

《程序设计语言实习》指导书

注意事项:

1. 修改《程序设计语言实习》模板封面上的“学号”、“姓名”和“指导老师”。
2. 修改正文区域的“页眉”部分，修改“学号”和“姓名”；中文为宋体，英文为 Times New Roman，字号为五号；单倍行距，“段前”、“段后”设为 0；“运行结果”可以文字输入，也可以截图；**注意排版，图片不要过大**，保证清晰度即可。
3. 根据不同模块要求任意选择题目。
4. 代码中注释率不得低于 30%。
5. 根据题目要求画流程图（需要画流程图的题目已标注）。流程图可以用 Visio, Axure 等工具绘画，或者使用 PPT 的画图工具，不推荐直接使用 Word 画图。
6. 评分标准：
 - 1) 面试得分占 70%，平时分占 30%，根据平时上机考勤。注重平时上机，老师会不定期检查学生进度，学生不得以自己有私人电脑为借口而不来上机。
 - 2) 程序能按照题目要求正确编写，编译通过并正确运行（若以各种理由降低要求，将不予给予分数）。
 - 3) 学生能对自己的程序面对教师提问并能熟练地解释清楚。
 - 4) 提交①电子稿，以“学号姓名”命名文件夹，包括《程序设计语言实习》电子稿（.doc）和题目的源文件（.c，学号+题目编号）（如：130801105.doc，13080110507.c，13080110524.c）。②《程序设计语言实习》打印稿。学委统一收齐交指导教师。以上几项缺一不可，否则不能到得相应学分。

模块一：选做 2 个题目（画流程图）：

1. **统计各年龄段人数**。N 个年龄通过调用随机函数获取，编写函数把 0~9 岁年龄段的人数放入数组下标[0]中，把 10~19 岁年龄段的人数放入数组下标[1]中，以此类推，把 100~120 岁年龄段的人数放入数组下标[10]中。
2. **求素数**。编写一个函数，将大于整数 m 且紧靠 m 的 k 个素数存入数组中，在命令行输入 m 和 k。如输入 17 和 5，输出 19, 23, 29, 31, 37。
3. **求最大公约数和最小公倍数**。输入两个正整数 m 和 n，求其最大公约数和最小公倍数。
4. **四则运算**。帮助某小学数学老师出试卷。随机生成 2 个 800 以内的正整数或零，随机计算+、-、*或/，答案也必须是 800 以内的**正整数**或零。共出 50 道题目，不能出现重复题目（2+3 和 3+2 算不同的题目），并将题目写入文件。
5. **算术运算**。输入一个算术表达式字符串，得到最后结果。算术运算符包括+、-、*、/，注意小括号可改变优先级。例如输入“(2+3)*4”得到 20。
6. **玫瑰花数**。如果一个四位数等于它的每一位数的 4 次方之和，则称为玫瑰花数，比如： $1634=1^4+6^4+3^4+4^4$ ，编程输出所有的玫瑰花数。
7. **借书组合问题**。张三有五本新书，要分别借给 A、B 和 C 三位同学，如果每人每次只能借一本，则可以有多少种不同的借法，并输出所有借法。
8. **猜数字游戏**。游戏随机生成 1 个各位互不相同的四位数，接受用户输入 1 个各位互不相同的四位数，根据用户输入输出“mAnB”的结果。其中 A 表示数字和位置都正确，B 表示数字正确，位置不正确。1 个数字最多可以猜 10 次，如果 10 次全猜错给出正确答案。例如随机数为“1234”，用户输入为“1435”，显示“2A1B”。
9. **完数问题**。一个数如果恰好等于它的因子之和，这个数就称为“完数”。例如 $6=1+2+3$ 。编程找出 1000 以内的所有完数。
10. **进制转换**。输入一个十进制数 N，将它转换成 R 进制数（ $2 \leq R \leq 16$, $R \neq 10$ ）。
11. **第二小整数**。求 n 个整数中倒数第二小的数。每一个整数都独立看成一个数，如有三个数分别是 1, 1, 3，那么，第二小的数就是 1。
12. **取球问题**。口袋里有红、黄、蓝、白、黑 5 种颜色的球若干个。每次从口袋中取出 3 个不同颜色的球，问有多少种取法。

模块二：选做 2 个题目（画流程图）：

13. 编写合并整数数组的函数。数组 $a[M]$ 中有 m 个元素 ($m < M$)，数组 $b[N]$ 中有 n 个元素 ($n < N$)，且 $m+n \leq M$ ，将 a 、 b 两个数组按存放的整数升序排序并合并放入数组 a 中。
14. 两个一元多项式相加。输入 2 个多项式，**例如** $A_{17}(x)=7+3x+9x^8+5x^{17}$ 和 $B_8(x)=8x+22x^7-9x^8$ ，得到 $C_{17}(x)=7+11x+22x^7+5x^{17}$ 。
15. 数组插入问题。生成一个 10 元素构成的一维数组，数组元素由用户随机输入。要求：先按照升序排列并输出。再输入一个数，按照升序的规律将其插入并输出。
16. 转方阵。对一个方阵转置，就是把原来的行号变列号，原来的列号变行号。例如，图 1 的方阵转置后变为图 2；但如果是对该方阵顺时针旋转（不是转置），却是如图 3。请编写**一个函数**，实现一个方阵顺时针旋转。注：方阵是一个 $N \times N$ 的矩阵。

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

图 1

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

图 2

13	9	5	1
14	10	6	2
15	11	7	3
16	12	8	4

图 3

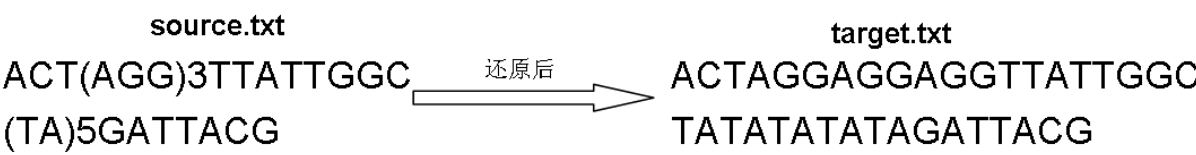
17. 链表操作。建立单向 int 链表，连续输入 5 个结点创建链表，并实现在原链表中插入数字、删除数字、查找数字的功能。
18. 链表合并。编写创建链表函数，链表中的结点包括学号 (int) 和成绩 (int)。创建 a 、 b 两个链表后，再将这两个链表合并，按学号升序排序（设学号都不相同）。

模块三：选做 2 个题目（画流程图）：

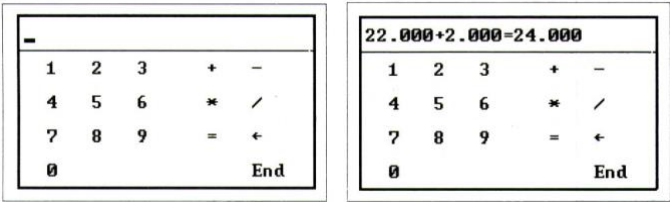
19. **统计频率。**读入一个文件，文件中包含字母，数字，空格，标点符号等。请统计文件中的字母，数字，空格和其他符号的数目，在屏幕上显示。
20. **字符串次数问题。**编写函数统计给定字符串在文中出现的次数，如文件中内容为 asd asasdfg asd as zx67 asd mklo，给定字符串为 as，则函数返回 6。文件名和给定字符串由命令行输入得到。
21. **统计字母个数。**读取文件中的字符串，统计从“a”到“z”26 个字母各自出现的次数，并将结果放入数组中。如文件中有字符串 abcdefgabcdeabc，输出 33322110000000000000000000。
22. **统计单词。**要求用**命令行**读取一个英文文章（比如：文件 newspaper.txt），文章内有数字、空格、标点符号等。请统计出该文章中单词的个数，并按字典顺序排序存入文件新文件中。注意单词只有全部由字母组成才被计数。
23. **单词复数。**某单词本（文件）中有若干行单词单数形式，请在该单词后面给出单词复数形式，即单词本中每一行是“单数 复数”形式。注：如果单词以 s、x、z、ch、或 sh 结尾，单词加 es；如果单词以 y 结尾，并且 y 前面是一个辅音，将 y 改为 ies；对其他单词，后面加 s。
24. **猜首都。**先自己书写一个 answer.txt 文件，在其中存放 10 个国家及其首都。从该文件中随机读取出一个国家，让玩家回答出该国家的首都是什么。如回答正确则给予鼓励（如“Good Job!”），并提示是否继续。如连续 3 次都回答错误，则公布正确答案，并提示是否继续。回答完 5 道题目，给予鼓励并退出游戏。玩家输入首都时的单词不考虑大小写、空格的问题（如答案是“Beijing”，玩家输入“Bei Jing”算正确答案）。
25. **删除字符。**从源文件中读取字符，编写函数将指定字符从文件中删除，结果写入目标文件中，文件中的字母不区分大小写。源文件文件名、目标文件文件名、指定字符均从命令行得到。如源文件中内容为 source.txt，指定字符为 n，目标文件中内容为 output.txt。注意需要用**命令行**输入参数。
26. **字符串压缩。**利用字符重复出现的次数，编写一种方法，实现基本的字符串压缩功能。比如，字符串 aabccccaaa 会变为 a2b1c5a3。若“压缩”后的字符串没有变短，则返回原先的字符串。可以假设字符串中只包含大小写英文字母（a 至 z）。

模块四：选做 1 个题目：

27. 分数比较问题。比较两个分数的大小。
28. 分数加减问题。输入 “a/boc/d”，其中 a、b、c、d 是一个 0-9 的整数。o 是运算符 “+” 或者 “-”。输出对于输入数据的每一行输出两个分数的运算结果。注意结果应符合书写习惯，没有多余的符号、分子、分母，并且化简至最简分数。
29. 年月日搜索问题。中国俗话说 “三天打渔两天晒网”。张三从 2000 年 1 月 1 号起开始 “三天打渔两天晒网”，请问张三在以后的某一天是 “打渔” 还是 “晒网”。
30. DNA 序列还原。某生物学文件内容为多条长度不等 DNA 序列，为书写方便其中有些片段用小括号括起并跟一个 int 数字，表示该片段需要重复多次。如 “(AGG)3” 表示 “AGG” 片段需要重复 3 次，请按要求将源文件还原后写入新的文件中。**注意：括号内的 DNA 片段长度不定长，重复次数可能是个多位数。**



31. 算术计算器。程序首先显示计算器界面，并等待用户输入一个 2 个操作数的算术计算式（见图 11 所示）。假如用户输入 22+2<CR>，程序将在计算器的显示屏上输出计算结果（见图 12 所示）。如果输入错误，显示 input error!。当用户输入一个算式后，计算器显示计算结果，同时屏幕顶部显示：Continue? 如果用户此时按 Y 键，则继续输入下一个算式，否则，程序终止。
- 提示：在屏幕的指定位置输出利用 gotoxy() 和 cprintf 函数实现。包含在 conio.h 头文件中。用户输入 “y” 或 “Y” 可用 getch 函数接收。边框线可以用 ASCII 中的 “制表线” 绘制。



模块五：选做 1 个题目：

32. 饭卡管理程序。

- (1). 建立饭卡信息：添加若干人的饭卡号、姓名、金额，要求饭卡号是唯一的；
- (2). 买饭：要求用户输入饭卡号、饭费，系统自动从该人的饭卡中减去饭钱，并分别显示买饭前后的金额，如果原来饭卡中的余额不足 5 元，则不能买饭，显示“余额不足，请充值”；
- (3). 充值：输入饭卡号、充值金额，充值完成后显示充值前后的金额。

33. 歌曲排行榜管理系统。

- (1). 每首歌曲包括编号、歌曲名称、歌手姓名、点播次数（如 001，告白气球，周杰伦， 123）；
- (2). 要求提供输入歌曲信息、删除歌曲信息、修改歌曲信息。（注删除、修改可按照歌曲名称查找）。
- (3). 程序要求有菜单，根据需求选择功能。

34. 学生成绩管理系统。

- (1). 输入 n 个学生的学号，姓名，高等数学、大学物理、英语和 C 语言四门课程的成绩。
- (2). 计算每门课平均分数并输出；
- (3). 按照某类课程的成绩高低排序；
- (4). 找出每门课最高分的同学学号和姓名（如果最高分有多名同学，都输出）。

35. 排考场座位。

假设考场有 8×8 的座位，本程序可自动对考生进行座位安排，每当一个考生进入考场就为他安排座位（要求用链表实现）。要求

- (1). 设定考场座位：用户输入准考证号和姓名，然后系统随机产生该考生座位的行号和列号，要求做到一个考生只有一个座位，且在已有考生的座位上不能再安排新的考生；
- (2). 取消考场座位；
- (3). 显示某座位考生信息：输入座位的行号和列号，显示该座位学生的信息
- (4). 查找学生座位：输入某考生准考证号，显示该学生的座位；
- (5). 显示考场座次表：在每个座位对应的行列上显示该考生的准考证号。

36. 防疫隔离系统。

江苏省按照防疫要求，要求所有入苏人员需要隔离 14 天（如某河南人从上海进入江苏，身份证省份为河南，来源省份为上海）。请你帮助工作人员设计防疫隔离系统。

- (1). 输入 n 个人的身份证号，姓名，来源省份（可能是境外），开始隔离日期（假设每人隔离 14 天）。
 - (2). 输入某日期，输出当日隔离人员情况（隔离期已满不显示）；
 - (3). 输入某日期，按照剩余隔离日期由多到少排序（隔离期已满不显示）；
 - (4). 根据身份证号，统计各省隔离人员人数（隔离期已满不显示）；
 - (5). 按照年龄输出输出当日隔离人员情况（隔离期已满不显示）；
 - (6). 按照性别输出当日隔离人员情况（隔离期已满不显示）；
 - (7). 输入某日期和来源省份，输出当日隔离人员情况（隔离期已满不显示）。
- （注：身份证前 1、2 位数字表示：所在省（直辖市、自治区）的代码，第 7—14 位数字表示：出生年、月、日，第 17 位数字表示性别：奇数表示男性，偶数表示女性。）

省市	号码
河南	41
湖北	42
湖南	43
广东	44
广西	45
海南	46
重庆	50
四川	51
贵州	52
云南	53
西藏	54
陕西	61
甘肃	62
青海	63
宁夏	64
新疆	65



省市	号码
北京	11
天津	12
河北	13
山西	14
内蒙古	15
辽宁	21
吉林	22
黑龙江	23
上海	31
江苏	32
浙江	33
安徽	34
福建	35
江西	36
山东	37

代码格式：

如下所示，其中“[代码]”要有适当的注释，注释量不低于代码量的 30%，可以用中文。
“[运行结果]”中要能够体现测试结果，边界值、特殊值的结果等。“[小结]”可以写遇到的困难，解决的方案，收获和体会等。

[题目]N 个人的成绩存放在 score 数组中，编写函数，用于返回低于平均分的人数，并将低于平均分的分数存放在 below 数组中。

[代码]

```
#include <stdio.h>
#define N 9
/* 找出低于平均分的人数，并返回 */
int fun(int score[ ], int m, int below[])
{
    int i, k=0, aver=0;           /* aver 保存总分和平均值 */
    while(i<m)                   /* 循环遍历所有分数 */
    {
        aver+=*(score+i);        /* 累加求总分 */
        i++;
    }
    aver/=m;                     /* 求平均分 */
    i=0;                         /* 再次遍历数组 */
    while(i<m)
    {
        if(*(score+i)<aver)       /* 寻找低于平均分的分数 */
        {
            below[k++]=*(score+i); /* 保存到数组 below 中 */
        }
        i++;
    }
    return k;                    /* 返回低于平均分的分数个数 */
}

int main(void)
{
    int i, n, score[N], below[N];
    printf("input students' score:\n"); /* 输入学生分数 */
    for(i=0; i<=N-1; i++)
    {
        scanf("%d", &score[i]);
    }
    n=fun(score, N, below);        /* 调用 fun 函数，求得低于平均分的人数 */
    printf("\nBelow the average score are:\n");
    for(i=0; i<=n-1; i++)         /* 输出低于平均分的人 */
    {
        printf("%d ", below[i]);
    }
    return 0;
}
```


[运行结果]

```
input students' score:
```

```
88 78 45 90 99 67 83 53 92
```

```
Below the average score are:
```

```
45 67 53
```

[小结]此题涉及到了数组、循环语句、查找算法、函数返回等内容。必须注意平均分 aver 的赋值情况。如不能在一开始赋值为 0，则结果将不正确。我一开始遇到了 XXXX 问题，通过 XXXX 方法解决了。