

**课 程 实 验 报 告**

**课程名称： C语言程序设计实验**

**专业班级： CS1712**

**学 号： U201714859**

**姓 名： 吴 柯**

**指导教师： 李 平**

**报告日期： 2017/11/21**

**计算机科学与技术学院**

**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（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void和main之间没有空格，正确形式为：

void main( void )

3) 第6行newint变量未定义，正确形式为：

short p, k, newint;

1. 第9、10、12行引号应使用半角引号

5) 第10行f前未加取地址符号，正确形式为：

scanf(“%d”, &f);

6) 第11行为整型数字运算，结果为整数，正确形式为：

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

7) 第15行double类型的读入应用%lf，正确形式为：

scanf("%lf", &r);

8) 第17行s前不应有取地址符号，且输出应用%lf，正确形式为：

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

9) 第20行短整数的16进制读入应用%hx，正确形式为：

scanf("%hx %hx", &k, &p );

10) 第21行移位逻辑错误，正确形式为：

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

（2）错误修改后运行结果：

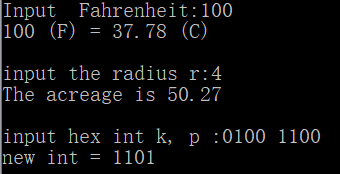


图1-1 改错题的程序运行结果

**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：**

将交换的语句改为a = a + b； b = a – b; a = a – b; 替换后的程序如下所示：

#include<stdio.h>

int main( )

{

int a, b;

printf(“Input two integers:”);

scanf(“%d %d”,&a,&b);

a = a + b； b = a - b; a = a - b;

printf(“a=%d,b=%d\n”,a,b);

return 0;

}

运行结果

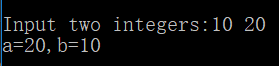


图1-2 替换题的程序运行结果

**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。

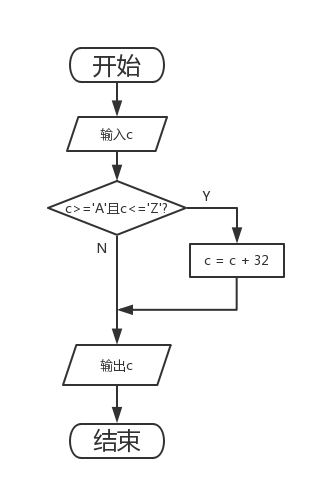


图1-3 编程题1程序流程图

2）源程序清单

#include <stdio.h>

int main()

{

char c;

scanf("%c", &c);

if(c >= 'A'&& c <= 'Z')

c += 32;

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

return 0;

}

3）测试

（a） 测试数据：

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 |
| c |
| 用例1 | A | a | a |
| 用例2 | a | a | a |
| 用例3 | ; | ; | ; |

（b）对应测试

测试用例1的运行结果 如图1-4所示。



图1-4 编程题1的测试用例1的运行结果

测试用例1的运行结果 如图1-5所示。



图1-5 编程题1的测试用例2的运行结果

测试用例1的运行结果 如图1-6所示。



图1-6 编程题1的测试用例3的运行结果

（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：

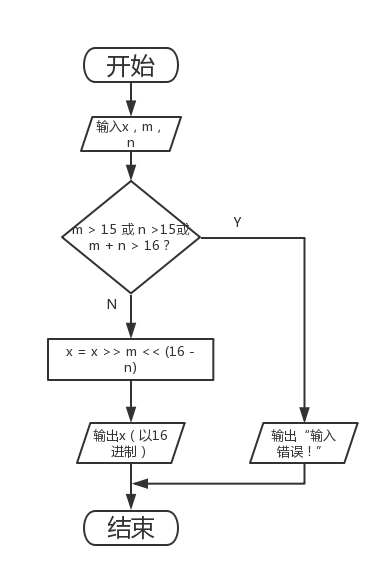


图1-7 编程题2的程序流程图

2）程序清单

#include <stdio.h>

int main()

{

unsigned short x, m, n;

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

if(m > 15 || n >15 || m + n > 16)

printf("输入错误！");

else

{

x = x >> m << (16 - n);

printf("%hx", x);

}

return 0;

}

3）测试

（a） 测试数据：

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 | 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） |  |
| 用例3 | 1101 0101 1000 0011（D583） | 13 | 5 | 输入错误（n值超范围） |  |

（b） 对应测试

测试用例1的运行结果如图1-8所示。

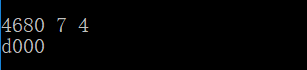


图1-8 编程题2的测试用例1的运行结果

测试用例2的运行结果如图1-9所示。

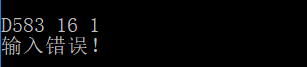


图1-9 编程题2的测试用例2的运行结果

测试用例3的运行结果如图1-10所示。

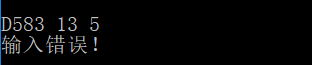


图1-10 编程题2的测试用例3的运行结果

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）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. 解题思路：

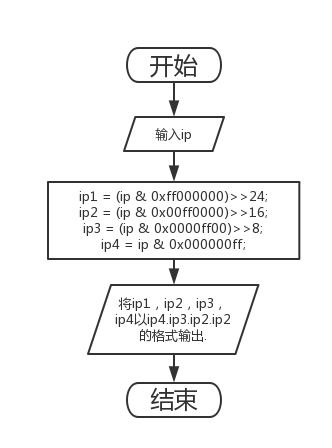


图1-11 编程题3的程序流程图

2）程序清单

#include <stdio.h>

int main()

{

unsigned int ip, ip1, ip2, ip3, ip4;

scanf("%u", &ip);

ip1 = (ip & 0xff000000)>>24;

ip2 = (ip & 0x00ff0000)>>16;

ip3 = (ip & 0x0000ff00)>>8;

ip4 = ip & 0x000000ff;

printf("%d.%d.%d.%d",ip4,ip3,ip2,ip1);

return 0;

}

3）测试

（a） 测试数据：

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 |
| ip |
| 用例1 | 10 | 10.0.0.0 | 10.0.0.0 |
| 用例2 | 676879571 | 211.92.88.40 | 211.92.88.40 |
| 用例3 | 4294967295 | 255.255.255.255 | 255.255.255.255 |

（b） 对应测试

用例1的运行结果如图1-12所示。



图1-12 编程题3的测试用例2的运行结果

用例2的运行结果如图1-13所示。

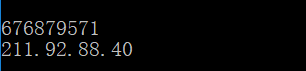


图1-13 编程题3的测试用例2的运行结果

对应测试用例3的运行结果如图1-14所示。

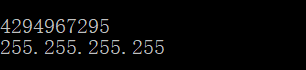


图1-14 编程题3的测试用例3的运行结果

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。

## 1.3 实验小结

遇到的问题：

在做编程题第二题时由于没有注意到题目中读入的为短整数而导致出错，经过再次读题之后解决了问题；编程题第三题由于没有理解译码的概念而导致输出结果反了，通过询问助教解决了问题。

体会：

在做题的时候一定要注意读题，抓住题目中的关键点之后再开始编程，遇到不懂的问题或概念一定要及时询问老师同学或查阅资料。