

第 3 章

上机实验

实验 1 顺序结构程序设计

【实验目的】

1. 熟悉 C 编译系统集成环境及基本操作方法,掌握在该系统上如何编辑、编译、连接和运行一个 C 程序。
2. 掌握表达式的运算以及表达式语句的用法。
3. 掌握各种类型数据的输入、输出方法,正确使用各种格式说明。
4. 掌握顺序结构程序设计方法及程序调试方法。
5. 掌握数学函数的使用。

【实验内容】

1. 进入 C 编译系统集成环境,熟悉集成环境界面和有关菜单的使用方法。
2. 运行下面程序,并分析输出结果。
(1) 请将以下程序段补充完整。

```
char c;  
int a;  
long m;  
float f;  
double d;  
c = 'a'; //e1  
a = 61;  
m = 50000;  
f = 333.3456789;  
d = 123456789.123456789; //e2  
printf("c = %c\n a = %d\n m = %d\n", c, a, m);  
printf("f = %f\n d = %15.6f\n", f, d);
```

- (2) 编译、连接、运行此程序,记录并分析结果。

- (3) 完成以下修改。

① 请用 `/ * ... */` 将 e1 行至 e2 行之间的赋值语句注释起来,改用 `scanf` 函数输入数据,添加 `scanf` 函数如下:

```
scanf("%c %d %ld %f %lf", &c, &a, &m, &f, &d);
```

编译、连接、运行此程序。

数据按照如下方式输入：

a _61 _50000 _333.3456789 _123456789.123456789 ✓

记录并分析结果。

② 若要求输入数据之间用逗号分隔,请用//将上述 scanf 函数注释起来,重写 scanf 函数。

③ 删去输入项表列中的 &,运行程序,观察并分析运行结果。

(4) 请将步骤(3)scanf 函数中的 %ld 和 %lf 分别改为 %d 和 %f,运行程序,观察并分析运行结果。

(5) 思考。

对照程序和运行结果,回答如下问题:

① 若漏写 scanf 函数输入项表列中的取地址运算符 &,会出现什么情况?

② 举例说明调用 scanf 函数输入数据时,数据间应该怎样分隔。

3. 编写程序,上机编译、连接、运行程序并分析结果。

(1) 从键盘输入一个小写英文字母,将其转换为大写字母后,输出转换后的大写字母及其十进制的 ASCII 值。

(2) 从键盘输入圆的半径,计算并输出圆的周长、圆的面积和圆球的体积(保留 2 位小数)(提示:圆的周长 $l=2\pi r$,面积 $s=\pi r^2$,球体积 $v=\frac{4}{3}\pi r^3$)。

(3) 定义两个整数,通过键盘输入两个整数的值,计算它们的和、差、积、商,并输出。

实验 2 选择结构程序设计

【实验目的】

1. 利用 if 语句及 if 语句的嵌套实现选择结构程序设计。
2. 利用 switch 语句实现选择结构程序设计。
3. 能够编写选择结构的程序,解决实际问题。
4. 学会调试程序的方法。

【实验内容】

1. 运行实现以下分段函数的程序,并分析输出结果。

$$y = \begin{cases} x^4 + 1 & (0 < x < 5) \\ 0 & (x = 0) \\ |x| + 4 & \text{其他} \end{cases}$$

```
#include <stdio.h>
#include <math.h>
int main(void)
{
    int x,y;
    scanf("%d",&x);
```

```
if(0 < x < 5)
    y = pow(x, 4) + 1;
else if(x = 0);
    y = 0;
else
    y = fabs(x) + 4;
printf("x = %d, y = %d\n", x, y);
return 0;
}
```

(1) 对源代码进行编译,解读相关的编译信息;修改程序的语法错误与逻辑错误;分析数学表达式与 C 语言表达式之间的联系与区别。

(2) 分别测试当 x 为 2.2,0,-3.2 时程序的运行结果。

(3) 单步调试程序,分析 if 语句的执行过程。

2. 已知平面上三个点的坐标分别是(2,4)、(4,8)和(-1,-2),判断这三点是否共线。

3. 从键盘输入平面上三个点的坐标 A(x1,y1)、B(x2,y2)、C(x3,y3),检验它们能否构成三角形。若不能,则输出相应的信息;若能,则输出三角形的面积和周长。

(1) 程序分析:已知两点的坐标,可以使用两点间距离公式计算两点间距离。组成三角形的条件是三角形两边之和大于第三边。已知三条边的长度,可以根据海伦公式求面积。注意表达式的书写和分支条件的设定。

(2) 对源代码进行编译,若存在编译错误,根据提示修改源代码。运行测试用例数据,如表 3-1 所示,若结果有误,则设置断点,利用断点调试运行。

表 3-1 测试用例

序 号	输 入	输 出
1	点 A 坐标: 4,5 点 B 坐标: 6,9 点 C 坐标: 7,8	周长=10.13,面积=3.00
2	点 A 坐标: 4,6 点 B 坐标: 8,12 点 C 坐标: 12,18	这三个点不能组成三角形

4. 输入任意一个字母,作为中间字母,输出三个相邻的字母。假设字母 z 与字母 a 是相邻的。若输入 z,则输出 yza;若输入 a,则输出 zab;若输入 c,则输出 bcd。

5. 模拟两个人的输入(石头、剪刀、布),判断谁是胜利者。

实验 3 循环结构程序设计

【实验目的】

- 1. 熟练使用 for、while 和 do-while 语句实现循环程序设计。
- 2. 理解循环条件和循环体,以及 for、while 和 do-while 语句的相同及不同之处。
- 3. 掌握循环嵌套结构的使用;掌握分支结构、循环结构的综合应用。

4. 掌握 break 和 continue 语句的使用。

【实验内容】

1. 编程求 $\sum_{i=1}^n i!$ 的值。

(1) 用二重循环实现求和。

(2) 用一重循环实现求和。

(3) 分析结果：

① 输入一个较小的正整数,分析程序执行结果。

② 输入 0 或一个负数,分析程序执行结果。

③ 输入一个较大的正整数,分析程序执行结果。思考:存放阶乘、阶乘和的变量应该用什么类型的数据?

④ 修改程序,当输入数据非法时,给出一个恰当提示。

2. 找出 1000 以内的所有“完数”。“完数”就是这样一个整数,它恰好等于它的因子之和。例如,6 的因子为 1、2、3,而 $6=1+2+3$,则 6 是完数。

```
#include <stdio.h>
int main(void)
{
    int n,m,k;
    for(n=1;n<=1000;n++)
    {
        k=0;
        for(m=1;m<n;m++)
            if(n%m==0)
                ①
        if(n==k)
            printf("%d ",n);
    }
    return 0;
}
```

(1) 请填空,使程序完整,分析其执行过程。

(2) 还能再修改“for(m=1;m<n;m++)”循环条件以减少循环次数吗?

(3) 思考:语句“k=0;”能在变量赋初值部分给出吗(多重循环要注意变量赋初值的位置)?

(4) 若不仅输出完数,还输出其各个因子,如输出“6 的因子有 1、2、3”,如何修改程序?

3. 思考以下程序的运行结果,并上机验证,体会 break 与 continue 的区别。

```
#include <stdio.h>
int main(void)
{
    int n,m;
    for(n=1;n<=10;n++)
    {
```

```

        if(n == 5)
            break;
        printf(" %d",n);
    }
    printf("\n");

    for(m = 1;m <= 10;m++)
    {
        if(m == 5)
            continue;
        printf(" %d",m);
    }
    return 0;
}

```

4. 给定 1、2、3、4 四个数字,请输出所有由它们组成的无重复数字的三位数。输出整数的顺序要求从小到大排列,每行六个整数。

问题分析: 这个问题可以通过穷举每一位和每个允许的数字的排列来解决。可以采用三重循环,每重循环对应三位数中的一位,每重循环循环四次(1、2、3、4)。

5. 用公式 $\pi/4 \approx 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots + (-1)^{n-1} \frac{1}{2n-1}$ 求 π 的值。当最后一项的绝对值小于 10^{-4} 时可认为达到了精度要求。

实验 4 数 组

【实验目的】

1. 掌握一维数组的定义及使用。
2. 掌握二维数组的定义及使用。
3. 掌握字符数组的定义及使用。

【实验内容】

1. 运行以下程序:

```

#include <stdio.h>
int main(void)
{
    char s[20];
    gets(s);
    puts(s);
    return 0;
}

```

- (1) 从键盘输入“How are you”,观察输出结果。

(2) 将 gets(s) 改为 scanf("%s",s),重新运行程序,观察输出结果有什么不同,分析原因。

(3) 如果用%c 格式符输入/输出字符串,程序该如何修改?

2. 以下程序的功能是从键盘输入十个数,检查3 是否包含在这些数中,若包含,则输出第一个3 出现的位置。检验程序是否正确,若有错误,请找出并改正。

```
#include <stdio.h>
int main(void)
{
    int a[10];
    i = 0;
    while(i < 10)
        scanf("%d", a[i]);
        i++;
    for(i = 0; i < 10; i++)
        if(a[i] == 3)
            printf("%d\n", i);
            continue;
    if(i == 10)
        printf("not found!\n");
    return 0;
}
```

3. 以下程序的功能是将字符串中的数字字符删除。检验程序是否正确,若有错误,请找出并改正。

```
#include <stdio.h>
int main(void)
{
    char a[10];
    int i, j;
    gets(a);
    for(i = 0, j = 0; a[i] != '\0'; i++, j++)
        if(a[i] < 0 || a[i] > 9)
            a[j] = a[i];
    puts(a);
    return 0;
}
```

4. 编写程序,将某数插入到一维数组中某处,如将8 插入到数组2,3,5,0,9,6 的第3 个位置,插入后数组为2,3,8,5,0,9,6。要插入的数及其插入位置由键盘输入。

5. 编写程序,将4 行、5 列的二维数组的第0 行与第2 行对调。

6. 输入元素个数 n ($n \leq 10$),再输入 n 个整数,存放在一个数组中,计算这 n 个数的平均数和中位数。中位数的计算方法是:把数据按大小排序,如果数据个数是奇数,则中间那个数就是中位数;如果数据个数是偶数,则中间两个数的平均值就是中位数。

实验5 函 数

【实验目的】

1. 掌握函数的定义方法。

2. 掌握函数的调用方法。
3. 掌握函数的实参与形参的对应关系。

【实验内容】

1. 写出以下程序的运行结果,然后上机验证。

(1)

```
#include <stdio.h>
int x = 20;
void f1(int x)
{
    x += 10;
    printf("f1: %d\n", x);
}
void f2()
{
    x += 10;
    printf("f2: %d\n", x);
}
int main(void)
{
    int x = 10;
    f1(x);
    f2();
    printf("main: %d\n", x);
    return 0;
}
```

(2)

```
#include <stdio.h>
int fun(int a)
{
    int b = 0;
    static int c = 0;
    b++;
    c++;
    return a + b + c;
}
int main(void)
{
    int i;
    for(i = 0; i < 3; i++)
        printf("%d\n", fun(i));
    return 0;
}
```

思考: 如果将程序中的 static 删除,运行结果会有什么不同? 为什么?

2. 下面 add 函数的功能是求两个参数之和并将其作为函数返回值。请检验函数是否

正确,若有错误,请改正并将主函数补充完整。

```
#include <stdio.h>
void add(int a, int b);
{
    int c;
    c = a + b;
}
int main(void)
{
    int x, y, z;
    scanf(" %d %d", &x, &y);
    z = ①; //调用函数, 计算 x、y 之和
    printf(②); //输出 x、y 之和
    return 0;
}
```

3. 下面程序的功能是统计一维数组中负数的个数。检验程序是否正确,若有错误,请找出并改正。

```
#include <stdio.h>
int main(void)
{
    int x[6], i, y;
    for(i = 0; i < 6; i++) scanf(" %d", &x[i]);
    sum(int x[], 6);
    printf(" %d\n", y);
    return 0;
}
void sum(int a[], int n)
{
    int i, m;
    for(i = 0; i < n; i++)
        if(a[i] < 0) m++;
    return m;
}
```

4. 编写一个函数,功能是求两个正整数的最大公约数。数据的输入与输出在主函数中完成。用递归与非递归两种方法完成。

5. 素数是只能被 1 和自身整除的正整数。试实现一个判断素数的函数,并利用该函数验证哥德巴赫猜想,即任何一个大于 6 的偶数均可表示为两个奇素数之和。

6. 用一维数组作函数参数,实现学生成绩管理系统。

- (1) 输入每个学生的学号和一门课的成绩。
- (2) 计算平均成绩及高于平均分的人数。
- (3) 按成绩由高到低排出名次表。
- (4) 按学号查询学生成绩。
- (5) 输出每个学生的学号、成绩。

实验 6 指 针

128

【实验目的】

1. 掌握指针的概念、定义方法及基本操作。
2. 掌握通过指针操作数组元素的方法。
3. 掌握通过指针操作字符串的方法。
4. 掌握指针数组和指向指针的指针的概念、定义方法及使用方法。

【实验内容】

1. 程序改错。

```
#include <stdio.h>
int main(void)
{
    float *p;
    scanf("%f", p);
    printf("%f\n", *p);
    return 0;
}
```

运行该程序,找出程序出错原因。改正错误,运行程序,得到正确结果。

2. 输出数组 a 的 10 个元素。以下程序正确吗?若有错,请改正并上机验证。

```
#include <stdio.h>
int main(void)
{
    int a[10] = {1,2,3,4,5,6,7,8,9,10}, i;
    for(i = 0; i < 10; i++)
    {
        printf("%d ", *a);
        a++;
    }
    return 0;
}
```

3. 分析以下程序运行后的输出结果,并上机验证。

```
#include <stdio.h>
int main(void)
{
    int a[] = {1,2,3}, *p = a, b;
    char *q = "abcde";
    b = *++p;
    printf("\n%d", b);
    printf("\n%d %d %d %d", *a, * (a+2), * (p+1), p[1]);
    printf("\n%d %c %s %s", *q, q[3], q+3, q);
    return 0;
}
```

4. 输入 3 个整数,按由小到大的顺序输出。请补充并且修改程序,运行该程序,得到正确的结果。

```
#include <stdio.h>
int main(void)
{
    int a,b,c;
    int *pa = &a, *pb = &b, *pc = &c;
    printf("input 3 integer a,b,c:");
    scanf("%d, %d, %d", &a, &b, &c);           //e1
    if (a > b)                                   //e2
    {pt = pa; pa = pb; pb = pt;}
    if (*pa > *pc){ ① ; pa = pc; pc = pt;}
    if (*pb > *pc){pt = pb; ② ; pc = pt;}
    printf("%d, %d, %d\n", a, b, c);           //e3
    return 0;
}
```

解决以下问题:

- (1) 程序 e1 行的输入语句是否可以改为“scanf(“%d,%d,%d”,pa,pb,pc);”?
 - (2) 程序 e2 行的 if 语句中的条件是否可以改为“*pa>*pb”?
 - (3) 程序 e3 行的输出语句是否正确? 如果不正确应如何修改?
 - (4) 仿照上面程序,编写输入 3 个字符串并按由小到大的顺序输出的程序。
 - (5) 比较分析这两个程序,用指针处理整数与字符串有什么不同? 例如:
 - ① 如何得到指向整数或字符串的指针?
 - ② 如何比较两个整数或字符串的大小?
 - ③ 如何交换两个整数或字符串?
5. 输入 10 个数,按与输入时相反的顺序将其输出。

实验 7 指针与函数

【实验目的】

1. 掌握指针变量作函数参数的使用方法。
2. 掌握返回值为指针的函数的概念、定义方法及使用方法。
3. 掌握指向函数的指针的概念、定义方法及使用方法。
4. 掌握指针数组作 main() 函数参数的使用方法。

【实验内容】

1. 输入 3 个整数,按由小到大的顺序输出。请补充并运行以下程序,得到正确的结果。

```
#include <stdio.h>
int main(void)
{
    int a,b,c;
```