

**课 程 实 验 报 告**

**课程名称： C语言程序设计实验**

**专业班级：**

**学 号：**

**姓 名：**

**指导教师：**

**报告日期： 2019.10.19**

**网络空间安全学院**

**目 录**

**[1 表达式和标准输入输出实验 1](#_Toc404837920)**

[1.1 实验目的 1](#_Toc404837921)

[1.2 实验内容 1](#_Toc404837922)

[1.3 实验小结 1](#_Toc404837923)

# 1 表达式和标准输入与输出实验

## 1.1 实验目的

(1)熟练掌握各种运算符的运算功能，操作数的类型，运算结果的类型及运算过程中的类型转换，重点是C语言特有的运算符，例如位运算符，问号运算符，逗号运算符等；熟记运算符的优先级和结合性。

（2）掌握getchar, putchar, scanf 和printf 函数的用法。

（3）掌握简单C程序（顺序结构程序）的编写方法。

## 1.2 实验内容

**1.2.1 源程序改错**

下面给出了一个简单C语言程序例程，用来完成以下工作：

（1）输入华氏温度f，将它转换成摄氏温度c后输出；

（2）输入圆的半径值ｒ，计算并输出圆的面积ｓ；

（3）输入短整数ｋ、ｐ，将ｋ的高字节作为结果的低字节，ｐ的高字节作为结果的高字节，拼成一个新的整数，然后输出；

在这个例子程序中存在若干语法和逻辑错误。要求参照1.3和1.4的步骤对下面程序进行调试修改，使之能够正确完成指定任务。

1. #include<stdio.h>

2 #define PI 3.14159;

3 voidmain( void )

4 {

5 int f ;

6 short p, k ;

7 double c , r , s ;

8 /\* for task 1 \*/

9 printf(“Input Fahrenheit:” ) ;

10 scanf(“%d”, f ) ;

11 c = 5/9\*(f-32) ;

12 printf( “ \n %d (F) = %.2f (C)\n\n ”, f, c ) ;

13 /\* for task 2 \*/

14 printf("input the radius r:");

15 scanf("%f", &r);

16 s = PI \* r \* r;

17 printf("\nThe acreage is %.2f\n\n",&s);

18 /\* for task 3 \*/

19 printf("input hex int k, p :");

20 scanf("%x %x", &k, &p );

21 newint = (p&0xff00)|(k&0xff00)<<8;

22 printf("new int = %x\n\n",newint);

}

**解答：** （1）错误修改：

1) 第2行的符号常量定义后不能有分号，正确形式为：

#define PI 3.14159

2) 第3行的voidmain( void )，正确形式为：

main( void )

3）第11行的scanf函数中，不能读取变量b的值，正确形式为：

scanf("%d",&f ) ;

4）第12行的表达式会使c成为一个整数，正确形式为：

c = 5.0/9\*(f-32) ;

5）第15行的scanf函数无法正确读入数据，正确形式为：

scanf("%lf", &r);

6）第17行的printf函数格式错误，正确形式为：

printf("\nThe acreage is %.2f\n\n",s);

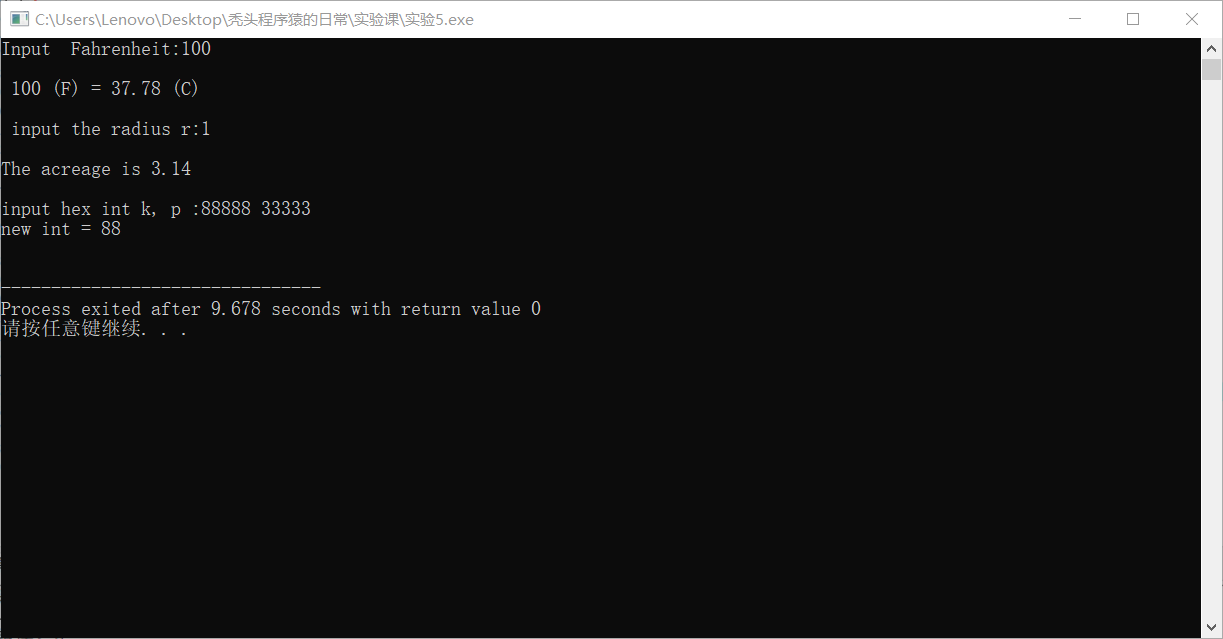
7）该函数未定义变量newint，故在21行前应加入：

short newint;

8）21行表达式中逻辑尺错误，正确形式为：

newint = (p&0xff00)|((k&0xff00)>>8);

（2）错误修改后运行结果：

****

**图1-1 改错题的运行结果**

**1.2.2 源程序修改替换**

下面的程序利用常用的中间变量法实现两数交换，请改用不使用第3个变量的方法实现。该程序中t是中间变量，要求将定义语句中的t删除，修改下划线处的语句，使之实现两数对调的操作。

#include<stdio.h>

void main( )

{

int a, b, t;

printf(“Input two integers:”);

scanf(“%d %d”,&a,&b);

t=a ；a=b；b=t；

prinf(“\na=%d,b=%d”,a,b);

}

**解答：**

可以巧妙地对a，b赋值而满足目标要求，替换后的程序如下所示：

#include<stdio.h>

void main( )

{

int a, b;

printf(“Input two integers:”);

scanf(“%d %d”,&a,&b);

a=a+b；b=a-b；a=a-b；

prinf(“\na=%d,b=%d”,a,b);

}

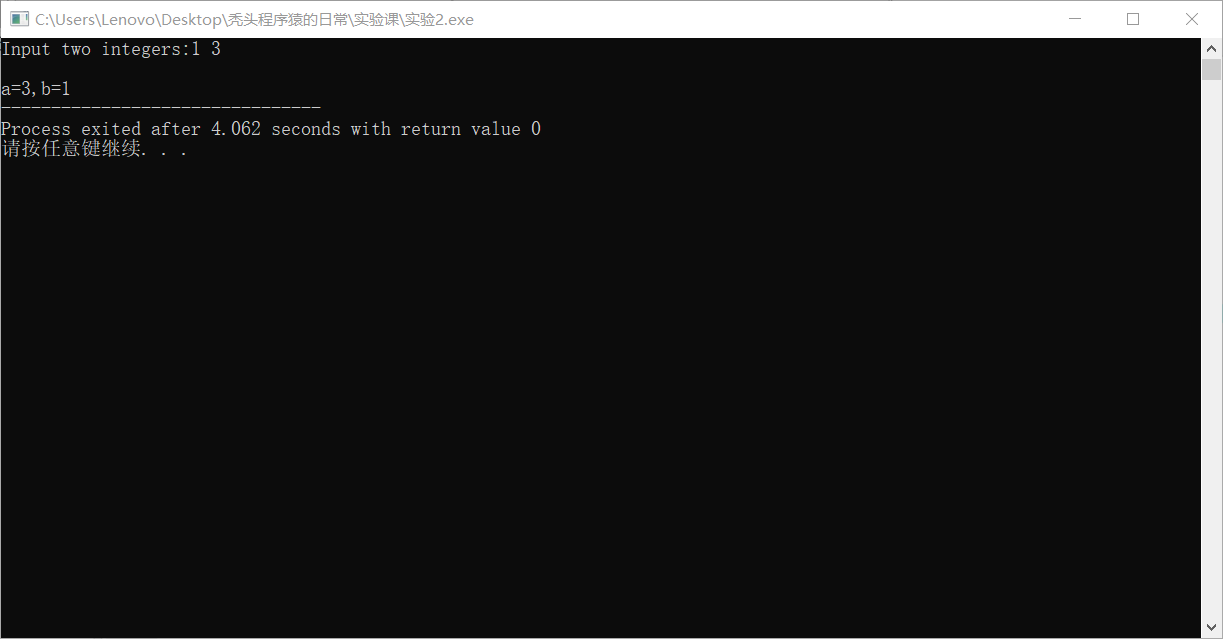


图1-2 程序修改替换题运行结果

**1.2.3 程序设计**

**（1）**编写一个程序，输入字符ｃ，如果ｃ是大写字母，则将ｃ转换成对应的小写，否则ｃ的值不变，最后输出ｃ。

**解答：**

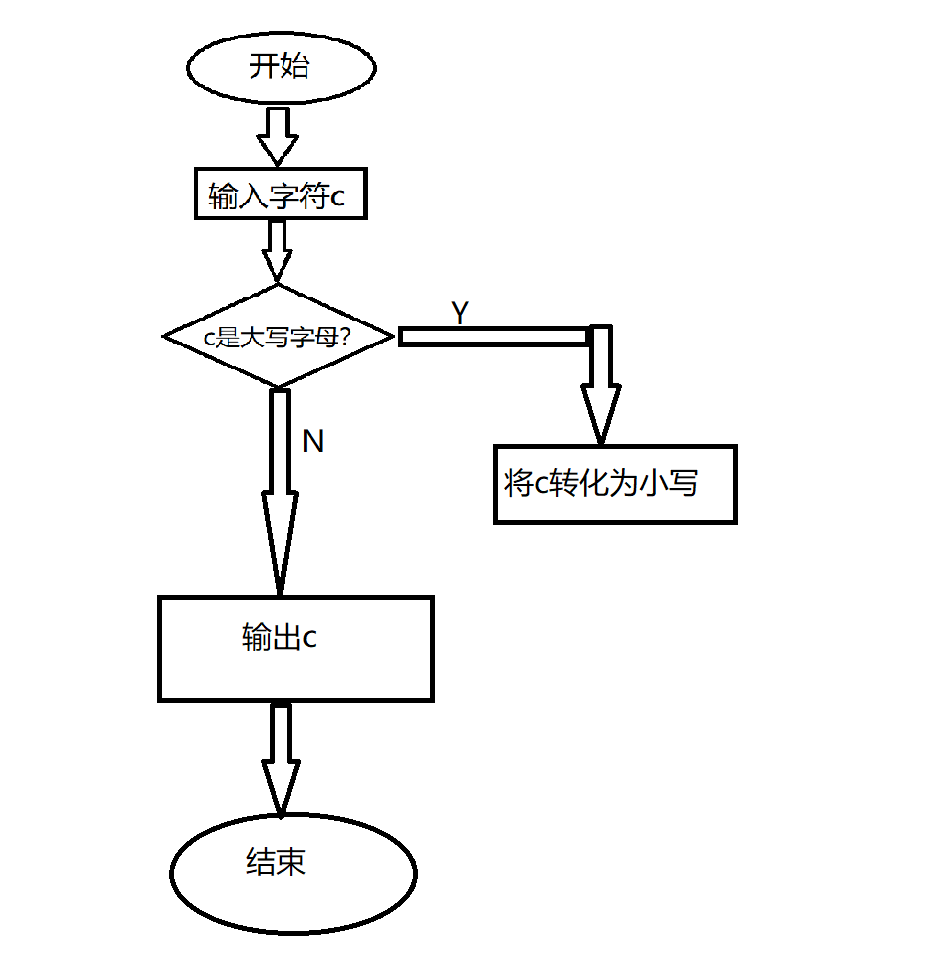
1） 算法流程如图1.1所示。

图1-3 编程题1的程序流程图

2）源程序清单

#include<stdio.h>

int main(void)

{

char f;

printf("Input A Word:");

scanf("%c",&f);

if(f>=65&&f<=90)

printf("%c\n",f+32);

else

printf("%c\n",f);

return 0;

}

3）测试

（a） 测试数据：

B z

（b） 对应测试数据的运行结果截图

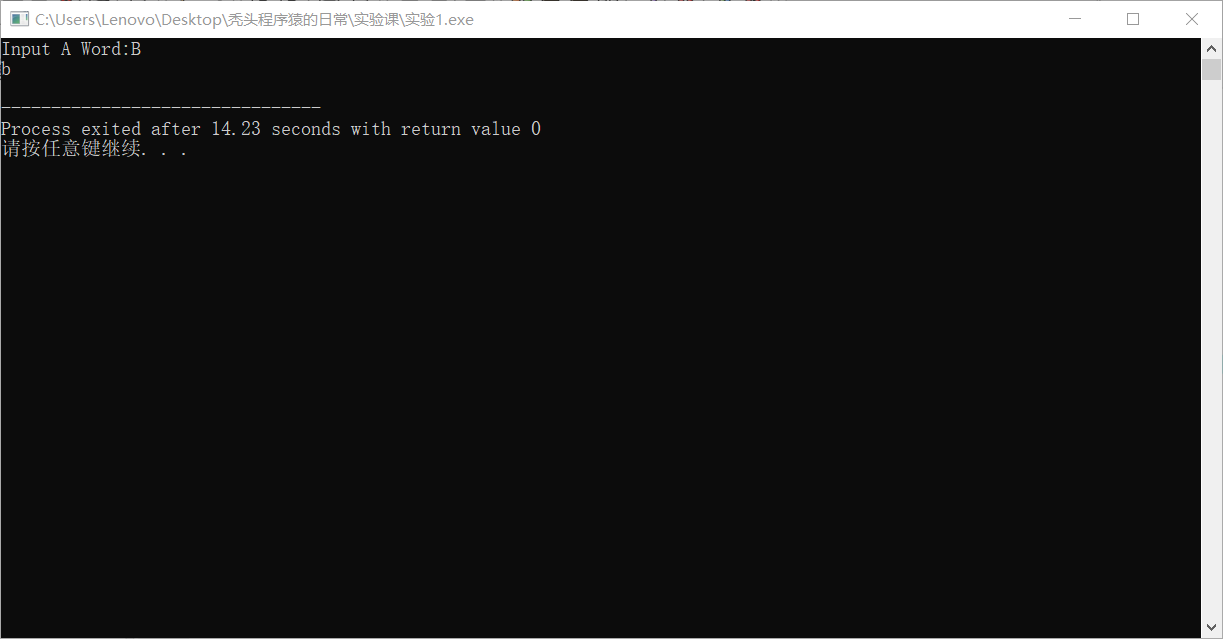


图1-4 编程题1的运行结果实例1

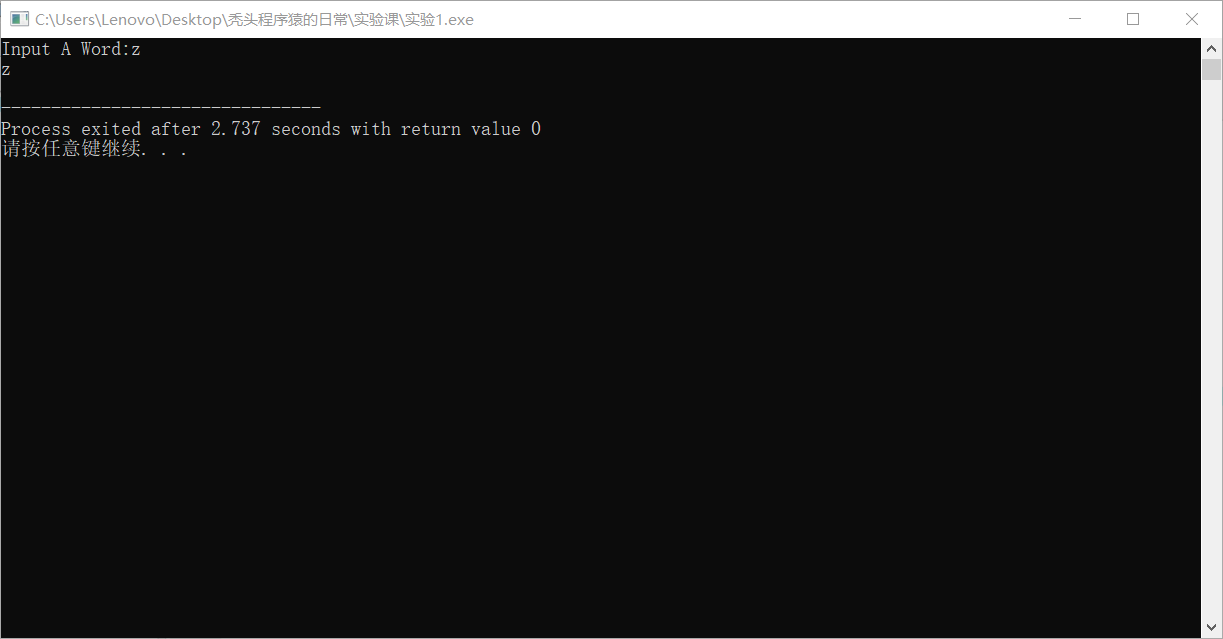
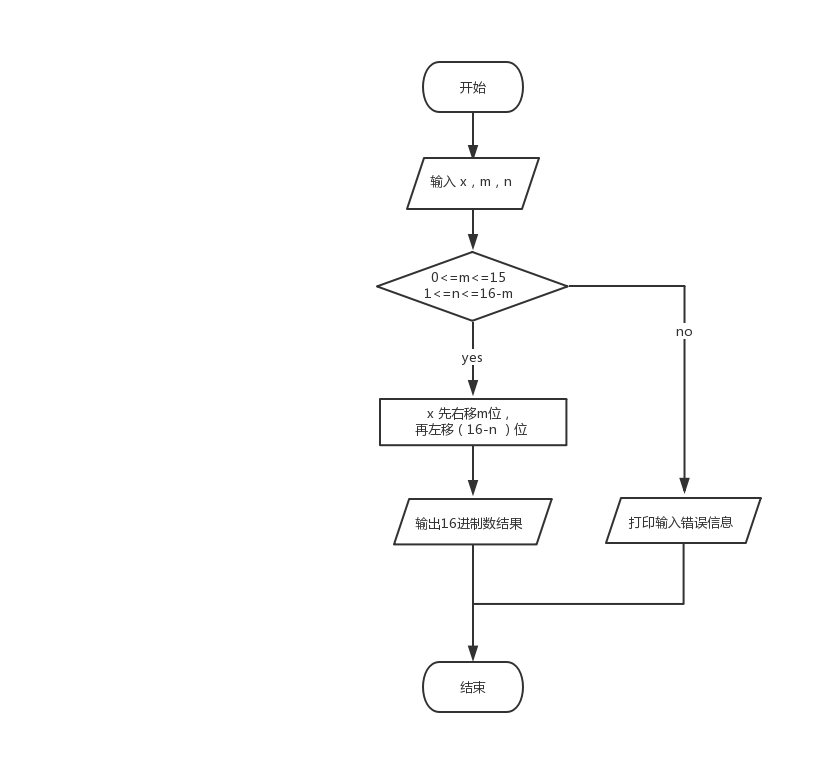


图1-5 编程题1的运行结果实例2

（2）编写一个程序，输入无符号短整数x，ｍ，ｎ（0 ≤ｍ≤ 15, 1 ≤ ｎ≤ 16-ｍ）,取出x从第ｍ位开始向左的ｎ位（ｍ从右至左编号为0～15），并使其向左端（第15位）靠齐。

**解答：**

1. 解题思路：



2）程序清单：

#include <stdio.h>

void main( void )

{

unsigned short x,m,n;

printf("输入x（16进制）、m（0~15）和n（1~16-m）: \n ");

scanf("%hx %hd %hd",&x,&m,&n);

if ((m>=0&&m<=15)&&(n>=1&n<=16-m) )

/\*判断m、n的值是否在合理范围内\*/

{

x = x>>m <<(16-n); /\*x右移m位，左移16-n位\*/

printf("ans=%hx\n",x);

}

else

{

printf("输入错误!\n");

}

}

3）测试

（a） 测试数据：

测试数据的方法如表1-1所示

表1-1 编程题3的测试数据

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试  用例 | 程 序 输 入 | | | 理 论 结 果 | 运 行 结 果 |
| x | m | n |
| 用例1 | 0100 0110 1000 0000（4680） | 7 | 4 | 计算结果  1101 0000 0000 0000 即D000 | D000 |
| 用例2 | 1101 0101 1000 0011（D583） | 16 | 1 | 输入错误  （m值超范围） | 输入错误！ |
| 用例3 | 1101 0101 1000 0011（D583） | 13 | 5 | 输入错误  （n值超范围） | 输入错误！ |

（b）运行结果：

对应测试测试用例1的运行结果如图所示。

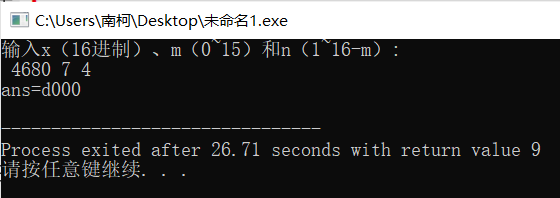


图1-6 编程题2的运行结果实例1

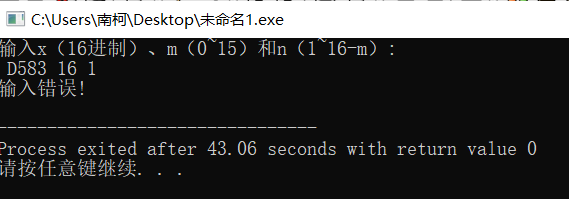
对应测试测试用例2的运行结果如图1-3所示。

图1-7 编程题2的运行结果实例2

对应测试测试用例3的运行结果如图1-4所示。

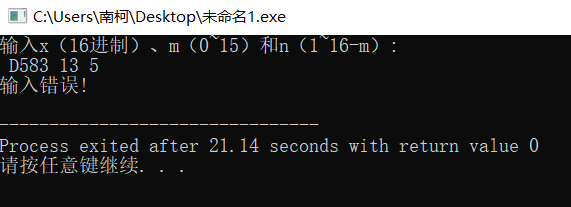


图1-8 编程题2的运行结果实例3

说明上述的运行结果与理论分析吻合，验证了程序的正确性。

**（3）**IP地址通常是4个用句点分隔的小整数，如32.55.1.102。这些地址在机器中用无符号长整形表示。编写一个程序，以机器存储的形式读入一个32位的互联网IP地址，对其译码，然后用常见的句点分隔的4部分的形式输出。

**解答：**

1. 算法流程如图所示。

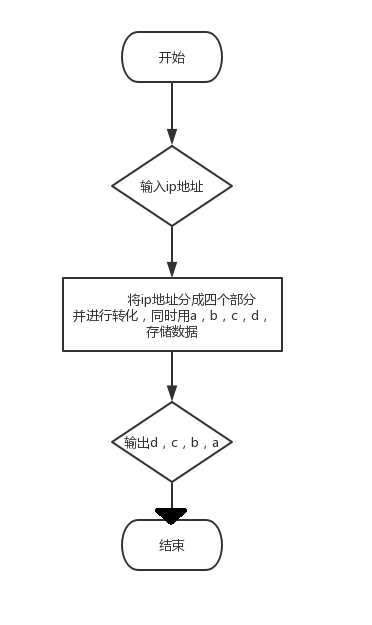


图1-9 编程题3的程序流程图

1. 源程序清单

#include <stdio.h>

int main()

{

unsigned long x,a,b,c,d;

printf("Input x:\n");

scanf("%ul",&x);

a=(x&0xff000000)>>24;

b=(x&0x00ff0000)>>16;

c=(x&0x0000ff00)>>8;

d=(x&0x000000ff);

printf("%d.%d.%d.%d",d,c,b,a);

return 0;

}

1. 测试
2. 测试数据：

用例1：56445645 理论结果：205.74.93.3

用例2：8965468 理论结果：92.205.136.0

用例3：456641 理论结果：193.247.6.0

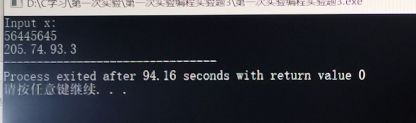
1. 对应测试测试用例1的运行结果如图所示。

图1-10 编程题3的运行结果实例1

对应测试测试用例2的运行结果如图所示。

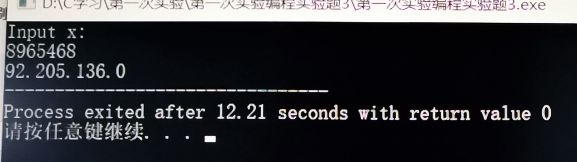


图1-11 编程题3的运行结果实例2

对应测试测试用例3的运行结果如图所示。

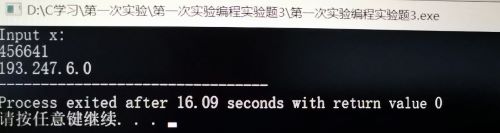


图1-12 编程题3的运行结果实例3

说明上述的运行结果与理论分析吻合，验证了程序的正确性。

## 1.3 实验小结

主要叙述实验过程中遇到的问题，如何解决的，通过分析、结果问题后的体会。  
 在这次实验的过程中，我在进行第三个编程实验时曾经遇到过不会正确调整逻辑尺对应位数的问题。最终我通过上网和查阅相关书籍解决了这一问题。

通过本次实验课，我深刻地体会到了“纸上得来终觉浅，绝知此事要躬行”这句话所蕴含的深刻道理。也许课上听完讲后我觉得自己已经掌握了相关的知识，但直到自己亲自动手编程是才发现自己的能力并没有想象中那么出色。希望我以后可以在这种活动中收获更多！

参考文献

[1] 曹计昌,卢萍,李开. C语言程序设计,北京： 科学出版社,2013

[2] 李开,卢萍,曹计昌. C语言实验与课程设计, 北京：科学出版社,2011