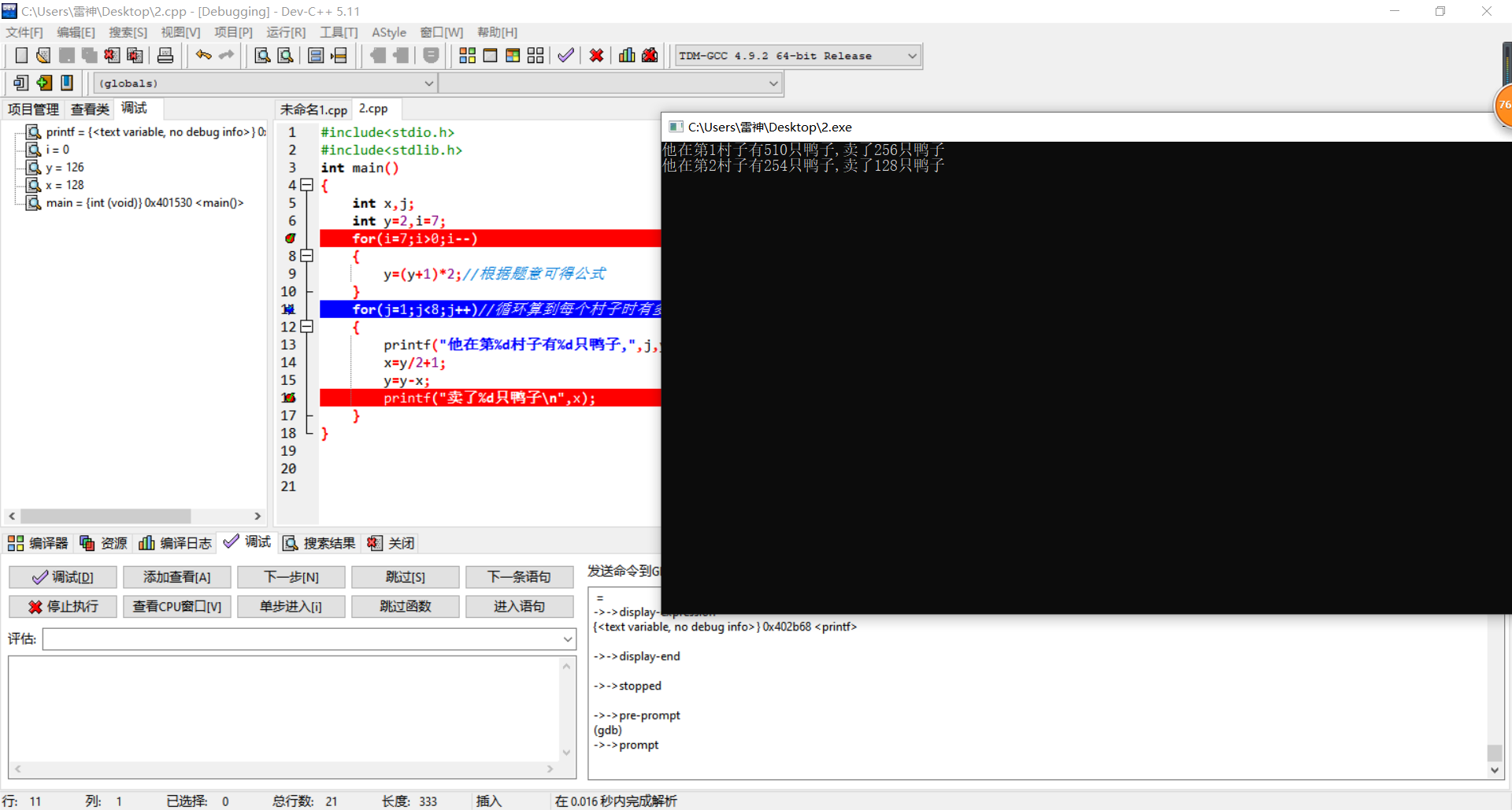
调试结果记录：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 村庄数i | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 卖出的鸭子数x | 254 | 126 | 62 | 30 | 14 | 6 | 2 |
| 鸭子数y | 510 | 254 | 126 | 62 | 30 | 14 | 6 |

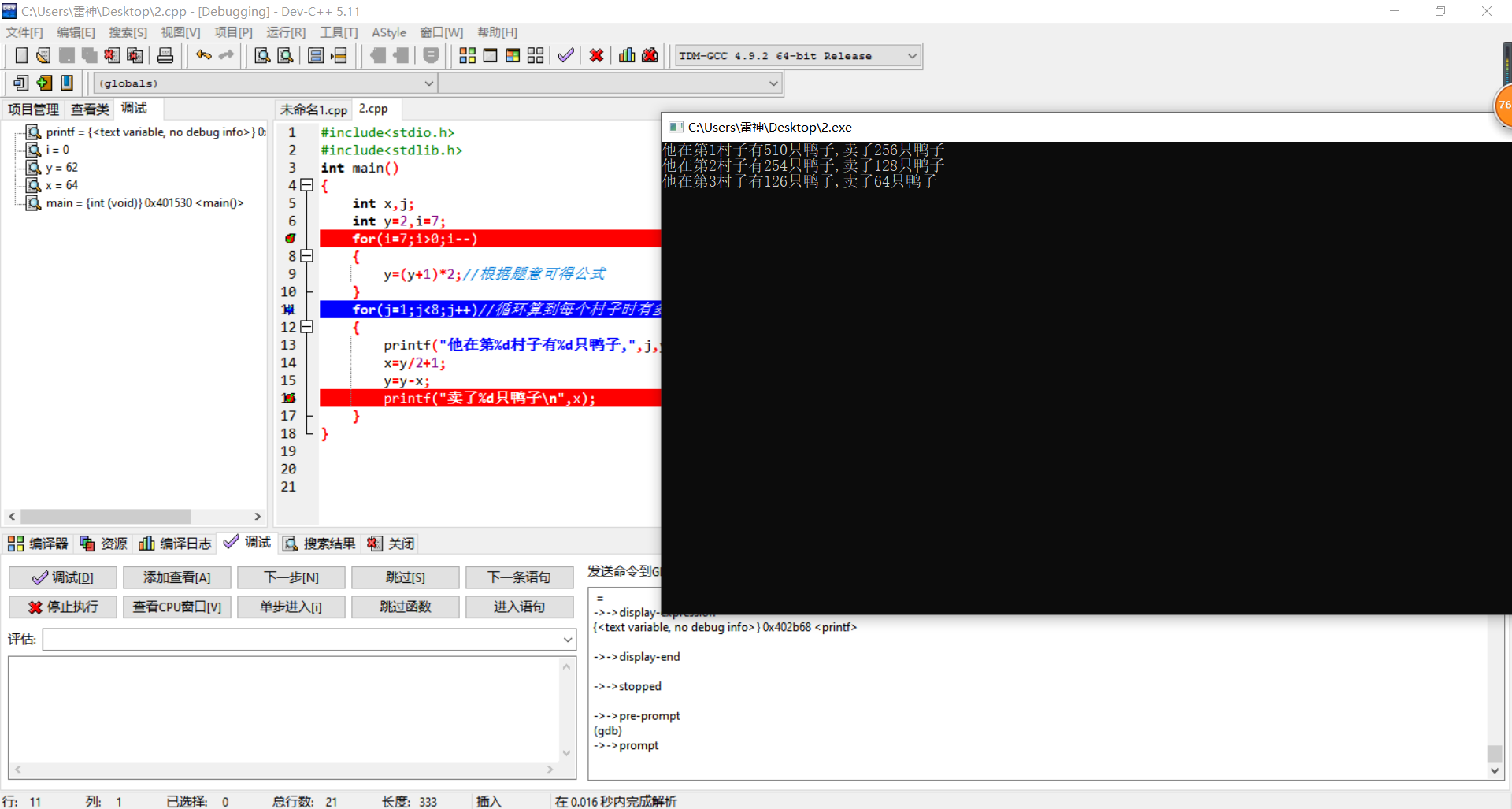
1. 在第一个村庄的信息打印输出



1. 在第二个村庄的信息打印输出

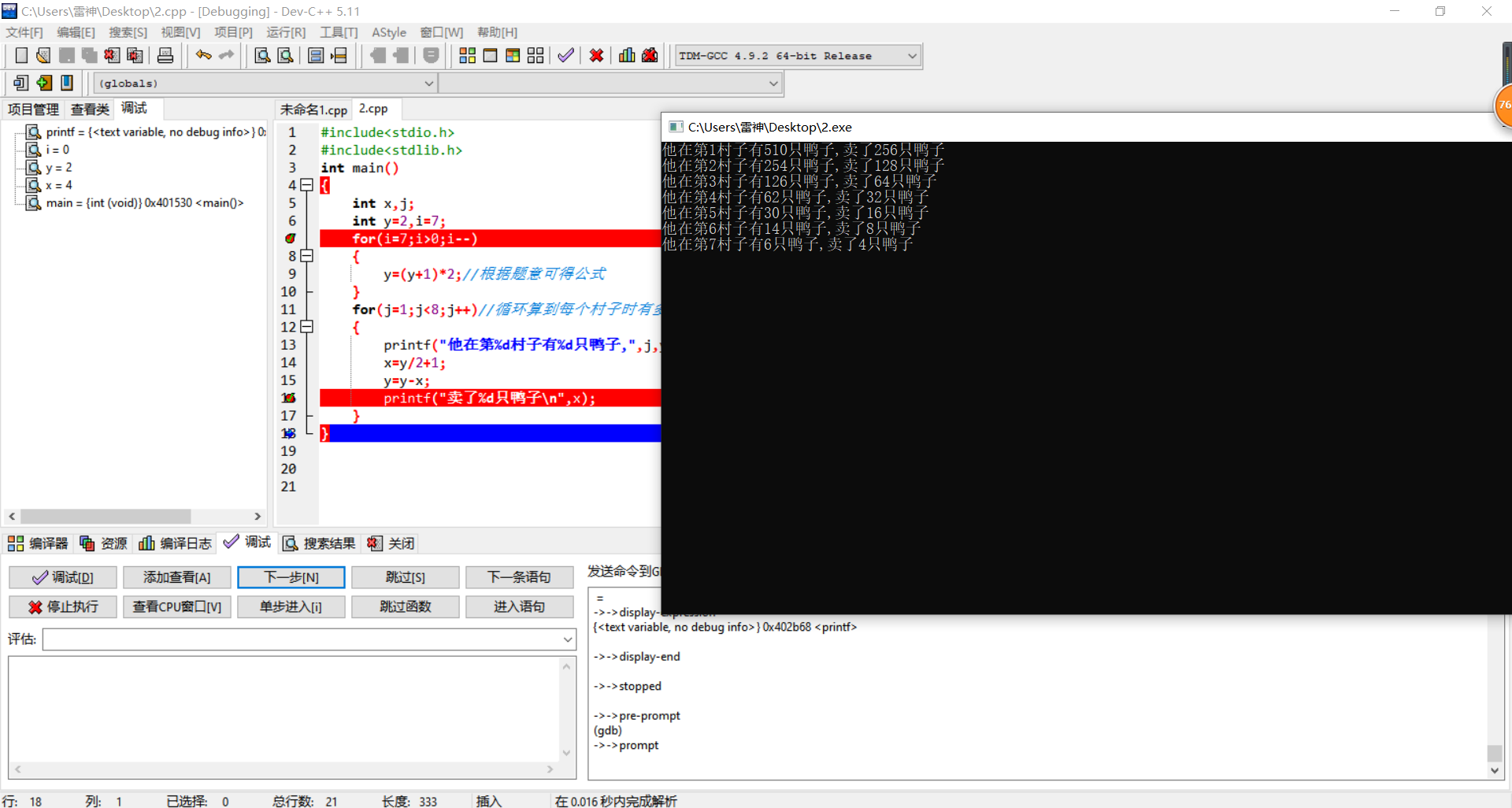


1. 在第三个村庄的信息打印输出



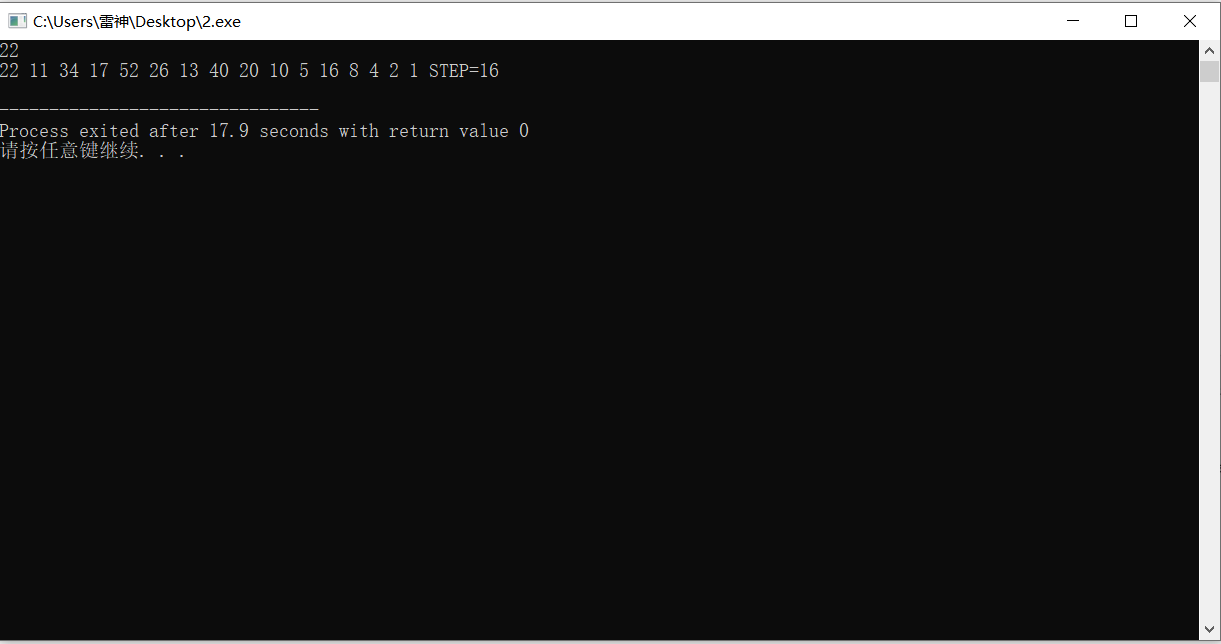
..........

1. 所有的信息打印输出

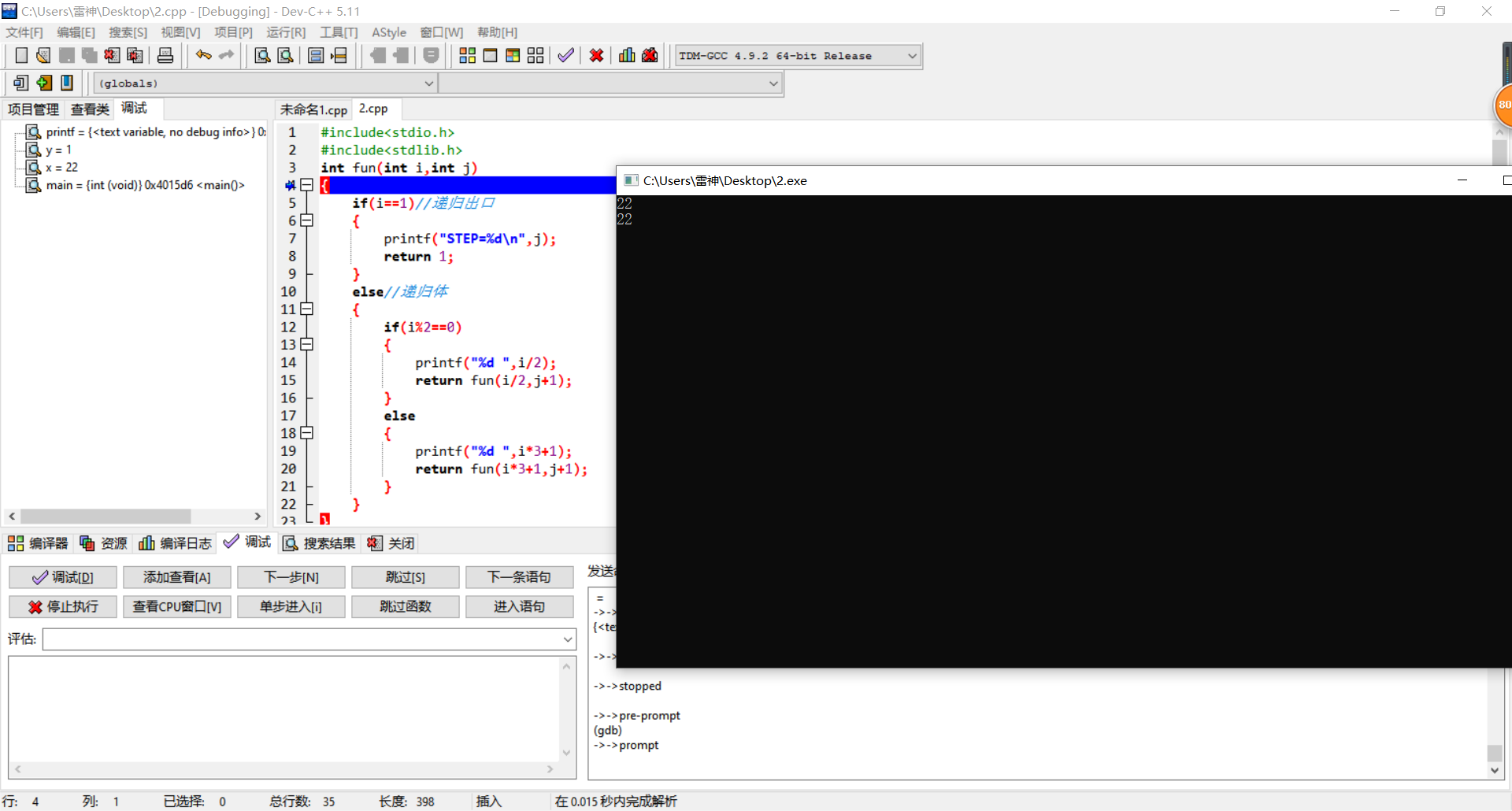


4.2角谷定理

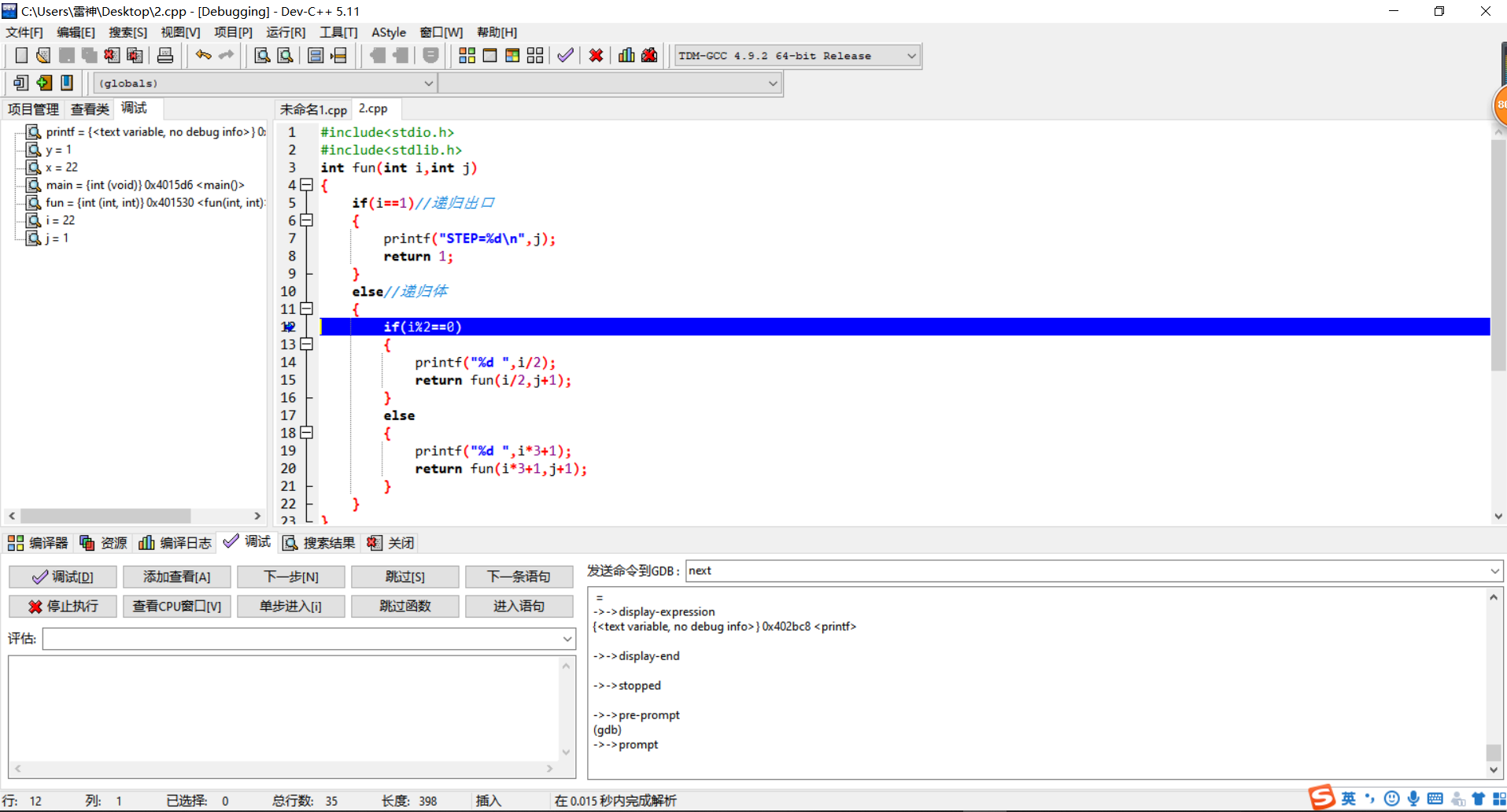
4.2.1运行结果

4.2.2递归调试截图

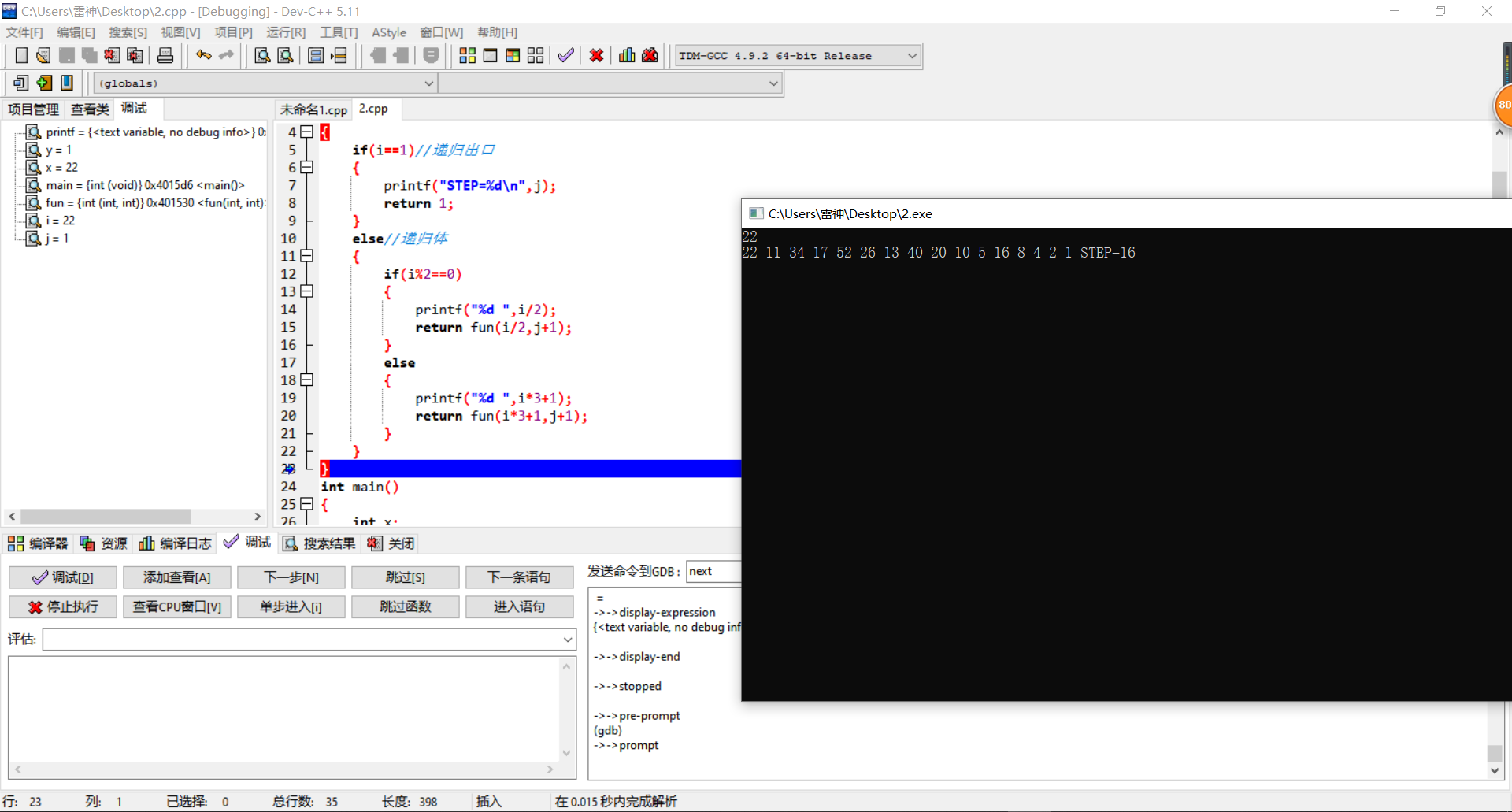
1. 进入函数语句



1. 进入递归体

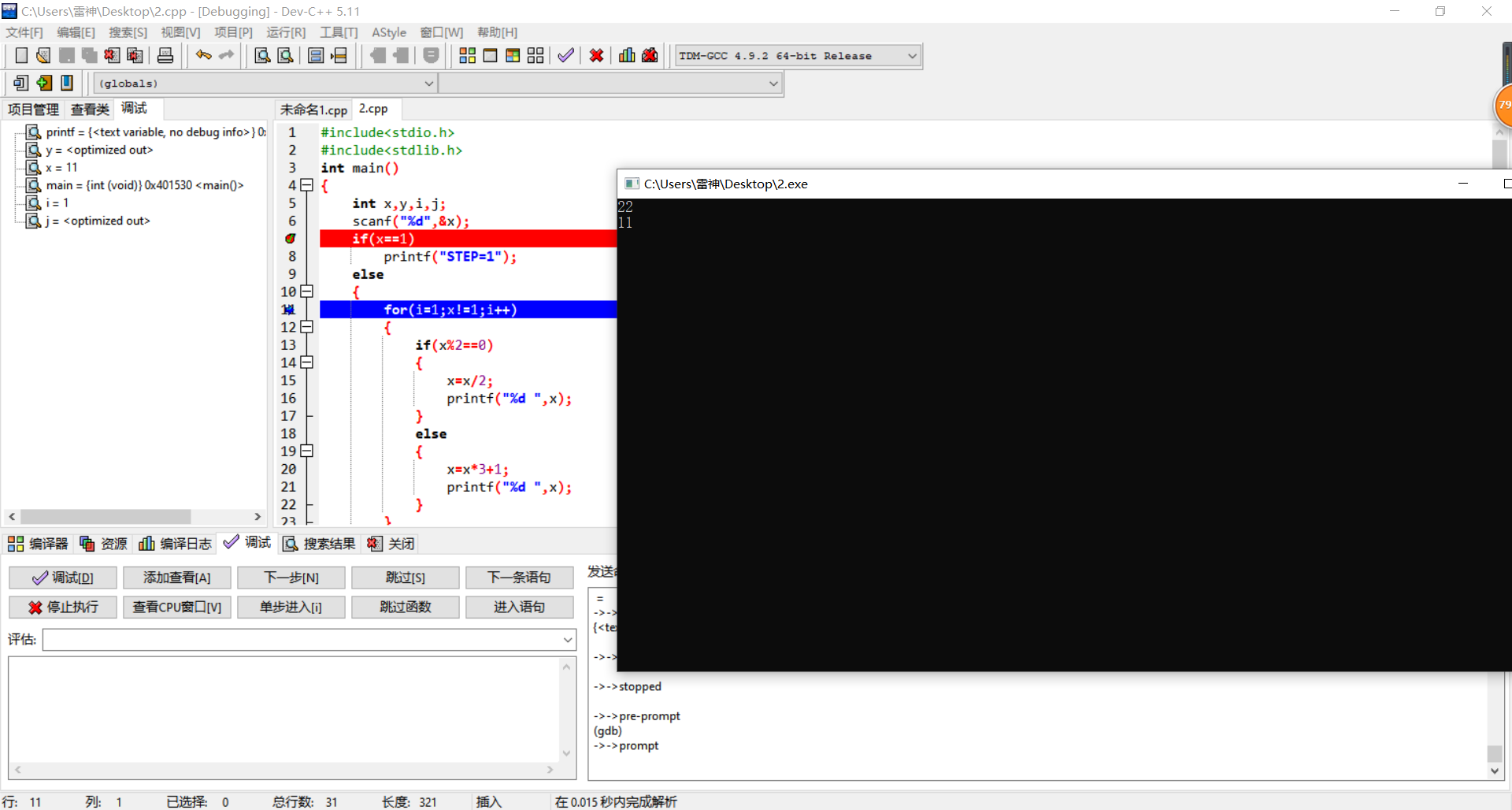


1. 跳出递归体

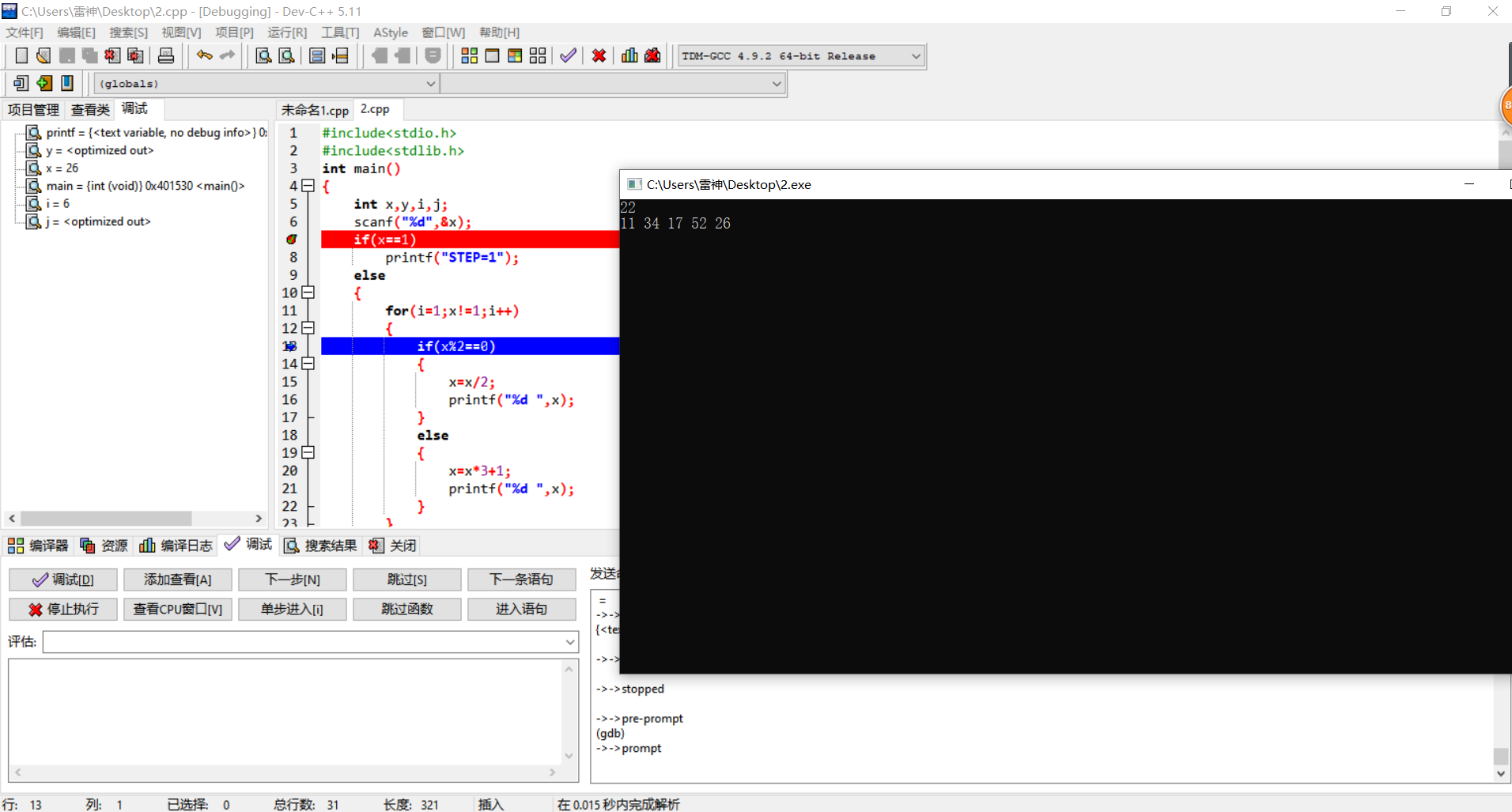


4.2.2非递归调试截图

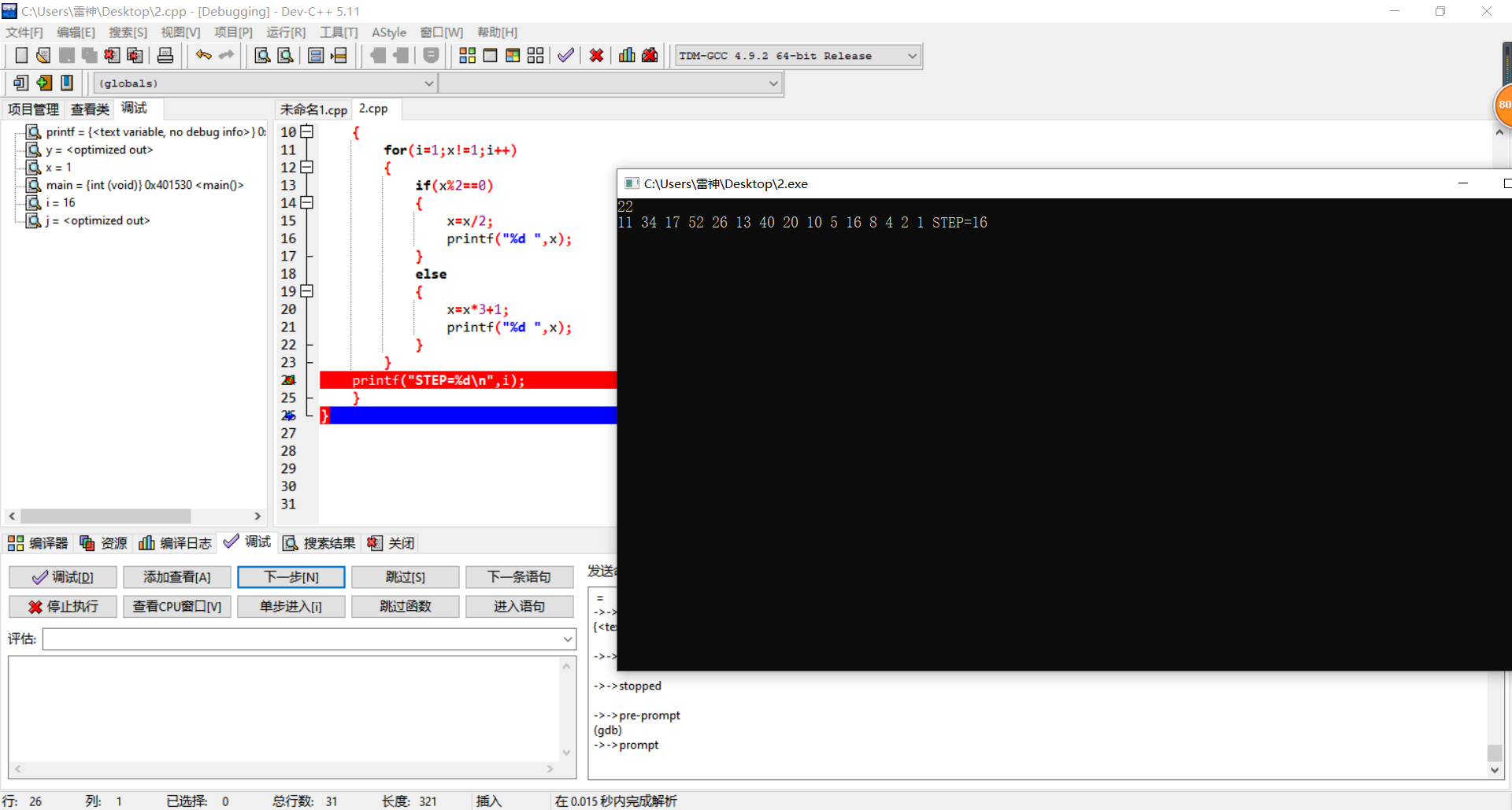
1. 进入for循环，判断是否为偶数，打印输出第一个数据



1. 进行判断输出



1. 输出步数，最终信息



1. 心得体会

在写递归方法遇到的问题主要是递归体的构造与变量的选定，有可能会进入死循环。但使用递归方法可以减少代码量，同时也更加灵活。主要是要找到递归出口与递归体，结构一般为if-else语句，递归体一般由一个或多个推导数学公式组成。