**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2.1.3和2.1.4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。

**程序代码：**

#include<stdio.h>

#define PI 3.14159;

voidmain( void )

{

int f ;

short p, k ;

double c , r , s ;

/\* for task 1 \*/

printf(“Input Fahrenheit:” ) ;

scanf(“%d”, f ) ;

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

printf( “ \n %d (F) = %.2f (C)\n\n ”, f, c ) ;

/\* for task 2 \*/

printf("input the radius r:");

scanf("%f", &r);

s = PI \* r \* r;

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

/\* for task 3 \*/

printf("input hex int k, p :");

scanf("%x %x", &k, &p );

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

printf("new int = %x\n\n",newint);

}

【**分析及改正**】

（1）#define PI 3.14159；

错误原因：宏定义在预f处理时会将后续程序中的标示符展开为对应全部字符串。

改正：去掉其后分号

（2）voidmain(void) {

错误原因：函数返回值与函数名之间应加空格，C90标准建议main函数应返回一个int作为状态值

改正：int main(void) {

（3）short p,k;

错误原因：后续用到了newint变量但并未声明

改正：short p,k,newint;

（4）scanf(“%d”, f );

错误原因：应该把要读入的变量的地址传递给scanf才能正确写入

改正：scanf("%d",&f);

（5）c = 5/9\*(f-32) ;

错误原因：5和9是整型常数，相除会截断小数部分，应该将其中任意一个输表示为或转换为浮点数类型

改正：c = 5.0/9\*(f-32);

（6）scanf("%f", &r);

错误原因：double类型对应的占位符是%lf

改正：scanf("%lf",&r);

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

错误原因：要输出s的值应该将s作为参数传递给printf，不应传递其地址

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

（8）scanf("%x %x", &k, &p );

printf("new int = %x\n\n",newint);

错误原因：k,p为short类型，对应十六进制占位符应为%hx

改正：scanf("%hx %hx",&k,&p);

printf("new int = %hx\n\n",newint);

（9）newint = (p&0xff00)|(k&0xff00)<<8;

错误原因：要求将k的高字节作为结果的低字节，应该将k的高字节提取出后右移至低字节

改正：newint=(p&0xff00)|(((k&0xff00)>>8)&0x00ff);

**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。

**程序代码：**

#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);

}

【原程序分析】

原程序使用中间变量t，将a的值存入t，再将b的值赋给a，最后从t中取出原来a的值赋给b，以实现交换a，b的值。

【修改**方案**】使用异或方式交换

#include <stdio.h>

int main(void){

int a,b;

printf("Input two integers:");

scanf("%d %d",&a,&b);

a = a+b;

b = a-b;

a = a-b;

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

return 0;

}

【**测试**】如果输入 12 21，应该输出：a=21，b=12，程序运行结果如图1.1所示，程序正确。

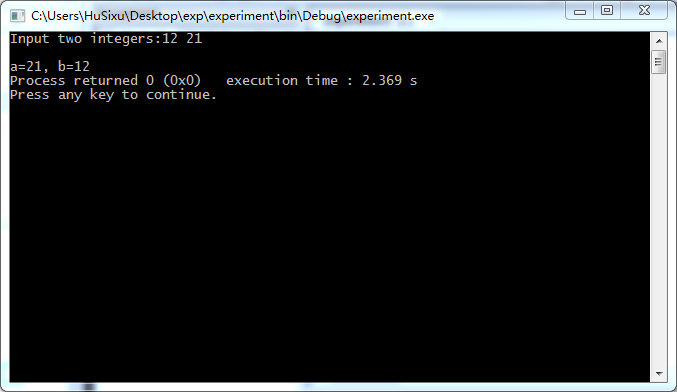


图1.1 交换两数的运行结果截图

**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。

【**分析**】因为同一个字母的大小写字母的ASCII码值相差32，若ch为大写字母，则ch+32为对应的小写字母。

【**程序**】

#include <stdio.h>

int main(void){

char ch;

puts("Enter a character:");

ch = getchar();

if(ch>='A'&&ch<='Z') // 若ch为大写，则变成小写

ch += 32;

putchar(ch);

return 0;

}

【**测试**】

1. 测试大写变小写。若输入B，应输出b，程序运行结果如图1.2所示，程序正确。

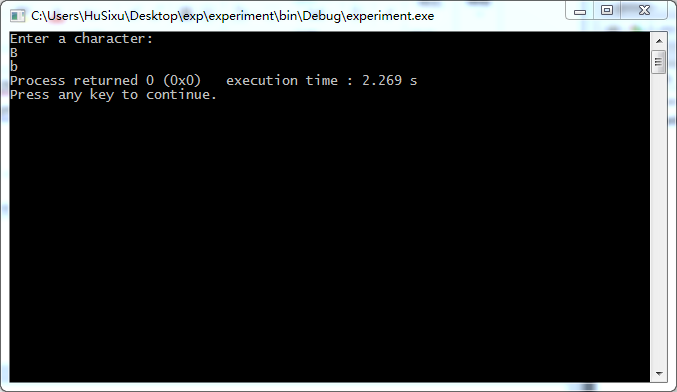


图1.2 字符转换的运行结果截图1

2. 测试其他字符。若输入a，应输出a，程序运行结果如图1.3所示，程序正确。

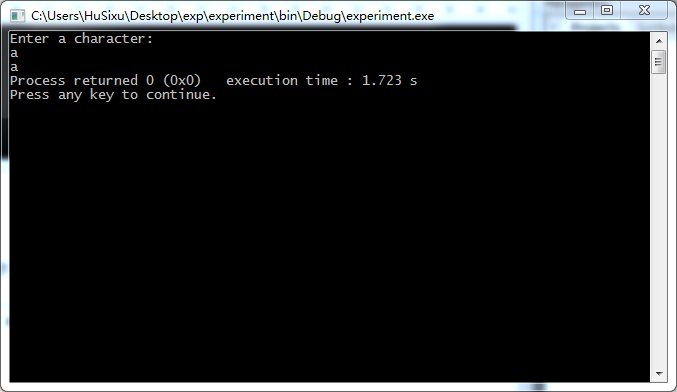


图1.3 字符转换的运行结果截图2

（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。

【**分析**】将x首先右移m位，再左移(16-m)位，则末尾会补0，即可将第m位开始向左的n位且向最高位靠齐

【**程序**】

#include <stdio.h>

int main(void){

unsigned short x,m,n;

puts("Please input x,m and n, separated by space:");

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

x>>=m;

x<<=(16 - n);

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

return 0;

}

【**测试**】若 x=58207，m=4，n=4，因为(58207)10 = (1110 0011 0101 1111)2，处理后为(101 0000 0000 0000)2= (5000)16 ，程序应该输出十六进制数：5000，运行结果如图1.4所示，程序正确。

（3）IP地址通常是4个用句点分隔的小整数（即点分十进制），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。例如，

整形676879571二进制表示就是：00101000 01011000 01011100 11010011，按照8位一组可表示为：40 88 92 211，由于CPU处理数据的差异，它的顺序是颠倒的，所有最终格式为211.92.88.40。

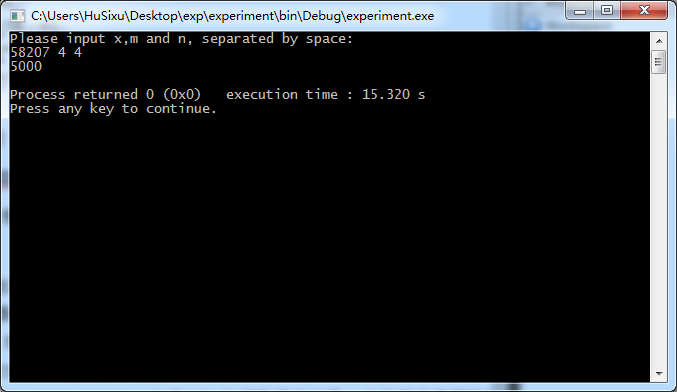


图1.4 程序设计第2题的运行结果截图

【**分析**】分别利用逻辑尺取出读入的长整形数的四部分，从最低位起每8位（1个字节）为一个部分，然后按照正常顺序输出其对应的十进制IP地址。

【**程序**】

#include <stdio.h>

int main(void){

unsigned long ipAdress;

unsigned short first, second, third, fourth;

puts("Input the IP Address in unsigned long int form:");

scanf("%lu",&ipAdress);

first = (ipAdress&0xFF000000)>>24;

second = (ipAdress&0xFF0000)>>16;

third = (ipAdress&0xFF00)>>8;

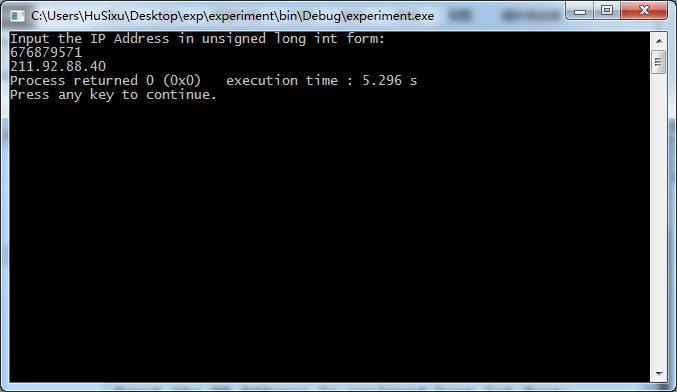
fourth = ipAdress&0xFF;

printf("%hu.%hu.%hu.%hu",fourth,third,second,first);

}

【**测试**】若输入676879571，应输出 211.92.88.40。运行结果如图1.4所示，表明程序正确。

图1.5 程序设计第3题的运行结果截图



**1.3 实验体会**

在做改错题时，我的方法是首先浏览一遍，找出明显的错误，然后将初步修改后的代码输入编译器并编译，再根据编译器的警告与错误提示修改程序至完全正确，这个过程很需要细心和耐心，这个过程能够很好的帮助我再今后不犯类似错误。

源程序修改与替换题锻炼我们举一反三的能力，我们在平时写程序时就应该思考对于一个问题有没有多钟解决方案，不使用中间变量交换两个整数是很常见的问题，我利用一个数异或自身等于0，0异或任意位保留原位的原理来交换。

程序设计题提高了我们对所学知识的熟练度和理解程度，只有实践才能学会C语言，在亲手编写代码的过程中发现问题，理解概念。