

程序设计（一）实验指导书

实验一 基础

1) 实验目的:

- (1) 了解 C 程序的构成及其编写、调试、运行的基本过程，熟悉 VC 编程环境；
- (2) 理解 C 语言中各种数据类型的适用范围、空间大小及其对程序设计的作用；
- (3) 熟练掌握输入输出函数的调用；
- (4) 学会使用赋值表达式和赋值语句。

2) 实验内容:

C 程序的构成及调试步骤；整型数据运算及溢出问题；浮点型数据运算及精度问题；赋值运算符和表达式；输入输出函数调用练习。

3) 实验要求:

根据下述实验题目，编写程序，并编译、链接、运行，实验报告要求提交源程序文件、运行结果截图，并回答相应的问题。

4) 实验题目

- (1) 运行以下程序，并解释运行结果。

```
#include <stdio.h>
int main(void)
{   int x = -1;
    unsigned u = 2147483648;
    printf("x = %u = %d\n", x, x);
    printf("u = %u = %d\n", u, u);
    return 0;
}
```

- (2) 运行以下程序，解释运行结果，并修改程序以使结果正确。

```
#include <stdio.h>
int main(void)
{   int x = 2147483647;
    printf("x + 1 = %d\n", x + 1);
}
```

```

        return 0;
    }

```

(3) 运行以下程序，解释运行结果。

程序 1:

```

#include <stdio.h>
int main(void)
{
    float a;
        double b;
        a = 123456.789e4;
        b = 123456.789e4;
        printf("%f\n%f\n",a,b);
        return 0;
}

```

程序 2:

```

# include <stdio.h>
int main()
{
    float a,b;
        a=123456.789e5;
        b=a+20;
        printf("%f\n",a);
        printf("%f\n",b);
        return 0;
}

```

(4) 求平方根：输入1 个实数x，计算并输出其平方根（保留1 位小数）。

(5) 华氏温度转换为摄氏温度：输入华氏温度 f，计算并输出相应的摄氏温度 c (保留 2 位小数)。 $c = 5/9(f-32)$ 。

(6) 大写字母转换成小写字母：输入一个大写英文字母，输出相应的小写字母。

实验二 选择

1 找最小值

输入四个整数，输出其中的最小值。

2 求三角形的面积和周长

输入三角形的三条边a, b, c, 如果能构成一个三角形, 输出面积area 和周长perimeter(保留2 位小数); 否则, 输出 “These sides do not correspond to a valid triangle”。在一个三角形中, 任意两边之和大于第三边。三角形的面积计算公式:

$$\text{area} * 2 = s(s-a)(s-b)(s-c) \quad \text{其中: } s = (a+b+c)/2$$

3 判断数的符号

输入整数x, 若x 大于0, y=1; 若x 等于0, y=0; 否则, y=-1, 最后输出y。

4 字母转换

输入一个字符, 如果它是大写字母, 输出相应的小写字母; 如果它是小写字母, 输出相应的大写字母; 否则, 原样输出。

实验三 循环

1 求n!

读入1 个正整数 n ($n \leq 20$), 计算并输出 n! 。

2 求 $1 - 1/3 + 1/5 - 1/7 + \dots$

读入1 个正实数eps, 计算并输出 $1 - 1/3 + 1/5 - 1/7 + \dots$, 直到最后一项的绝对值小于eps(保留6 位小数)。

3 求 $a + aa + aaa + \dots + a$

输入2 个正整数a 和n, 求 $a + aa + aaa + \dots + a$ (n 个a)之和。

4 求各位数字的立方和等于它本身的数

输入2 个正整数m 和n ($1 \leq m, n \leq 1000$), 输出m 到n 之间的所有满足各位数字的立方和等于它本身的数。

5 统计素数并求和

输入2 个正整数m 和n ($1 \leq m, n \leq 500$)，统计并输出m 和n 之间的素数的个数以及这些素数的和。

6 统计单词

输入一行字符，统计其中单词的个数。各单词之间用空格分隔，空格数可以是多个。

7 输出一个整数的各位数字

，从高位开始逐位输出它的各位数字。

实验四 数组

一.实验目的

- (1) 熟练掌握一维数组、二维数组的定义、初始化，以及输入、输出的方法；
- (2) 熟练掌握字符数组和字符串函数的使用；
- (3) 掌握与数组有关的常用算法。

二. 实验指导

1. 对任意一个一维数组，从中找出数组元素的最大值和最小值并输出。

要求：

- 1) 数组有 10 个元素；
- 2) 使用 scanf 函数实现数组元素的输入，输入前给出必要的提示信息；
- 3) 输出时，首先输出数组的十个元素，然后输出其最大值和最小值。

(一)算法分析

首先定义一个数组，通过循环从键盘输入十个数组元素的值；再使用循环，通过将数组每个元素与设定的 min, max 进行比较，求出数组元素的最大值和最小值。

(三)实验目的

- 1) 掌握数组定义的方法；
- 2) 掌握数组的输入和输出方法；
- 3) 掌握访问数组元素的方法。

2. 三个学生、五门课程的成绩分别存放在 4×6 矩阵的前 3×5 列，计算出每个学生的平均成绩存放在该数组的最后一列的对应行上。计算单科的成绩总和存放在最后一行的对应列

上，并输出单科成绩的平均分。

要求：

- (1) 数组类型为实型，输入成绩由scanf函数实现；
- (2) 输出已输入的原始成绩数据（3×5）；
- (3) 计算每个学生的平均成绩以及单科的成绩总和并按要求填入到数组中，输出填入结果后的数组（4×6）；
- (4) 计算单科平均成绩后输出。

(一)算法分析

首先定义二维数组，用以存放 4×6 矩阵。对二维数组每行数据进行累加和，再求平均，放在数组最后一列的对应行上，并输出每个学生的平均成绩。对二维数组每列数据进行累加和，再求平均，存放在最后一行的对应列上，并输出每科成绩的平均分。

(四)实验目的

- 1) 掌握二维数组的定义及输入、输出的方法；
- 2) 掌握二维数组应用问题程序的编写方法

3. 输入一个大于 2 且小于 10 的整数 n，定义一个二维整型数组（n×n），初始化该数组，将数组中最大元素所在的行和最小元素所在的行对调。

要求：

- 1) n×n 数组元素的值由 scanf 函数从键盘输入，然后输出该数组；
- 2) 查找最大值最小值所在的行（假定最大值、最小值不在同一行上）；
- 3) 将数组中最大元素所在的行和最小元素所在的行对调，并输出对调后的数组；
- 4) 数组按 n 行 n 列的方式输出。

(一)算法分析

通过将数组中的每个元素与设定的max,min比较大小，找到最大元素所在的行和最小元素所在的行，然后将两行的元素进行对调。

(四)实验目的

- 1) 掌握二维数组存储矩阵的方法；
- 2) 掌握有关二维数组比较复杂算法的编写方法。

5. 在给定的字符串中查找指定的字符。

要求：

- 1) 字符串采用直接初始化的方式处理；
- 2) 通过scanf函数读入一个任意字符；
- 3) 在字符串中查找该字符，如果存在该字符，输出该字符在字符串中出现的位置，如果再给定的字符串中不存在该字符，则给出相应的说明信息。

(一)算法分析

对给定的字符，在字符数组中各元素依次比较，如果相同则找到，否则，在碰到字符串结束标志时返回，说明字符串中不存在该字符。

(四)实验目的

- 1) 掌握字符数组的使用；
- 2) 掌握字符串的使用方法。

实验五 函数

一、实验目的

1. 掌握函数定义、调用和声明的方法；
2. 掌握实参和形参之间的传递方式；
3. 掌握函数的嵌套调用和递归调用的方法；
4. 掌握全局变量和局部变量、动态变量、静态变量的概念和使用方法；

二、实验指导

实验 1. 计算多项式 $1-x+x^2/2!-x^3/3!+\dots$ 的和直到末项的绝对值小于 ϵ 为止。

要求：

- 1) 编写 x^n 的函数。
- 2) 写 $n!$ 的函数。

(一) 算法分析

采用循环语句，进行累乘可以实现 $n!$ 函数；采用循环， x 乘 n 次，可以实现 x^n 函数。

(四) 实验目的

- 1) 掌握函数形式参数和实际参数的使用；
- 2) 掌握函数的定义和调用。

实验 2. 输入二个整数 $m, n (1 < m < n)$, 统计区间内素数的个数。

要求：

- 1) 整数 m, n 由输入获得；
- 2) 编写判断素数的函数 `int Prime(int n) { }`：如果 n 是非素数，返回 0， n 是素数，返回 1。

(一) 算法分析

对整数 n ，循环测试从 2 到 m 的各整数，如果 n 能被其中一个整除，则 n 是非素数；如果没有整除，则 n 是素数。

(四) 实验目的

- 1) 掌握函数体内可以有多个返回语句，但函数的返回值只能有一个。
- 2) 掌握较为复杂算法的编写；

实验 3. 求两个整数的最大公约数和最小公倍数。

要求：

- 1) 不用全局变量；
- 2) 分别用两个函数求最大公约数和最小公倍数。两整数在主函数中输入，传递给函数 1，以求出最大公约数，然后再与两个整数一起作为实参传给函数 2，以求出最小公倍数，返回主函数输出最大公约数和最小公倍数。

(一) 算法分析

求两个整数的最大公约数可以从两者较小的一个开始依次减 1 进行测试，看是否被两个整数整除，如果能，则为最大公约数；不能，最大公约数为 1。求最小公倍数采用两个整数的乘积除以两个整数的最大公约数得到。

(四) 实验目的

- 1) 掌握比较复杂的算法；
- 2) 掌握函数嵌套调用的方法。

实验 4. 打印 1 到 5 的阶乘值。

要求：采用静态变量实现。

(一) 算法分析

根据题意，采用静态变量保存 $(n-1)!$ ，在计算 $n!$ 时只需在原来计算结果的基础上乘以 n 即可。

(四) 实验目的

- 1) 掌握静态变量的用法；
- 2) 掌握静态变量程序的编写。

实验 5. 用递归法将一个整数 n 转换成字符串。

要求：

- 1) 整数转换成字符串功能写成一个函数；
- 2) 在主函数中调用整数转换字符串功能。

(一) 算法分析

总体思路是将整数拆分成多个数字，在拆的时候考虑输出顺序。一般而言，拆分整数是从低位到高位拆，要使输出的字符串与整数顺序保持一致，则要使先拆除的数字后输出。

(四) 实验目的：

- 1) 掌握递归程序的运行机制；
- 2) 掌握递归程序的编写方法。

4. 编写函数 `cmpStr(char s1[],char s2[])` 比较两个字符串 $s1$ 和 $s2$ 的大小。如果 $s1>s2$ ，返回一个正数； $s1=s2$ ，返回 0； $s1<s2$ ，返回一个负数。在主函数中调用该函数完成字符串的比较操作，并输出返回的值。

要求：

- 1) 不能使用 `strcmp()` 函数；
- 2) 用 `gets()` 函数在主函数中输入两个字符串；
- 3) 返回的正数或负数的绝对值是进行比较的两个字符串中相应字符的 ASCII 码的差值。

(一) 算法分析

两个字符串从第一个字符开始一一对应进行比较，直到不相等的那个字符，从而求得它们的 ASCII 码的差值。

(四) 实验目的

- 1)掌握把数组作为函数参数传递进行函数调用的方法;
- 2)掌握访问和操作字符数组元素的算法;
- 3)掌握与字符数组有关的比较算法。

实验六 指针

7.1 查找整数

定义一个函数 `search(int *list, int n, int x)`，在数组`list`中查找元素`x`，若找到则返回相应下标，否则返回-1。

7.2

将一个长度不超过4的数字字符串转换成一个整数。如字符串是“3248”，则转换的整数是3248。

7.3 删除字符

编写一个函数`delchar(char *s, char c)`，该函数将字符串`s`中出现的所有`c`字符删除。自己定义`main`函数，并在其中调用`delchar(s, c)`函数。

7.4编写子函数：`insert(char *s1, char *s2, int f)`，其功能是在字符串 `s1` 中的指定位置第 `f` 个字符之后插入整个字符串 `s2`形成一个新的字符串`s1`。在主函数中完成从键盘输入`s1,s2`以及`f`的值，并在调用`insert`函数时传递给`s1,s2`以及`f`，并输出完成插入后的新字符串。（自编`main`函数验证）

7.5 .编写子函数`fun (int *n)`，`fun`的功能是：将形参`n`所指变量中，各位上为偶数的数去除,剩余的数按原来从高位到低位的顺序组成一个新的数，并通过形参指针`n`传回所指变量。（例如，输入一个数：27698438，新的数：为739）。

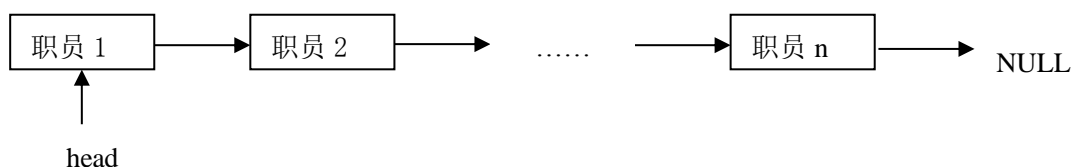
实验七 结构体和链表

8.1

已知一个结构体定义如下：

```
typedef struct{
    long id;           //工号
    char name[20];     //姓名
    float salary;      //薪水
    struct stud *next;
}EMP;
```

假定已建立了一个按 salary 值升序排列的链表（无专用头结点），要求编写子函数 EMP *InsertNode(EMP *head)，实现新结点插入功能，新结点插入后链表的 salary 值仍然保持升序，该函数应包含结点所需内存的申请以及职员信息的录入功能。链表如下图所示。（只写此子函数，不用写主函数）



8.2 成绩处理

有两个单向链表，头指针分别为list1、list2，链表中每一结点包含姓名、工资基本信息，请编一函数，把两个链表拼组成一个链表，并返回拼组后的新链表。

8.3 链表排序（只能改指针指向，不能交换结点）

有一个已经存在的无专用头结点的单向链表，链表结点用如下结构体表示：

```
struct stud{
    long id; //学号
    char name[20]; //姓名
    float score; 成绩
    struct stud *next; //下一个结点的地址
};
```

编一子函数 struct stud *backlist(struct stud * head)，实现将此链表逆序排列，即：将链头当链尾，链尾当链头。初始链表如下图所示。（只写此子函数，不用写主函数）



实验八 文件

9.1 统计字母、数字和字符

统计一个文本文件中字母、数字及其他字符各有多少个。

9.2 删除注释

将一个C 语言源程序文件中所有注释去掉后，存入另一个文件。

9.3 文件合并

文本文件a1.txt 和a2.txt 中包含若干从小到大排过序的整数，现要求把两个文件中的数据合起来，仍按从小到大顺序写入文件a3.txt 中，试编写相应程序。

9.4

已知一个结构体定义如下：

```
typedef struct {  
    long id;  
    char name[20];  
    float score;  
} STUD;
```

有一个二进制文件 stu1.dat 存放有若干个这种 STUD 类型的数据。编写一个完整的程序，将该文件中所有 score 小于 60 的那些记录写到 stu2.dat 文件中，并统计写入 stu2.dat 中的记录个数。