# 实验1 表达式和标准输入与输出实验

（黑体小2加粗居中, 字母、阿拉伯数字为Time New Roman小2号加粗,段前0.5行）

## 1.1 实验目的 （二级标题黑体4号加粗, 字母、阿拉伯数字为Time New Roman4号加粗，段前0.5行）

(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。

（正文部分：宋体小4号，行间距固定1.5倍行距，字符间距为标准）

## 1.2 实验内容

**1.2.1 源程序改错** (小四加粗，将所有的必做题和选做题按三级标题顺序编号)

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> (程序改错题，加上语句行号，便于后面的错误说明)

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：

×××××××

2) 第3行的×××××××，正确形式为：

×××××××

**.............**

（2）错误修改后运行结果：

(给一个运行截图简单说明修改方案的正确性)

**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：**

×××××××，×××××××，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=×× ；b=××；a=××；

prinf(“\na=%d,b=%d”,a,b);

}

(给一个运行截图简单说明修改方案的正确性)

**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1-1 编程题1的程序流程图

（图标题：位于图下方，居中黑体小4号，字母、阿拉伯数字为Time New Roman小4号）

2）源程序清单

××××××××××××××××××

××××××××××××××××××

3）测试

（a） 测试数据：

××××××××××××××××××

（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：(可用流程图代替)

1.输入x，m，n，为了方便分析测试结果，x的输入采用16进制

2.如果0 ≤ｍ≤ 15, 1 ≤ ｎ≤ 16-ｍ，转2.1，否则转3.

2.1 首先x>>m，将要处理的n位移动到最右；

2.2 再将上一步的结果左移×××××位，即： ×××××

2.3 用16进制输出结果并转4.

3. 显示输入错误信息；

4. 结束

2）程序清单 (注意程序的缩进编排、关键位置加上注释)

#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 (。。。。。。。。。。。。。。。。。。。。) /\*判断m、n的值是否在合理范围内\*/

printf("ans=%hx\n",。。。。。。。。。。。。);

else printf("输入错误!\n");

}

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(注意测试数据的覆盖性) | | | 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 或 截图 |
| 用例2 | 1101 0101 1000 0011（D583） | 16 | 1 | 输入错误（m值超范围） | （运行结果这栏也可以不要，在此表后面给出各类测试用例的截图 |
| 用例3 | 1101 0101 1000 0011（D583） | 13 | 5 | 输入错误（n值超范围） | 如后面的图1-2等等） |

（表标题：位于表格上方，黑体小4号，字母、阿拉伯数字为Time New Roman小4号，表内容：宋体5号，字母、阿拉伯数字为Time New Roman 5号，处理方法是先全部选择后用黑体或宋体，在全部选择后用Time New Roman ）

（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

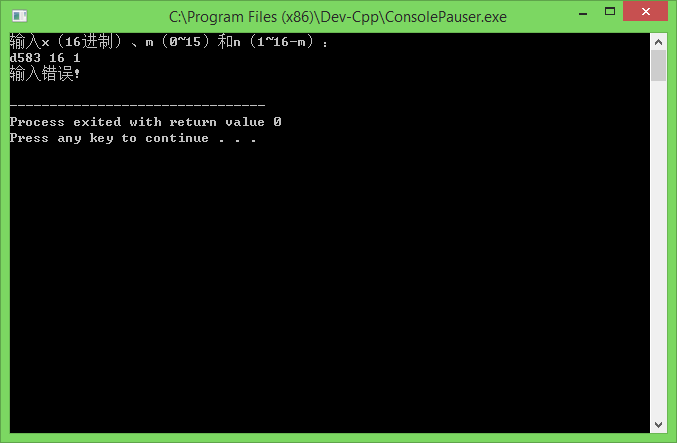
对应测试测试用例2的运行结果如图1-3所示。

图1-3 编程题3的测试用例二的运行结果

（图标题：位于图下方，黑体小4号，字母、阿拉伯数字为Time New Roman小4号）

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。

## 1.3 自设题（这部分不是必须的，作为奖励额外的奖励分，如果没有就不要这个标题）

**（1）** 自设实验题目：实型数的浮点表示误差的简单验证

**（2）** 实验目的：通过设计实验程序，理解实数的浮点数表示上的误差，浮点数不宜做等值比较。

**（3）**实验程序：

#include "stdio.h"

int main()

{

float a=10.33;

double b=10.33;

printf("%d\n",a==b);

}

**（4）**实验用例：～

**（5）**实验结论：～

## 1.4 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。