

**课 程 实 验 报 告**

**课程名称： C语言程序设计**

**专业班级： cs1601**

**学 号： U201614526**

**姓 名： 田志伟**

**指导教师： 甘早斌**

**报告日期： 2017.3.20**

**计算机科学与技术学院**

# 实验1 表达式和标准输入与输出实验

# 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 intmain( 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);

23 return 0;

}

**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intmain( void )，正确形式为：

int main()

3）第9行的printf(“Input Fahrenheit:” ) ;正确形式为：

printf("Input Fahrenheit:" ) ;

4）第10行的scanf(“%d”, f ) ;正确形式为：

scanf(“%d”, &f ) ;

5）第17行printf("\nThe acreage is %.2f\n\n",&s);正确形式为：

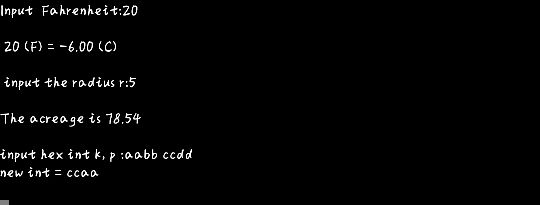
printf("\nThe acreage is %.2f\n\n",s);

6）第20，21之间应加入int newint；

7）第21行newint = (p&0xff00)|(k&0xff00)<<8;正确形式为：

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

（2）错误修改后运行结果：

****

**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>

int 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);

return 0;

}

**解答：**

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);

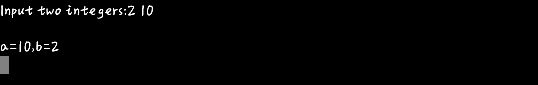
a+=b ；b=a-b；a-=b；

prinf(“\na=%d,b=%d”,a,b);

return 0;

}

错误修改后运行结果:



**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

开始

输入字母“word”

word为大写字母

Y

N

输出word word=word的小写形式

结束

图1-1 编程题1的程序流程图

1. 源程序清单
2. #include <stdio.h>

2 int main()

3 {

4 char word;

5 scanf("%c", &word);

6 int asc=word;

7 if(asc<='Z'&&asc>='A')

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

9 else

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

11

12 return 0;

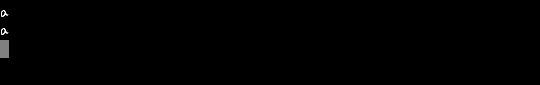
13 }

3）测试

（a） 测试数据：

A a

（b） 对应测试数据的运行结果截图

（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：

1.输入x，m，n, mask1,mask2

2.如果0 ≤ｍ≤ 15, 1 ≤ ｎ≤ 16-ｍ，转2.1，否则转3.

2.1 首先mask1=~0<<m，mask2=~0>>(16-m-n)创建两个中间m-n位为1的逻辑尺。

2.2 再将x与逻辑尺取与：x=x&mask1&mask2;

2.3 输出结果并转4.

3. 显示输入错误信息；

4. 结束

2）程序清单

#include <stdio.h>

int main()

{

unsigned short x, m, n;

scanf("%hu", &x);

scanf("%hu", &m);

scanf("%hu", &n);

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

{

unsigned short mask1, mask2;

if(m<1)

mask1=~0;

else

mask1=(~0)<<m;

mask2=(~0)>>(16-m-n);

x = x&mask1&mask2;

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

printf("%hu\n", x);

}

else

{

printf(“worng!\n”);

}

return 0;

}

3）测试

（a） 测试数据：

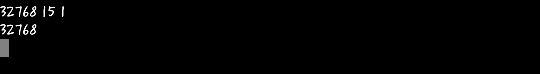
叙述选择测试数据的方法。。。如表1-1所示。

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 | 32768 | 15 | 1 | 计算结果1101 0000 0000 0000 即D000 | 32768 |
| 用例2 | 1 | 16 | 1 | 输入错误（m值超范围） | worng! |
| 用例3 | 1 | 13 | 5 | 输入错误（n值超范围） | worng! |

（b） 对应测试测试用例1的运行结果如图1-2所示。

图1-2 编程题3的测试用例一的运行结果



对应测试测试用例2的运行结果如图1-3所示。



图1-3 编程题3的测试用例二的运行结果

对应测试测试用例3的运行结果如图1-4所示。



图1-4 编程题3的测试用例三的运行结果

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。

## 1.4 实验小结

本次实验中遇到了许多困难，也发现了自己的一些漏洞，在实验过程中审题不够仔细，代码阅读也不够认真，导致出现了许多小bug，在以后的实验里我应该认真审题，细心才能完成好每一次的题目。