

**课 程 实 验 报 告**

**课程名称： C语言程序设计**

**专业班级： 物联网工程1601**

**学 号： U201614898**

**姓 名： 潘翔**

**指导教师： 刘芳**

**报告日期： 2017.3.23 ­­**

**计算机科学与技术学院**

**目录**

[实验1 表达式和标准输入与输出实验 1](#_Toc479497513)

[1.1 实验目的 1](#_Toc479497514)

[1.2 实验内容 1](#_Toc479497515)

[1.3 自设题 10](#_Toc479497516)

[1.4 实验小结 14](#_Toc479497517)

[实验2 流程控制实验 15](#_Toc479497518)

[2.1实验目的 15](#_Toc479497519)

[2.2实验内容 15](#_Toc479497520)

[2.3 实验小结 22](#_Toc479497521)

# 实验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voidmain，正确形式为：

void main

3) 第9行的“”，正确形式为：

“”

4) 第10行的“”，正确形式为：

“”

5) 第10行的f应该有&，正确形式为：

&f

6) 第11行的int运算有丢失应转成double，正确形式为：

(double) 5/9\*(f-32);

7) 第12行的“”，正确形式为：

“”

8) 第17行的&s，正确形式为：

s

9) 第15行的f，正确形式为：

lf

10) 第20行的short输入应为%hx而不是x，正确形式为：

hx

11) 第21行的short右移应为>>，正确形式为：

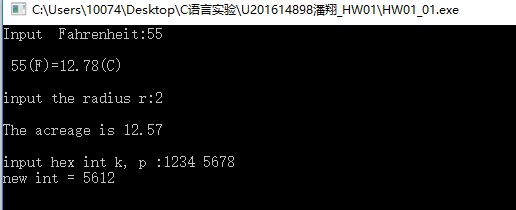
hx

12) 第22行的short输出应为%.4hx

，正确形式为：

%.4hx

（2）错误修改后运行结果：



**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：**

将t=a ；a=b；b=t；替换为a = b-a; b = b-a; a =b+a;替换后的程序如下所示：

#include<stdio.h>

void main( )

{

int a, b;

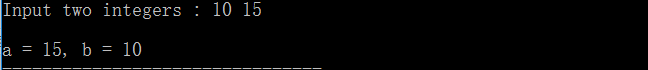
printf(“Input two integers:”);

scanf(“%d %d”,&a,&b);

a = b-a; b = b-a; a =b+a;

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()

{

char c;

c=getchar();

c=c>='A'&&c<='Z'?c+32:c;

putchar(c);

return 0;

}

3）测试

（a） 测试数据：

A a B b C c

（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：

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16-n位，即： x=x<<(16-n)

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



图1-2 编程题2的程序流程图

2）程序清单

#include<stdio.h>

int main()

{

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 (0 <= m && m <= 15 )

{

if (1 <= n && n <= 16 - n)

{

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

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

}

else

{

printf("n值超范围");

}

}

else

{

printf("m值超范围");

}

return 0;

}

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2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。

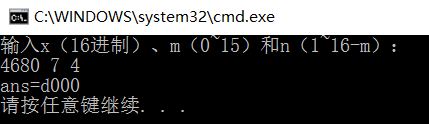


图1-2 编程题2的测试用例一的运行结果

对应测试测试用例2的运行结果如图1-3所示。

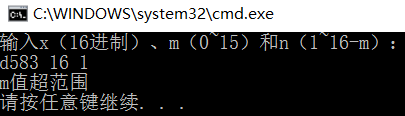


图1-3 编程题2的测试用例二的运行结果

对应测试测试用例3的运行结果如图1-4所示。

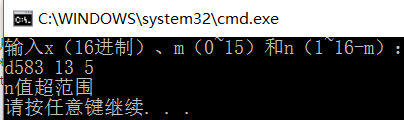


图1-4 编程题2的测试用例三的运行结果

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，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. 解题思路：



1. 程序清单

#include<stdio.h>

int main()

{

int N;

scanf("%d", &N);

char str\_ip\_index[4] = { '\0' };

unsigned long int IP;

unsigned long int IP1;

unsigned long int IP2;

unsigned long int IP3;

unsigned long int IP4;

for (int i = 1; i <= N; i++)

{

scanf("%lu", &IP);

IP1 = ((IP & 0xff000000) >> 24);

IP2 = ((IP & 0x00ff0000) >> 16);

IP3 = ((IP & 0x0000ff00) >> 8);

IP4 = ((IP & 0x000000ff));

printf("%lu.%lu.%lu.%lu\n", IP4, IP3, IP2, IP1);

}

return 0;

}

3）测试

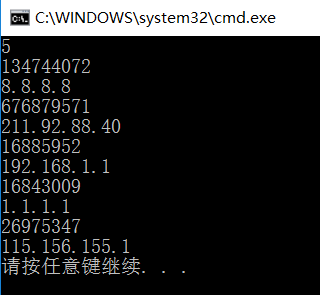
（a） 测试数据：

叙述选择测试数据的方法使数据覆盖尽量广泛 各节相同和不同如表1-2所示。

表1-2 编程题3的测试数据

|  |  |  |
| --- | --- | --- |
| 测试  用例 | 程 序 输 入 | 理 论 结 果 |
| X |
| 用例1 | 134744072 | 8.8.8.8 |
| 用例2 | 676879571 | 211.92.88.40 |
| 用例3 | 16885952 | 192.168.1.1 |
| 用例4 | 16843009 | 1.1.1.1 |
| 用例5 | 26975347 | 115.156.115.1 |

（b） 对应测试测试用例的运行结果如图1-2所示。



## 1.3 自设题

**（1）** 自设实验题目：编写一个程序, 对用户录入的产品信息进行格式化, 程序运行后需要有以下会话:

INPUT:

GroupNumber:1

Enter item number: 583

Enter unit price: 13.5

Enter purchase date(mm/dd/yy): 10/24/95

OUTPUT:

Item      Unit Price      Purchase Date

583        $ 13.50          10/24/95

其中, 数字项和日期项单位价格左对齐, 美元数量最大取值为9999.99

**（2）** 实验目的：通过设计实验程序，熟练标准输入输出，掌握输出表格的方法

**（3）**实验程序：

#include <stdio.h>

int main()

{

int GroupNumber=0;

int number[9] = {0};

float price[9] = { 0 };

int year[9] = { 0 };

int month[9] = { 0 };

int day[9] = { 0 };

printf("GroupNumber:");

scanf("%d", &GroupNumber) ;

for(int i=1;i<=GroupNumber;i++)

{

scanf("%d", &number[i]);

scanf("%f", &price[i]);

scanf("%d/%d/%d", &month[i], &day[i], &year[i]);

}

printf("Item \t Unit\t Purchase\n");

for (int i = 1; i <= GroupNumber; i++)

{

printf("%-9d$ %-9.2f%02d/%02d/%02d\n", number[i], price[i], month[i], day[i], year[i]);

}

getchar();

return 0;

}

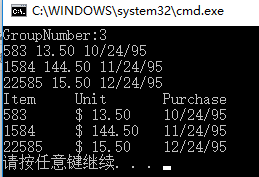
**（4）**实验用例：

3

583 13.50 10/24/95

1584 144.50 11/24/95

22585 15.50 12/24/95



**（5）**实验结论：

可以使用格式化输入输出制作表格，使打印清晰

## 1.4 实验小结

**（1）**对于可能工程中不太常见的输入输出格式不熟悉，通过练习得到了巩固。

**（2）**对于输入需要输出进行检测避免后续处理错误的输入数据导致DEBUG困难。

**（3）**学习使用测试用例和有效的测试方法。

**（4）**注重细节，培养习惯才能在工程代码中少出错。

# 实验2 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voidmain，正确形式为：

void main

3) 第9行的“”，正确形式为：

“”

4) 第10行的“”，正确形式为：

“”

5) 第10行的f应该有&，正确形式为：

&f

6) 第11行的int运算有丢失应转成double，正确形式为：

(double) 5/9\*(f-32);

7) 第12行的“”，正确形式为：

“”

8) 第17行的&s，正确形式为：

s

9) 第15行的f，正确形式为：

lf

10) 第20行的short输入应为%hx而不是x，正确形式为：

hx

11) 第21行的short右移应为>>，正确形式为：

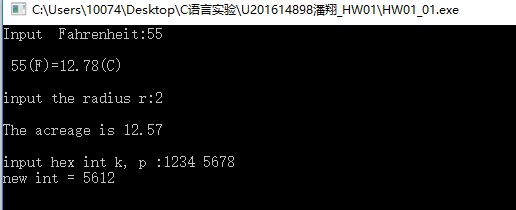
hx

12) 第22行的short输出应为%.4hx

，正确形式为：

%.4hx

（2）错误修改后运行结果：



**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：**

将t=a ；a=b；b=t；替换为a = b-a; b = b-a; a =b+a;替换后的程序如下所示：

#include<stdio.h>

void main( )

{

int a, b;

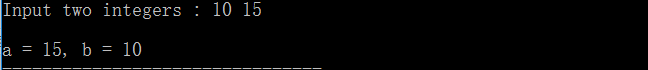
printf(“Input two integers:”);

scanf(“%d %d”,&a,&b);

a = b-a; b = b-a; a =b+a;

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()

{

char c;

c=getchar();

c=c>='A'&&c<='Z'?c+32:c;

putchar(c);

return 0;

}

3）测试

（a） 测试数据：

A a B b C c

（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：

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16-n位，即： x=x<<(16-n)

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



图1-2 编程题2的程序流程图

2）程序清单

#include<stdio.h>

int main()

{

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 (0 <= m && m <= 15 )

{

if (1 <= n && n <= 16 - n)

{

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

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

}

else

{

printf("n值超范围");

}

}

else

{

printf("m值超范围");

}

return 0;

}

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2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。

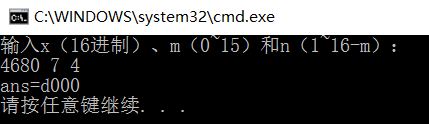


图1-2 编程题2的测试用例一的运行结果

对应测试测试用例2的运行结果如图1-3所示。

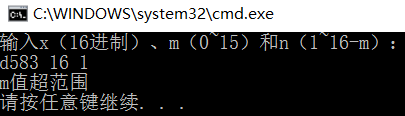


图1-3 编程题2的测试用例二的运行结果

对应测试测试用例3的运行结果如图1-4所示。

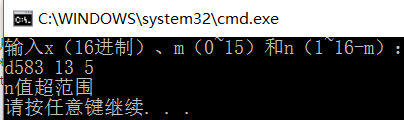


图1-4 编程题2的测试用例三的运行结果

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，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. 解题思路：



1. 程序清单

#include<stdio.h>

int main()

{

int N;

scanf("%d", &N);

char str\_ip\_index[4] = { '\0' };

unsigned long int IP;

unsigned long int IP1;

unsigned long int IP2;

unsigned long int IP3;

unsigned long int IP4;

for (int i = 1; i <= N; i++)

{

scanf("%lu", &IP);

IP1 = ((IP & 0xff000000) >> 24);

IP2 = ((IP & 0x00ff0000) >> 16);

IP3 = ((IP & 0x0000ff00) >> 8);

IP4 = ((IP & 0x000000ff));

printf("%lu.%lu.%lu.%lu\n", IP4, IP3, IP2, IP1);

}

return 0;

}

3）测试

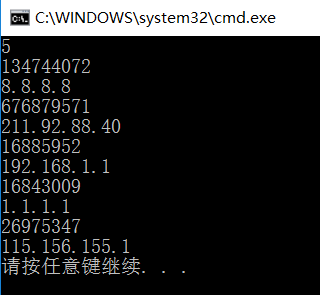
（a） 测试数据：

叙述选择测试数据的方法使数据覆盖尽量广泛 各节相同和不同如表1-2所示。

表1-2 编程题3的测试数据

|  |  |  |
| --- | --- | --- |
| 测试  用例 | 程 序 输 入 | 理 论 结 果 |
| X |
| 用例1 | 134744072 | 8.8.8.8 |
| 用例2 | 676879571 | 211.92.88.40 |
| 用例3 | 16885952 | 192.168.1.1 |
| 用例4 | 16843009 | 1.1.1.1 |
| 用例5 | 26975347 | 115.156.115.1 |

（b） 对应测试测试用例的运行结果如图1-2所示。



## 1.3 自设题

**（1）** 自设实验题目：编写一个程序, 对用户录入的产品信息进行格式化, 程序运行后需要有以下会话:

INPUT:

GroupNumber:1

Enter item number: 583

Enter unit price: 13.5

Enter purchase date(mm/dd/yy): 10/24/95

OUTPUT:

Item      Unit Price      Purchase Date

583        $ 13.50          10/24/95

其中, 数字项和日期项单位价格左对齐, 美元数量最大取值为9999.99

**（2）** 实验目的：通过设计实验程序，熟练标准输入输出，掌握输出表格的方法

**（3）**实验程序：

#include <stdio.h>

int main()

{

int GroupNumber=0;

int number[9] = {0};

float price[9] = { 0 };

int year[9] = { 0 };

int month[9] = { 0 };

int day[9] = { 0 };

printf("GroupNumber:");

scanf("%d", &GroupNumber) ;

for(int i=1;i<=GroupNumber;i++)

{

scanf("%d", &number[i]);

scanf("%f", &price[i]);

scanf("%d/%d/%d", &month[i], &day[i], &year[i]);

}

printf("Item \t Unit\t Purchase\n");

for (int i = 1; i <= GroupNumber; i++)

{

printf("%-9d$ %-9.2f%02d/%02d/%02d\n", number[i], price[i], month[i], day[i], year[i]);

}

getchar();

return 0;

}

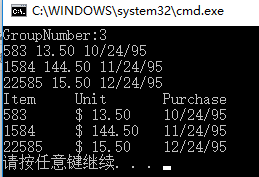
**（4）**实验用例：

3

583 13.50 10/24/95

1584 144.50 11/24/95

22585 15.50 12/24/95



**（5）**实验结论：

可以使用格式化输入输出制作表格，使打印清晰

## 1.4 实验小结

**（1）**对于可能工程中不太常见的输入输出格式不熟悉，通过练习得到了巩固。

**（2）**对于输入需要输出进行检测避免后续处理错误的输入数据导致DEBUG困难。

**（3）**学习使用测试用例和有效的测试方法。

**（4）**注重细节，培养习惯才能在工程代码中少出错。