

**课 程 实 验 报 告**

**课程名称： C语言程序设计实验**

**专业班级： 本硕博202001班**

**学 号： U202112205**

**姓 名： 刘浩阳**

**指导教师： 吴俊军**

**报告日期： 2021年10月30日**

**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)

[**2 流程控制实验 2**](#_Toc404837924)

[2.1 实验目的 1](#_Toc404837921)

[2.2 实验内容 1](#_Toc404837922)

[2.3 实验小结 1](#_Toc404837923)

[**3 函数与程序结构实验 3**](#_Toc404837929)

[3.1 实验目的 1](#_Toc404837921)

[3.2 实验内容 1](#_Toc404837922)

[3.3 实验小结 1](#_Toc404837923)

[**4 编译预处理实验 4**](#_Toc404837934)

[4.1 实验目的 1](#_Toc404837921)

[4.2 实验内容 1](#_Toc404837922)

[4.3 实验小结 1](#_Toc404837923)

[**5 数组实验 5**](#_Toc404837938)

[5.1 实验目的 1](#_Toc404837921)

[5.2 实验内容 1](#_Toc404837922)

[5.3 实验小结 1](#_Toc404837923)

[**6 指针实验 6**](#_Toc404837943)

[6.1 实验目的 1](#_Toc404837921)

[6.2 实验内容 1](#_Toc404837922)

[6.3 实验小结 1](#_Toc404837923)

[**7 结构与联合实验 7**](#_Toc404837948)

[7.1 实验目的 1](#_Toc404837921)

[7.2 实验内容 1](#_Toc404837922)

[7.3 实验小结 1](#_Toc404837923)

[**8 文件实验 8**](#_Toc404837953)

[8.1 实验目的 1](#_Toc404837921)

[8.2 实验内容 1](#_Toc404837922)

[8.3 实验小结 1](#_Toc404837923)

[**参考文献 9**](#_Toc404837957)

# 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程序的编写方法。

（4）熟悉C语言程序的开发环境，并学会调试程序的方法。

## 1.2 实验内容

**1.2.1 程序改错与跟踪调试**

下面的实验1-1程序用来完成以下任务：

（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）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，拼成一个新的整数后输出。

在这个程序中存在若干语法和逻辑错误，要求先编译程序改正语法错误，再采用单步执行的方式调试程序找出逻辑错误。在单步执行程序的过程中，观察以下变量值：

（1）执行完c = 5/9 \* (f-32)，c的值为多少？

（2）执行完scanf(“%f”, &r)，r的值为多少？

（3）执行完newint = p&0xff00|k>>8，newint的值是多少？表达式k>>8的值是多少？

根据观察结果分析代码并修改程序，使之能够正确完成指定任务。

/\*实验1-1程序改错与跟踪调试题源程序\*/

1 #include<stdio.h>

2 #define PI 3.14159;

3 int main( void )

4 {

5 int f ;

6 short p, k ;

7 double c, r, s ;

/\* 任务1 \*/

8 printf("Input Fahrenheit: " ) ;

9 scanf("%d", f ) ;

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

11 printf( "\n %d (F) = %.2f (C)\n\n ", f, c ) ;

/\* 任务2 \*/

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

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

14 s = PI \* r \* r;

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

/\* 任务3 \*/

16 k = 0xa1b2, p = 0x8432;

17 newint = p&0xff00|k>>8;

18 printf("newint = %#x\n\n",newint);

19 return 0;

20 }

**解答：**

（1）语法错误修改：

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9行的scanf变量名前缺少&，正确形式为：scanf（“%d”，&f）；

3）第15行的printf变量名前多余&，正确形式为：

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

4）第17行的newint未定义，应添加：unsigned short newint;

（2）修改完以上错误后，在单步执行过程中，观察以下变量值：

1. 执行完c = 5/9 \* (f-32)，c的值为：0.00。
2. 执行完scanf(“%f”, &r)，r的值为：0.000000。
3. 执行完newint = p&0xff00|k>>8，newint的值是0xffa1。

表达式k>>8的值是0xffffffa1。

（3）基于以上的单步执行观察结果，说明还存在如下逻辑错误：

1）c = 5/9 \* (f-32)； ，由于整型溢出，应该修改成：c = 5.0/9 \* (f-32)； 。

2）scanf(“%f”, &r)； ，由于%f是对应float类型，而r是double型，故应该修改成：scanf(“%lf”, &r)； 。

3）newint = p&0xff00|k>>8; ,由于k的补位会补1，所以只保留末8位，应该修改成：newint = p&0xff00|(k>>8&0x00ff); 。

修改后，源程序清单如下：

1 #include<stdio.h>

2 #define PI 3.14159

3 int main( void )

4 {

5 int f ;

6 short p, k ;

7 double c, r, s ;

/\* 任务1 \*/

8 printf("Input Fahrenheit: " ) ;

9 scanf("%d",& f ) ;

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

11 printf( "\n %d (F) = %.2f (C)\n\n ", f, c ) ;

/\* 任务2 \*/

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

13 scanf("%lf", &r);

14 s = PI \* r \* r;

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

/\* 任务3 \*/

16 unsigned short newint;

17 k = 0xa1b2, p = 0x8432;

18 newint = p&0xff00|（k>>8&0x00ff）;

19 printf("newint = %#x\n\n",newint);

20 return 0;

21 }

（4）错误修改后运行结果如图1-1所示。

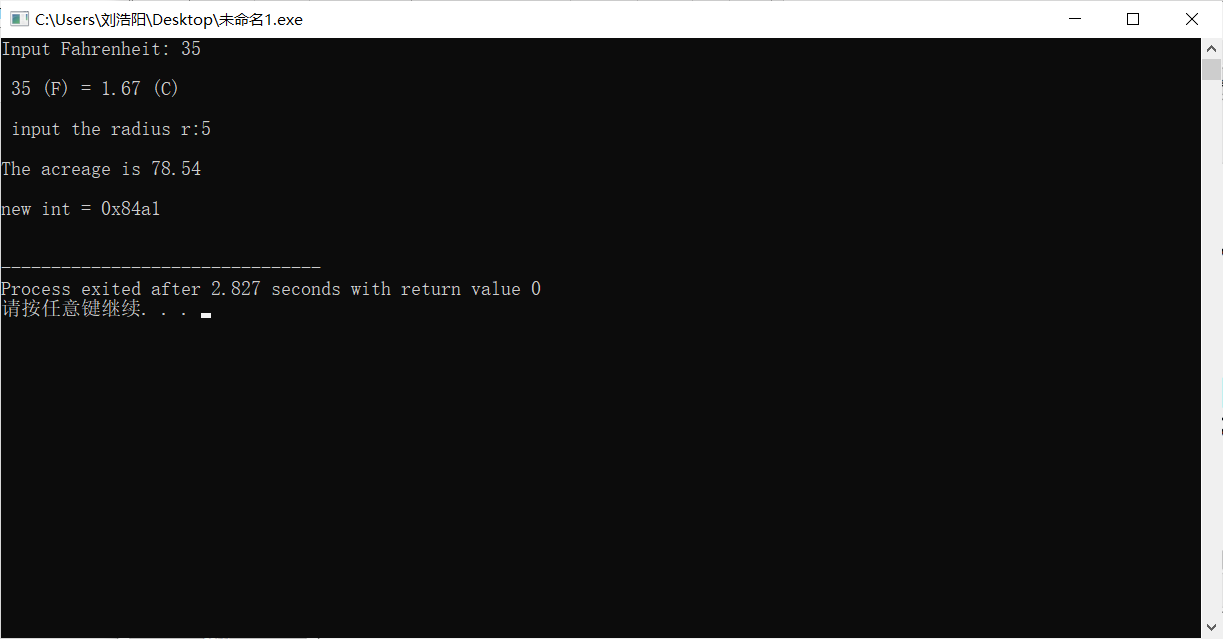


图1-1 实验1-1修改后运行结果截图

**1.2.2 程序设计**

（1）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，输入Ctrl+Z程序结束。要求：①用条件表达式；②字符的输入输出用getchar和putchar函数。程序应能循环接受用户的输入，直至输入Ctrl+Z程结束。例如：

A （键盘输入）

a

^Z （键盘输入）

**解答：**

1. 算法流程如图1-2所示。



图1-2 程序设计题1的程序流程图

2）源程序清单

#include<stdio.h>

int main()

{

char c;

while(c!=EOF)

{

c=getchar();

(c>='A'&&c<='Z')?c=c-'A'+'a':c;

if(c!=EOF)

putchar(c);

}

return 0;

}

3）测试

（a） 测试数据：如表1-1所示

表1-1 编程题1的测试数据

|  |  |  |  |
| --- | --- | --- | --- |
| 测试  用例 | 程序输入 | 理论结果 | 运行结果 |
| 用例1 | A | a | a |
| 用例2 | c | c | c |
| 用例3 | // | // | // |
| 用例4 | 12 | 12 | 12 |
| 用例5 | Ctrl+z | 结束 | 结束 |

（b） 对应测试数据的运行结果截图

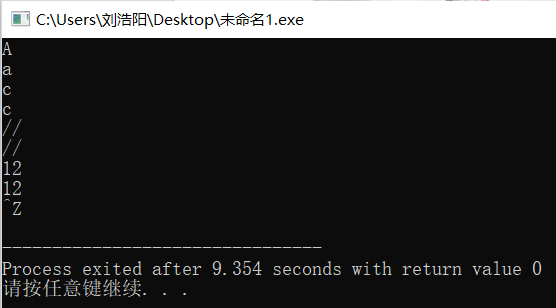


图1-3 程序设计题1的测试用例1、2、3、4、5的运行结果

（2）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（x从右至左编号为0～15），并使其向左端（第15位）靠齐。要求：①检查m和n的范围；②x的值以十六进制输入，m和n以十进制输入；③结果以十六进制输出。

**解答：**

1. 解题思路：如图1-4所示



图1-4程序设计题2的程序流程图

1. 程序清单：

#include<stdio.h>

int main()

{

unsigned short x,m,n;

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

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

printf("%hx",x<<=(16-m-n));

else printf("error");

return 0;

}

3）测试

（a） 测试数据：如表1-2所示

表1-2 编程题2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 | 0010 1000 0001 0100（2814） | 5 | 6 | 280 | 280 |
| 用例2 | 1011 0000 1111 1010（b0fa） | 13 | 2 | 61f4 | 61f4 |
| 用例3 | 1111 0000 1100 0011（f0c3） | 13 | 5 | error | error |

（b） 对应测试测试用例一的运行结果如图1-5所示。

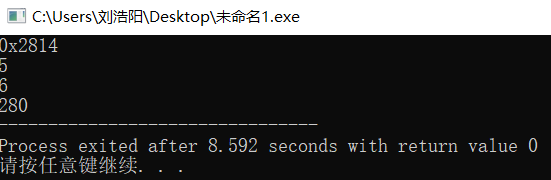


图1-5 程序设计题2的测试用例1的运行结果

对应测试测试用例二的运行结果如图1-6所示。

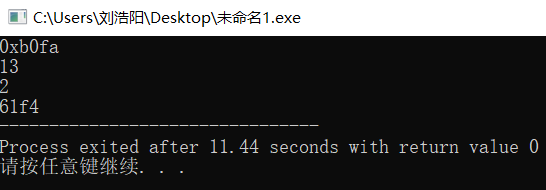


图1-6 程序设计题2的测试用例2的运行结果

对应测试测试用例三的运行结果如图1-7所示。

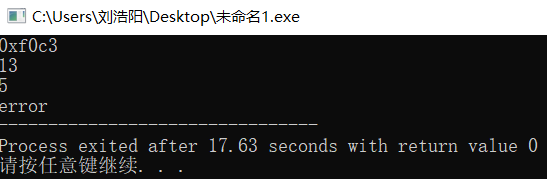


图1-7 程序设计题2的测试用例3的运行结果

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）

读入无符号长整型数表示的互联网IP地址，对其译码，以常见的点分十进制形式输出。要求循环输入和输出，直至输入Ctrl+Z结束。

**解答**：

1. 解题思路：如图1-8所示



图1-8 程序设计题3的程序流程图

1. 程序清单

#include<stdio.h>

int main()

{ unsigned long IP,A,B,C,D;

const unsigned long a=0x000000ff;

const unsigned long b=0x0000ff00;

const unsigned long c=0x00ff0000;

const unsigned long d=0xff000000;

while(scanf("%lu",&IP)!=EOF){

A=a&IP;

B=b&IP;B>>=8;

C=c&IP;C>>=16;

D=d&IP;D>>=24;

printf("%lu.%lu.%lu.%lu\n",D,C,B,A);

}

return 0;

}

（a）测试数据如表1-3所示

表1-3 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 |
| 用例1 | 3232235876 | 192.168.1.100 | 192.168.1.100 |
| 用例2 | 66322345376 | 113.30.173.160 | 113.30.173.160 |
| 用例3 | 1112223333 | 66.75.46.101 | 66.75.46.101 |
| 用例4 | 66322345376 1112223333 | 113.30.173.160 66.75.46.101 | 113.30.173.160 66.75.46.101 |
| 用例5 | Ctrl+z | 结束 | 结束 |

（b） 对应测试测试用例一的运行结果如图1-9所示。

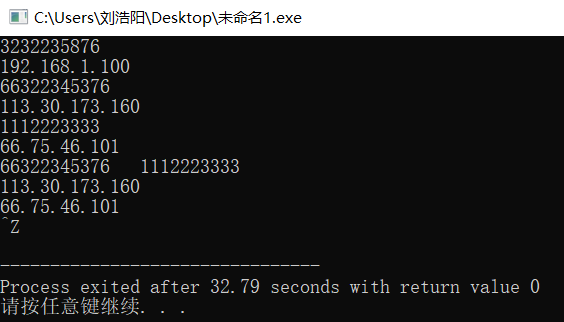


图1-9 程序设计题3的测试用例1的运行结果

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

由于本次是第一次实验，实验前对实验题目也没有过多的了解和尝试，导致实验时遇到了很多困难。其中遇到了一些不熟悉的循环语句的使用，比如while if等语句，这方面确实预习不够。还有一部分就是因为没有思路，比如实验1-3对mask的作用第一时间没有想到，导致刚开始做的时候毫无头绪，其中对EOF的使用方法也不是很清楚，后来在自己的查找书籍和与同学的讨论下才勉强解决。同时，在与同学的交流中，也更加熟悉了mask的作用，比如实验1-3中我定义了4个mask，有的同学只定义了1个mask同样也能实现目标，这对我简化代码提供了一定思路。这次实验，虽然没有在实验课上按时完成，但回去后也是很快解决了，是令我比较开心的，但同时也让我认识到了自己知识的不足与能力的欠缺，今后要增强对书本的预习复习包括自己的实践动手能力。

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] 卢萍,李开,王多强等. C语言程序设计典型题解与实验指导, 北京：清华大学出版社, 2019