

**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：**

**软件学院**

**目 录**

[1 表达式和标准输入与输出实验 2](#_Toc85722382)

[1.1 实验目的 2](#_Toc85722383)

[1.2 实验内容 2](#_Toc85722384)

[1 源程序改错 2](#_Toc85722385)

[2 程序设计 4](#_Toc85722386)

[1.3 实验小结 8](#_Toc85722387)

# 

# 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 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，应改为

#define PI 3.14159

2) 第5行的f类型应该为double，应改为

double f ;

3) 第10行f的类型为double，应改为

scanf(“%d”,&f);

4) 第11行c的类型为double，应改为

c = 1.0 \* 5/9 \* (f-32) ;

5) 第12行c的类型为double，应改为

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

6) 第15行r的类型为double，应改为

scanf("%lf", &r);

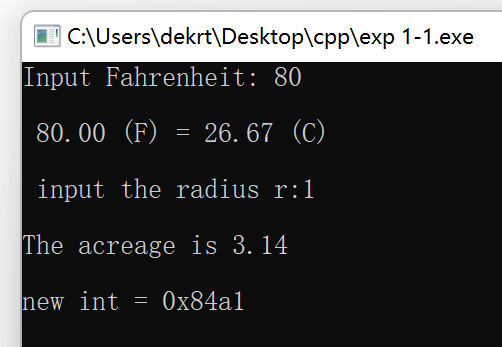
7) 第17行存在printf语法错误，应改为

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

8）第21行存在newint类型、算法错误，应改为

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

（2）错误修改后运行结果：



### 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程序结束。

**解答：**

1） 算法流程如图1.1所示。

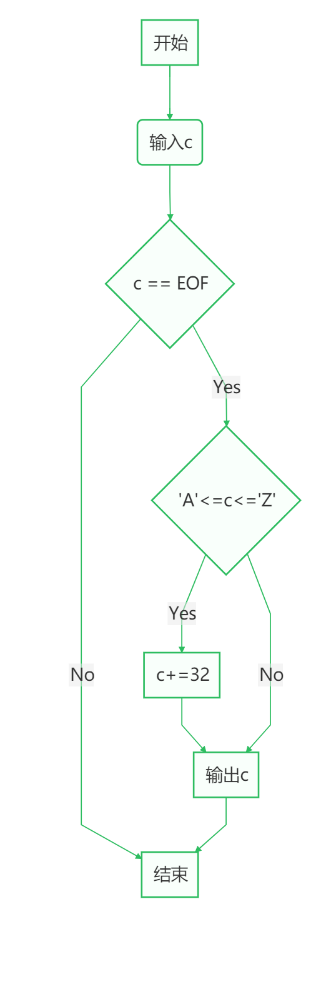


图1-1 编程题1的程序流程图

2）源程序清单

#include <stdio.h>

int main()

{

char c ;

while ( scanf("%c",&c) != EOF )

{

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

{

c += 32 ;

printf("%c",tmp) ;

}

else printf("%c",c) ;

}

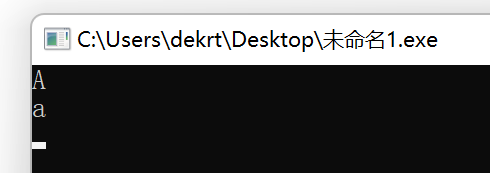
return 0;

} 3）测试

（a） 测试数据：

A

（b） 对应测试数据的运行结果截图



（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（ｍ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:

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15-n-1位，即：(x >> m ) << (15 - n - 1) 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

#include <stdio.h>

#define re register

int main()

{

unsigned short int x , m , n , ans ;

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

// return 0;

if (m < 0 || m > 15 || m + n <0 || m + n >16)

{

printf("error");

return 0;

}

ans = (x >> m ) << (15 - n - 1) ;

printf("%x",ans ) ;

return 0;

}

3）测试

（a） 测试数据：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 |
| X | m | N |
| 用例1 | 0100 0110 1000 0000（4680） | 7 | 4 | 计算结果1101 0000 0000 0000 即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1-2所示。



图1-2 编程题3的测试用例一的运行结果

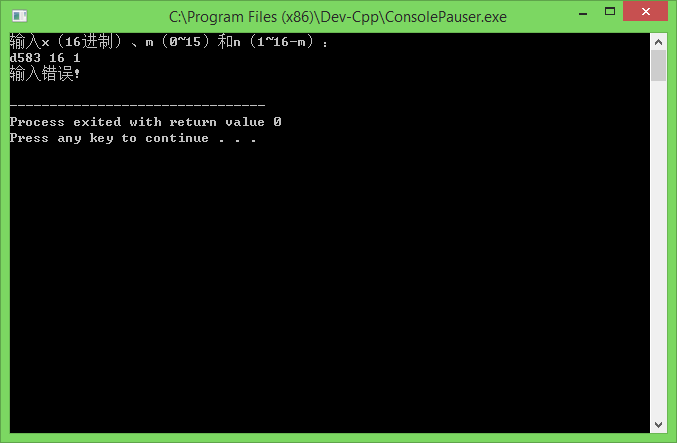
对应测试测试用例2的运行结果如图1-3所示。

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。

（3）IP地址通常是4个用句点分隔的小整数（即点分十进制），但这些地址在机器中是用一个无符号长整型数表示的。例如3232235876，其机内二进制表示就是11000000 10101000 00000001 01100100，按照8位一组用点分开，该IP地址就写成192.168.1.100。

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:

1.输入x为了方便分析测试结果，x的输入采用unsigned long int型

2.如果x != EOF转2.1，否则转3.

2.1 首先定义mask = 0x000000ff ;

2.2计算结果并进行输出，(x>>24)&t. (x>>16)&t. (x>>8)&t . (x&t),

然后转3：

3. 结束

2）程序清单

#include <stdio.h>

#define re register

unsigned long int x;

int main()

{

while (scanf("%ld",&x) != EOF )

{

unsigned long int t = 0x00000ff ;

printf("%d.%d.%d.%d\n",(x>>24)&t,(x>>16)&t,(x>>8)&t,(x&t)) ;

}

return 0;

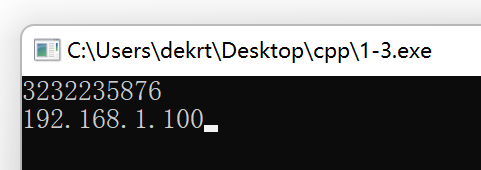
}

3）测试

（a） 测试数据：

3232235876

（b） 对应测试数据的运行结果截图



## 1.3 实验小结

实验中对于部分算法、语法存在许多疑惑，但经过查阅资料、自己摸索、借助ide的debug、调试等功能，成功解决了问题。经过这次实验，我体会到了自学探索的重要性，对有关知识的理解更深了。