

**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**

# 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. void main( 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. }

**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5行的f类型错误，正确形式为：double f ;

3)第10行的格式符错误，正确形式为：scanf("%lf",&f);

4)第 11 行 c 的类型为 double，正确形式为：c = 5.0/9 \* (f-32) ;

5)第 12 行格式符错误，正确形式为：printf( "\n %.2lf (F) = %.2lf (C)\n\n ", f, c ) ;

6)第 15 行格式符错误，正确形式为：scanf("%lf",&r);

7)第 17 行printf用法错误，输出不应是地址，正确形式为：printf("\nThe acreage is %.2lf\n\n",s)

8)第 21 行使用位声明变量，且算法错误，正确形式为：int newint = (p&0xff00) | ((k&0xff00)>>8) ;

1. 错误修改后运行结果：

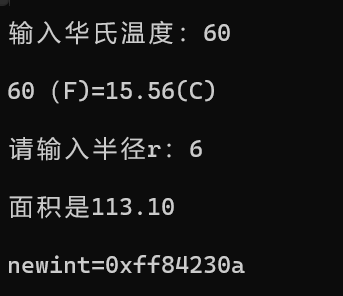


图 1-1 错误修改后的程序运行结果示意图

### 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1.2所示



图1-2 编程题1的程序流程图

1. 程序清单
2. #include<stdio.h>
3. int main()
4. {
5. char c;
6. while((c=getchar())!=EOF)
7. {
8. if(c>='A'&&c<='Z')*//检验范围*
9. putchar(c+32);*//根据ASCII表，大小写字母差值为32*
10. else
11. putchar(c);
12. }
13. return 0;
14. }
15. 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 |
| 用例1 | A | a |
| 用例2 | b | b |
| 用例3 | C | c |
| 用例4 | Ctrl+Z | 程序退出 |

（b） 对应测试数据的运行结果截图

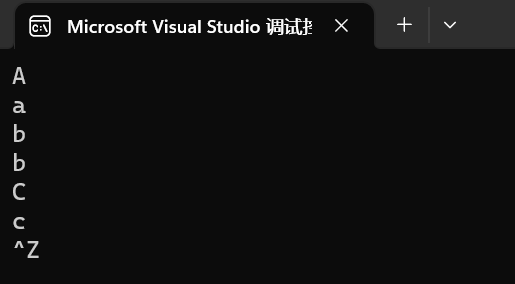


图1-3 编程题1的测试用例的运行结果

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。

（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. 算法流程如图1.4所示

。

图1-4 编程题2的程序流程图

1. 程序清单
2. #include <stdio.h>
3. int main() {
4. unsigned short x;
5. unsigned short m, n;
6. scanf("%hx%hd%hd", &x,&m,&n);
7. if(m>15||m<0||n<1||n>16-m)
8. {
9. printf("输入错误！");
10. }
11. else{
12. unsigned short a = (x>>(m-1));*//顶到最右边*
13. unsigned short result = (a<<(15-n));*//向左靠齐*
14. printf("%hx", result);}
15. return 0;
16. }
17. 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） | 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） | 输入错误（m值超范围） |

（b） 对应测试测试用例1的运行结果如图1-5所示。

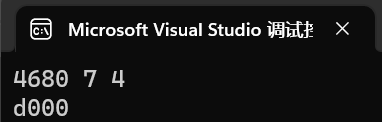


图1-5 编程题2的测试用例一的运行结果

对应测试测试用例2的运行结果如图1-6所示。

## QQ截图20231009143744

图1-6 编程题2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1-7所示

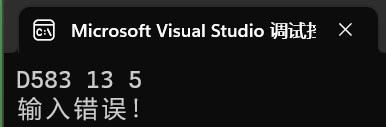


图1-7 编程题2的测试用例三的运行结果

## 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） 算法流程如图1.8所示。



图1-8 编程题3的程序流程图

2）源程序清单

1. #include <stdio.h>
2. int main()
3. {
4. int x;
5. while (scanf("%d", &x) != EOF) {
6. int a = (x >> 24) & 0xFF;  *// 第一个八位*
7. int b = (x >> 16) & 0xFF;  *// 第二个八位*
8. int c = (x >> 8) & 0xFF;   *// 第三个八位*
9. int d = x & 0xFF;          *// 第四个八位*
10. printf("%d.%d.%d.%d\n", a, b, c, d);
11. }
12. return 0;
13. }

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 |
| 用例1 | 3232235876 | 192.168.1.100 |
| 用例2 | 66322345376 | 113.30.173.160 |
| 用例3 | 187232376 | 11.40.240.120 |

（b） 对应测试数据的运行结果截图

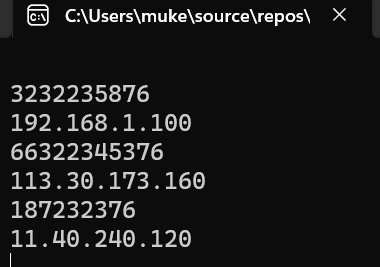
 ×

图1-9 编程题3的测试用例的运行结果

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

在完成这次实验后，我获得了以下几点体会和经验：

1. **重视细节，善于纠错**： 在C语言中，细节非常重要。一个小小的语法错误或逻辑错误都可能导致程序无法正常工作。通过这次实验，我了解了各种常见错误的原因和报错显示，例如全半角逗号分号会报错未识别字符，scanf后面的变量忘记加&会报错C4477,#define后加分号可能导致后面变量被识别为指针，VS2022需要使用scanf\_s以保证安全性等等。
2. **输入输出函数的熟练使用**： scanf 和 printf 是C语言中常用的输入输出函数。通过实验，我熟悉了几种常见格式字符的用法，和两个函数的使用方法
3. **程序结构：** 这个实验强调了顺序结构程序的编写。在实际编程中，理解程序的结构和流程对于正确完成任务非常重要。同时，实验还涉及了循环输入，通过实验，我了解了while的用法和EOF的概念，学会如何循环输入和结束输入，对程序的输入输出有了更深的了解。
4. **流程图的画法：**通过实验，我了解visio软件的基本使用，熟悉了用流程图表示程序结构，整理思路。