# 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行的函数主题部分少了空格，正确形式为：

void main(void)

3) 第10行的赋值语句错误，正确格式为：

scanf(“%d”,& f ) ;

4) 第11行数字输入格式错误，正确格式为：

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

5) 第15行数字输入形式不对，正确形式为：

scanf("%lf", &r);

6）第17行的输出值不对，正确形式为：

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

7）第21行的移位不对而且输入格式不对，正确形式为：

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

（2）错误修改后运行结果：

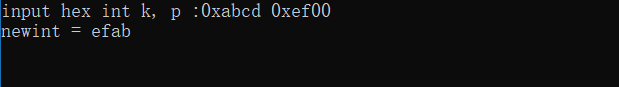
Task1:



Task2:



**Task3:**



### 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：**

可以先将a与b的和赋给a，然后减去b赋值给b，再将a减去

b赋值给a，替换后的程序如下所示：

#include<stdio.h>

void main( )

{

int a, b;

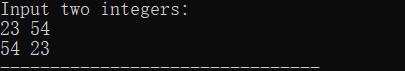
printf(“Input two integers:”);

scanf(“%d %d”,&a,&b);

a=a+b；b=b-a；a=a-b；

prinf(“\na=%d,b=%d”,a,b);

}



### 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。

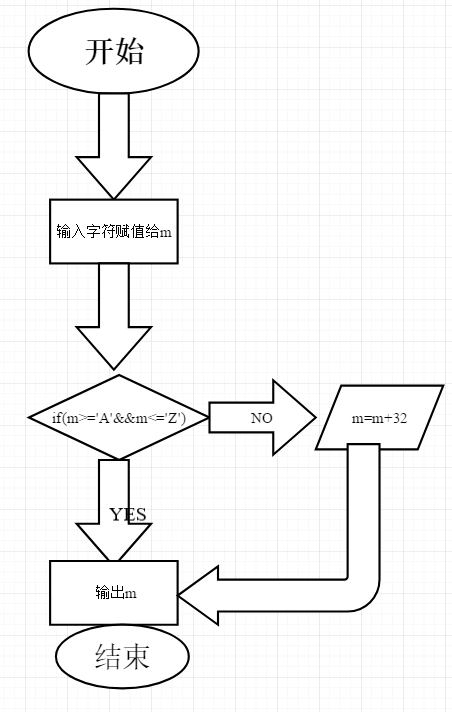


图1-1 编程题1的程序流程图

2）源程序清单

#include "stdio.h"

int main()

{

int m;

m=getchar();

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

{

m=m+32;

putchar(m);

}

else

putchar(m);

return 0;

}

3）测试

（a） 测试数据：

输入‘d’，输出‘d’；

输入‘N’，输出‘d’；

（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）程序清单

#include "stdio.h"

int main()

{

unsigned short x,m,n;

printf("输入x(十六进制)，m(0~15)和n(1~16-m):\n");

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

if(m>=0 && m<=15 && n>=1 && n<=(16-m)) //判断输入是否符合题意

{

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

printf("%0X",x);

}

else

{

printf("输入错误！\n");

}

return 0;

}

2）解题思路

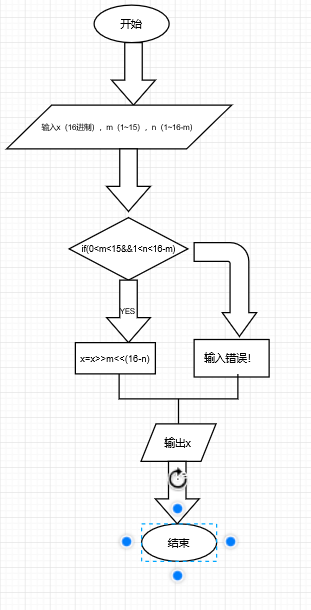


图1-2 编程题2的程序流程图

3）测试

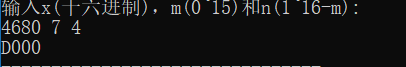
（a） 测试数据：

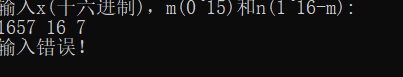
输入4680；7；4；输出：D000

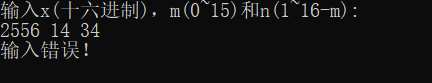
输入1657；16；4；输出：输入错误！

输入2556；14；34；输出：输入错误！

（b）运行效果截图







（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. 算法流程

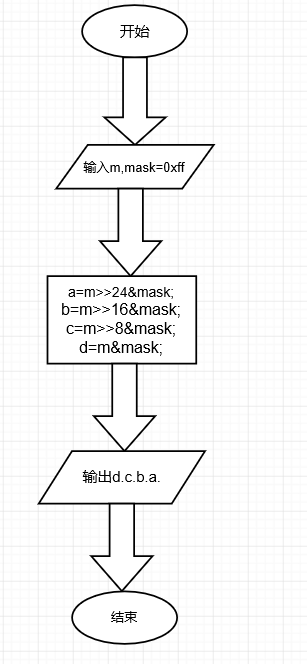


图1-3 编程题3的程序流程图

1. 编译代码

#include "stdio.h"

int main()

{

unsigned long m;

printf("Please printf your number:");

scanf("%d",&m);

unsigned mask=0xff;

int a,b,c,d;

a=m>>24&mask;

b=m>>16&mask;

c=m>>8&mask;

d=m&mask;

printf("%d.%d.%d.%d",d,c,b,a);

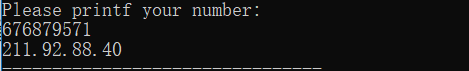
return 0;

}

1. 测试
   1. 测试数据

输入：676879571；输出：211.92.88.40；

（b）运行效果截图



## 1.3实验小结

1）本次实验为第一次实验，所以操作起来遇到了很多困难，实验课上由于对C语言的不太了解，导致没能在课堂上及时让助教检查，没能在规定时间完成任务。但是在课下我抽出了很多时间来完成作业，并且问同学，最后完成了作业。这警示我要努力学习C语言，一定要多下功夫来完成任务。

2）由于自己的打字速度很慢，所以效率就不高，在以后一定要多练习，在实践中提高自己。

3）C语言的学习告诉了我做事一定要严谨，因为你的一个标点符号的错误就会导致程序的失败，然而你却不知道自己错在哪里。

4）实验报告的书写更加让我了解了自己的程序每一步表示什莫意思，更加深入的了解了C语言，并且激发了我学习C语言 的兴趣，看着自己的实验报告，感觉好有成就感呀！