

C 语言 课 程 设 计

(2021 级潇湘计算机专业用)

指 导 书

编著：向德生

湖南科技大学潇湘学院计算机系

2022 年 2 月 20 日

第 1 章 课程设计总体要求

C 语言课程设计是 C 语言教学中一个重要的实践环节。一般来讲，课程设计要比课后作业及课程实验复杂一些，要求所涉及的内容应该有一定的深度和广度，并更加接近现实应用。通过课程设计的综合训练，培养学生实际分析问题和解决问题的能力，最终目标是想通过这种形式，帮助学生系统掌握 C 语言课程的主要内容，为今后的程序开发及后续课程的学习打下扎实的基础。下面主要介绍 C 语言课程设计的训练目标、要求和任务。

1.1 课程设计训练的目标

使学生综合使用所学过的 C 语言程序设计知识，充分发挥自主学习的能力和查找资料的能力，解决稍微复杂的结构化程序设计问题，加深对所学知识的理解与掌握，增强学生利用自己所学知识解决实际问题的能力，为以后的程序开发打下基础。

课程设计训练的目的和要求：

- 1) 使学生巩固和加强《C 语言程序设计》课程的理论知识。
- 2) 进一步理解和运用结构化程序设计方法，学会利用流程图或 N-S 图表示算法。
- 3) 使学生掌握利用 Visual C++、Code::Blocks、Dev C++等工具调试程序的基本方法。
- 4) 使学生学会撰写课程设计报告，逐渐熟悉软件设计相关文档的撰写。
- 5) 通过书刊资料和网络资源，培养学生独立分析问题和解决问题的能力。
- 6) 初步掌握开发一个小型实用系统的基本方法。结合实际应用的要求，使课程设计既覆盖知识点，又接近工程实际需要。
- 7) 培养学生良好的程序设计风格。在实际编程中，为了提高编程质量，对空行、空格和注释均有要求。学生在课程设计书写代码时，应该严格按照要求处理，以便建立良好的程序设计风格（可参阅“[高质量编程指南](#)”一文）。

1.2 课程设计训练的内容

本设计分为 3 部分任务。**训练和考查重点是“结构体数组、链表和文件”**。最低要求是**“任务 1 必须至少完成 5 道链表题目，且任务 2 和任务 3 必须各完成 1 道题目”**。各班课程设计的具体要求最终由[班级指导老师](#)确定。

课程设计题目集为[科大 OJ 网站](#)“2022 年上学期计算机学院 2021 级《C 语言课程设计》”（如下图所示），请谨慎诚信提交代码（若有明显抄袭行为，可直接计为**不及格**）。

ID	Name	Status
2550	2022年上学期计算机学院2021级《C语言课程设计》	运行中 剩余 5天4小时7分43秒

1.2.1 任务 1

首先学习[附件文件“程序设计实例”](#)第 1 节的内容，然后在[科大 OJ 网站](#)上找到实验题目集“2022 年上学期计算机学院 2021 级《C 语言课程设计》”中题目名没有以“任务”二字为前缀的题目。题目主要是训练大家熟悉一些常用的算法设计方法，引导大家开始考虑算法的效率。

说明：这部分内容由于平时在课程实验中已经有了一定的训练量，且在上学期期末考试之前已经在 OJ 网站上发布并提醒大家在寒假完成，故本部分设计内容只安排**半天**的时间（即周一上午），时间不够的话，请各位合理利用课余时间完成。

1.2.2 任务 2

任务 2 训练模块化程序设计方法。首先学习并消化附件文件“程序设计实例”第 254 页，然后仿照编程，实现对自己所选课题的解答。

本部分设计内容安排 2 天的时间（即周一下午至周三上午）。

1.2.3 任务 3

任务 3 通过设计一个实用的应用程序来提高学习的热情与兴趣。

本部分设计内容安排 2 天的时间（即周三下午至周五上午）。

说明 1：任务 2 和任务 3 是非常有代表性的题目，若全部完成，学生可与班级指导老师议定增加与本人所学专业密切相关且能用 C 语言实现的题目。

说明 2：周五下午 3:00—5:00 为课程设计考试时间。考试成绩折算成测试分（40 分）。

1.3 课程设计训练的要求

1.3.1 训练安排

1) 训练时间：2022 年上学期第 1 周各工作日 1—8 节。

2) 训练地点：湖南科技大学逸夫楼，具体安排如下表所示。

课程名称	授课教师	排课人数	开课班级	地点
C语言课程设计	陈燕晖	32	21大数据1班	217-2
C语言课程设计	陈向	31	21大数据2班	217-2
C语言课程设计	文宏	32	21大数据3班	217-2
C语言课程设计	符开耀	30	21计算机1班	330
C语言课程设计	刘敏	30	21计算机2班	330
C语言课程设计	赵肄江	29	21计算机3班	330
C语言课程设计	何勇	30	21计算机4班	330
C语言课程设计	朱海燕	30	21计算机5班	330
C语言课程设计	廖苗	29	21计算机6班	222
C语言课程设计	李锋	28	21计算机7班	222
C语言课程设计	蒋清明	34	21软件1班	220
C语言课程设计	康国胜	35	21软件2班	220
C语言课程设计	林剑	34	21软件3班	220
C语言课程设计	刘毅志	36	21物联网1班	222
C语言课程设计	余庆春	35	21物联网2班	222
C语言课程设计	向德生	33	21信息安全1班	222
C语言课程设计	欧阳军林	32	21信息安全2班	326
C语言课程设计	李曙红	31	21信息安全3班	326
C语言课程设计 A	郑秋匀	32	21计算机01班	217-2
C语言课程设计 A	刘敏	29	21计算机02班	217-2
C语言课程设计 A	赵肄江	31	21计算机03班	217-2
C语言课程设计 A	李锋	30	21计算机04班	220
C语言课程设计 A	朱建军	28	21计算机05班	220
C语言课程设计 A	朱海燕	28	21计算机06班	220
C语言课程设计 A	谢艳春	30	21计算机07班	220

3) 学生训练的注意事项:

- a) 最重要的是同学们要勤奋，不要偷懒！
- b) **1 周各工作日 1—8 节课均在实验室进行训练**，按照要求认真编写和调试程序。不得无故缺勤，不允许迟到早退，不得闲聊和玩游戏。这些平时表现的成绩占总评成绩的 20%。
- c) 课程设计任务以自主钻研为主，遇到解决不了的问题时，可及时与班级指导老师（即上表中的授课教师）交流，或与同学交流，或查阅网络资源。但请注意，代码相似度高的代码将会被重点检查是否有明显抄袭行为。若被认定为抄袭，将被直接认定为“不及格”等级。
- d) 课程设计期间，报告内容要及时填写，班级指导老师将定期检查。
- e) 总评成绩将根据平时表现、设计成果和报告质量按比例进行综合评分，班级指导老师最终按“优、良、中、及格或不及格”给出总评成绩等级。

1.3.2 实验环境

- 1) 硬件：PC 机，CPU 档次不低于 PIII 667，内存>512M，硬盘>10G。
- 2) 软件：操作系统为 Windows，编译器为 Visual C++、Code::Blocks、Dev C++等。

1.3.3 评分标准

总评成绩将根据以下 4 部分成绩进行综合评分后划分为 5 个等级，分别为优秀、良好、中等、及格和不及格。

- 1) **平时分（20 分）**。平时分由班级指导老师根据学生平时表现进行综合评定。指导老师进行不定期考勤，学生不得以自己有私人电脑为借口而不来实验室。在实验室要求认真编写和调试程序，不得无故缺勤，不允许迟到早退，不得闲聊和玩游戏。若发现**无故缺勤或玩游戏等行为**，**班级指导老师可根据其表现进行扣分**。若平时表现良好，且达到最低设计要求**“任务 1 必须至少完成 5 道链表题目，且任务 2 和任务 3 必须各完成 1 道题目”**，平时分即可给满分 20 分。
- 2) **测试分（40 分）**。测试分根据第一周五下午 3:00—5:00 的**课程设计考试成绩**进行折算计分。课程设计考试主要考查**“结构体数组、链表和文件”**，考查题目的设计将与任务 2 和任务 3 密切相关。由于任务 2 和任务 3 是多年陈题，人工测试的区分度不够，故本次课程设计采用考试的方式检查课程设计的完成质量。平时练习就请测试 OJ、编译器和浏览器等考试环境是否能正常使用，若对机房电脑的性能不满意，可自带笔记本参加设计或考试。
- 3) **创新分（10 分）**。创新分由班级指导老师根据你**一题多解算法优化的能力和提交代码的相似度**进行综合评定。学生在平时练习时，可随时与班级指导老师交流，以加深指导老师对你创新能力的了解。在完成任务 2 或任务 3 后，请及时向班级指导老师汇报并请老师检查，以让老师记录你的完成情况和创新能力。同学们请注意，若课程设计期间，未与班级指导老师有任何交流，**则创新分极可能为 0 分**。
- 4) **报告分（30 分）**。报告分由班级指导老师根据报告内容的规范度、丰富度和质量进行综合评定。课程设计报告的具体上交时间由指导老师确定。若**未交报告**，**则总评成绩直接计为不及格**。报告排版要规范，要有解题思路、算法描述（包含格式规范的流程图或 N-S 图）、算法分析（有多种解法的可对算法效率进行说明或比较）、遇到的问题及解决办法等。若报告内容仅仅只有代码，报告分最多计 10 分。报告内容中，任务 1 至少写 5 道链表题目，任务 2 至少写 1 道题目，任务 3 至少写 1 道题目。

1.3.4 其它说明

- 1) 各班课程设计的过程管理与质量控制**由各班指导老师全权负责**。
- 2) 实验报告表格提供一份 word 版电子稿。关于实验报告是手工填写还是采用电脑填写，一般的要求是手工填写，即先打印实验报告表格，然后手工填写实验内容。部分编程能力强的同学可向指导老师申请采用电脑填写，指导老师同意后方可采取先填写实验内容后打印的方式。**各班以指导老师的要求为准**。
- 3) 由于本指导书编者水平有限，并受惯性思维约束，任务的设计缺乏创意，因此，恳请各班指导老师，您可对本次课程设计的任何题目酌情调整工作量及难度，**也可另外布置题目**，以更好地训练同学们的 C 语言运用及编程能力。
- 4) 课程设计期间的技术支持、经验交流、通知发布等，请关注本班班级指导老师的通知及[科大 OJ 网站](#)。
- 5) 若有疑问或建议，可直接与编者联系，邮箱：30047495@qq.com；电话：18607323285。

第 2 章 任务 1（趣味程序设计）

2.1 设计目的

通过趣味程序，激发学习兴趣和热情，训练学生分析问题、算法设计、编码、调试、测试程序的能力，引导学生熟悉常用的算法设计方法（参考附件“程序设计实例”第 1 节）。

重点训练知识点：控制结构、函数、数组。

2.2 设计要求

先认真阅读、分析并消化附件“程序设计实例”第 1 节的实例，并从网络查询了解更多算法设计方法；然后再开始做任务 1 的训练题。

题目的选择不受限，可以自由选择。以充分挖掘各位的潜力。

注意：与别人代码的相似度超过 50% 的解答不允许写在解题报告上。鼓励一题多解！

2.3 题目

在科大 OJ 对应题目集提交，会选用 20 道程序设计题。

做题目时，可以寻求老师、同学与网络的帮助，但**禁止原版抄袭！**

OJ 系统具有代码查重功能，代码相似度将以百分比的形式显示在评判结果之后，供有相应权限的管理人员查看。

凡是**代码相似度超过 70% 将进行人工复核。**

2.4 评判系统的常见问题解答

Q: 如何参加本次课程设计任务的练习？

A: 在科大 OJ 中注册一个帐号（以学号为用户名，以姓名为昵称），然后在主菜单上点选“实验&作业”，找到 ID 为 2238 的题目集链接并进入，开始课程设计任务。

Q: 评判系统使用的编译环境？

A: 评判系统英文名称为 Online Judge，一般简称 OJ，评判系统运行于 Ubuntu Linux。使用 GNU GCC/G++ 作为 C/C++ 编译器。

Q: 为什么我的程序在自己的电脑上正常编译，而评判系统反馈有编译错误！

A: GCC/G++ 的编译标准与 VC6 有些不同，更加符合 C/C++ 标准：

- main 函数必须返回 int，void main 的函数声明会报编译错误。
- i 在循环外失去定义 "for(int i=0...){...}"
- itoa 不是 ansi 标准函数。
- __int64 不是 ANSI 标准定义，只能在 VC 使用，提交 OJ 会出错。

Q: 数据超过了 int 的表示范围，要用 64 位数据，变量定义要注意什么？

A: 编译环境不同，定义不同。但在 OJ 上提交时，需改成 long long 类型再提交。

- 在 VC6 中调试，变量定义用 `__int64` 类型，输入输出格式用 `%I64d`。（注意：`__int64` 前是 2 条下划线，`%I64d` 中的 I 是字母 i 的大写）
- 在 windows 版的 Codeblocks 中调试，变量定义用 `__int64` 类型或 `long long`，输入输出格式用 `%I64d`
- 在 linux 版的 Codeblocks 中调试或在 OJ 上提交时，变量定义只能用 `long long` 类型，输入输出格式用 `%lld`。（注意：`%lld` 中的 ll 是 long long 的 2 个头字母）

Q: 数据存储量大时，变量定义要注意什么？

A: 变量一般定义成局部变量，但其分配内存不能超过 1M，否则运行时会提示“应用程序发生异常”。若超过 1M，请定义成全局变量，全局变量的内存分配可高达 2GB。

Q: 系统返回信息都是什么意思？

A: 详见下述：

- 等待：系统忙，你的答案在排队等待。
- 等待重判：因为数据更新或其他原因，系统将重新判你的答案。
- 运行并评判：正在运行和判断。
- 正确：程序通过！
- 格式错误：答案基本正确，但是格式不对。
- 答案错误：答案不对，仅仅通过样例数据的测试并不一定是正确答案，一定还有你没想到的地方。
- 时间超限：运行超出时间限制，检查下是否有死循环，或者应该有更快的算法。课程设计题目会重点对时间加以限制，因此，请注意算法的优化。
- 内存超限：超出内存限制，数据可能需要压缩，检查内存是否有泄露。
- 输出超限：输出超过限制，你的输出比正确答案长了两倍。
- 运行错误：运行时错误，非法的内存访问，数组越界，指针漂移，调用禁用的系统函数。
- 编译错误：编译错误，请点击后获得编译器的详细输出。

第3章 任务2（小型管理系统程序设计）

3.1 设计目的

通过小型管理系统的开发，引导学生熟悉模块化程序设计方法（参考附件文件“程序设计实例”第2节），培养学生综合运用知识解决实际问题的能力。

重点训练知识点：函数、数组、指针、结构体、文件。

3.2 设计要求

1) 学习并消化附件文件“程序设计实例”第2节。在老师提供的软件包中，找到子目录“成绩管理系统（采用结构体数组）的示例参考程序”和“成绩管理系统（采用链表）的示例参考程序”，阅读研究源代码并调试运行。程序中的每一条语句要看懂并能回答老师提问，**重点难点是结构体数组、链表和文件的使用。**

2) 每位同学可以从“2.5 参考题目”中选择至少一个题目完成，也可以与指导老师议定一个新的同类型题目。

重要声明：系统源码中，要包括多个函数，各个函数要带一个或多个参数，重要数据都要通过函数参数进行传递，除非确有必要，否则**禁止定义不带参数的函数。**

3.3 设计步骤

- 1) 需求分析。分析课题要解决的问题是什么，有什么要求和限制条件。
- 2) 总体设计。分析系统功能，划分功能模块，确定各模块函数名称及数据存储方式。
- 3) 详细设计。主程序算法设计和各模块算法设计。
- 4) 编程实现。编程工具不限，但机房以 VC6 和 Code::Blocks 软件为主。
- 5) 调试和测试。熟练阅读出错提示，熟练使用 debug 工具。主要测试正确性与容错性。
- 6) 完成设计报告。参照后面的参考格式撰写。

3.4 实验报告撰写格式

任务2的实验报告请按照下面的规范进行撰写。

任务2 XXXXXXXX 系统（题目，居中）

1 需求分析（描述问题：简述课题要解决的问题是什么，有什么要求和限制条件）

1.1 问题提出（详细叙述本人所要实现的题目中的问题）

例如：本人计划编写一个学生信息管理系统，主要用来管理学生基本信息及成绩信息。……

1.2 本系统涉及的知识点

例如：循环、分支、数组、函数、结构体、指针、文件

1.3 功能要求（叙述本人所要实现的题目功能）

例如：

- 1) 建立学生信息，每个学生的信息包括：学号、姓名、性别、班级、成绩；
- 2) 求出每个学生的平均分；
- 3) 排序：分别按平均分排序、按学号排序；
- 4) 根据学号查询成绩；
- 5) 输出：……

2 总体设计（进行功能设计与模块划分）

2.1 功能设计（规划本系统完成的功能），例如：

本系统需要实现的功能：

- 1) 利用 switch 语句设计“主菜单”：

*****主菜单*****

请输入选项编号（1~6）

1—— 输入信息

2—— 求平均分

3—— 排 序

4—— 查 询

5—— 输 出

2) 选择 3 后，调用排序函数，进入函数后利用 switch 语句实现一个“排序菜单”，该菜单中每个选项调用一个函数

*****排序菜单*****

请输入选项编号（1~2）

1—— 按平均分排序

2—— 按学号排序

- 3) 确定各模块函数名称，设计数据存储方式。

- a) 输入函数 input()：利用循环输入的方式，将有关信息存入结构数组，结构数组根据实际的人数，采用动态方式建立……
- b) 求平均分函数 average()：…….
- c) …….

2.2 模块图

例如：

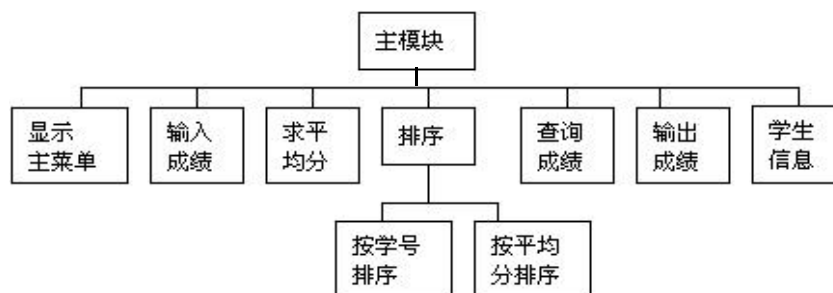


图 1 成绩管理系统模块图

3 详细设计（各模块的功能设计及算法设计）

3.1 模块功能说明

分模块（函数）简述其功能（如函数功能、入口及出口参数说明，函数调用关系描述等）；

例如：

3.1.1 输入模块

- 1) 函数原型：void input(struct student stu[], int n)
- 2) 功 能：利用 for 循环语句和 scanf、gets、getchar 函数完成对结构数组的输入，存放 n 个学生的信息（学号、姓名、性别、4 门成绩）。
- 3) 参数说明：
 - a) int n ： 整型形参变量，接收 main()传过来的数组人数。
 - b) struct student stu[]：接收主函数传过来的数组首地址。
- 4) 说明：执行完输入函数后，会在键盘缓冲区中保存回车键，后面再对字符型变量赋值时，会将缓冲区中的回车键当成数据存入变量中，所以要在某些输入语句后面加 getchar()函数。（若不需说明部分可省略）
- 5)

3.2 部分模块的算法流程图（对部分重要模块，画出其流程图或 N—S 图。）

4 编码与调试

（调试方法，调试过程中遇到的主要问题及解决办法。）

5 测试

（用几组测试数据，测试算法设计的正确性；测试结果的分析与讨论）

6 用户手册。（软件使用说明）

7 结束语。本人在程序设计中感想，如：收获；遇到的困难（因某知识点欠缺，编写的程序哪部分有错；因马虎或××方面欠考虑，运行结果不相符等）；如何解决问题及困难等；并用简短的文字对在本次设计过程中给予帮助的人员表示感谢。

8 附录。包括：按规范格式书写的全部源代码（源程序要有足够的注释）；其它参考资料或说明等。

3.5 参考题目

以下几个选题，每位同学至少选择一题完成。参考题目如下：

1、隐私信息管理系统

本题要求采用结构体数组（或链表）存储数据。具体要求如下：

- 1) 程序执行时，首先要进行密码检测，以不让非法用户使用本程序。标准密码预先在程序中设定，程序运行时，若用户的输入密码和标准密码相同，则显示“口令正确！”并转去执行后续程序；若不相等，重新输入，3次都不相等则显示“您是非法用户！”并终止程序的执行。
- 2) 现在每个人在不同网站都有用户名和密码等信息，还有银行卡号及密码信息，众多的信息经常忘记，最好的方法是将这些信息用文件的形式保存起来。但是很多私密数据（特别是密码）不能明文存储，比较保险的办法是采用密文存储，即先将密码明文加密成密文后，再进行存储。加密方法自行设计，例如输入密码明文为字符 1，但存储时却存储的是其密文字符 a，这只要通过 ASCII 值加一个整数值即可实现。
- 3) 需要保护的数据信息包括编号，帐号位置，帐号描述，帐号名，密码等，如：

编号	帐号位置	帐号描述	帐号名	密码
1	https://next.xuetangx.com	学堂在线	xiangdesheng	123456
2	https://webvpn.hnust.edu.cn/	湖科大 WebVPN	hnust8001	654321
3	https://www.icourse163.org	中国大学 MOOC	30047495@qq.com	123456

此表仅为示例，设计时请根据自己的设计确定由哪些信息组成结构体。

- 4) 系统具体提供以下功能：
 - 系统以菜单方式工作。开始运行程序时要进行密码验证。
 - 信息的录入功能、浏览功能。
 - 信息的查询功能。按帐号名查询，如输入“hnust”可查出上表第 2 条记录。
 - 信息的删除、修改功能。
 - 信息存入文件。信息中的密码必须要加密后才能存入文件，建议先输入一串字符作为加密密钥，对信息中的密码加密后再进行存储，即确保存入文件中的信息里面，“密码”是加密后的字符串。
 - 将文件中的信息导入。与信息存入文件的过程相反，将密码解密后导入。

- 5) 训练的主要知识点：数组、指针、结构体、文件。

上面的功能需求为编者提供的建议性方案，同学们可进行完善性修改。

2、红包管理系统

本题要求采用结构体数组（或链表）存储数据。具体要求如下：

- 1) 系统有多个用户，用户登录口令检测正确后，进入红包管理系统，要实现类似 QQ 红包或微信红包的功能模拟。
- 2) 每个用户的数据包含用户名，昵称，学号，零钱余额，红包等信息。
- 3) 发红包功能：用户可以发不超过零钱余额的红包（分 X 个包共 Y 元）。
- 4) 抢红包功能：有人发红包之后，才能抢红包。所抢得的红包累加到零钱包。
- 5) 每次数据变更都要存入文件中，这样另一个用户登录就能访问文件中的最新数据。
- 6) 能查询红包的发放明细、零钱包的收入支出明细。

上面的功能需求为编者提供的建议性方案，同学们可进行完善性修改。尽量模拟一下抢红包和设计一个红包的随机分配算法（可用到 `srand()` 和 `rand()` 函数，参考 3.6.3 示例）。

3、单项选择题标准化考试系统

本题要求采用结构体数组（或链表）存储数据。具体要求如下：

- 1) 系统以菜单方式工作。开始运行程序时要进行密码验证。
- 2) 用文件保存试题库。（每个试题包括题干、4 个备选答案、标准答案）。
- 3) 试题录入：可随时增加试题到试题库中。
- 4) 试题抽取：每次从试题库中可以随机抽出 N 道题（N 由键盘输入）。
- 5) 答题：用户可实现输入自己的答案。
- 6) 自动判卷：系统可根据用户答案与标准答案的对比实现判卷并给出成绩。
- 7) 项目组可与指导老师议定增加新功能模块。
- 8) 训练的主要知识点：数组、指针、结构体、文件。

上面的功能需求为编者提供的建议性方案，同学们可进行完善性修改。

3.6 关键问题释疑

3.6.1 实例 1：采用结构体数组编写学生成绩管理系统

仔细阅读附件文件“程序设计实例”第 254—266 页，编译调试程序（软件包中有源码）并跟踪程序设计的执行，希望大家通过这个程序熟悉结构体数组的使用，熟悉函数参数的传递及其他模块化程序设计方法。本程序的难点是文件的读写。

3.6.2 实例 2：采用链表编写学生成绩管理系统

仔细阅读附件文件“程序设计实例”第 254—266 页，编译调试程序（软件包中有源码）并跟踪程序设计的执行，希望大家通过这个题熟悉链表的操作，为学习数据结构打下良好的基础。本程序的难点是链表的建立、插入、删除等操作。

3.6.3 随机数的生成

红包算法可能用到随机数的生成。随机数的生成可调用 `rand()` 函数，下面的程序实现生成 5 个取值为 1-10 的随机数字。

使用说明可参考 <http://www.cnblogs.com/xianghang123/archive/2011/08/24/2152404.html>。

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int main(void)
{   int i;
    srand((unsigned)time(NULL));
    for(i=1;i<=5;i++)
        printf("%d\n",rand()%10+1);
    return 0;
}
```

第 4 章 任务 3（小型应用程序设计）

4.1 设计目的

通过小型应用程序的开发，培养综合运用知识解决实际问题的能力，训练分析问题、算法设计、编码及调试程序的能力。

重点训练知识点：数组、指针、结构体、文件。

4.2 设计要求

1) 阅读“3.5 关键问题释疑”，学习相关背景知识。

2) 在老师提供的软件包中，找到子目录“电子印章题目 1 的示例参考程序”、“电子印章题目 2 的示例参考程序”。这是 2 个演示用可运行程序，意味着你在后面 2 个题目的编程时至少要达到这样的运行效果。

3) 在老师提供的软件包中，找到子目录“制作 BMP 图片的示例参考程序”。研究源码并修改，即可完成下面 4.4 小节中参考题目的编程任务。

4.3 实验报告撰写格式

参见任务 1 的解题报告示例。

4.4 参考题目

以下 2 个题目中，至少完成题目 1 的程序设计。

题目 1：先为自己的名字制作一个汉字点阵 txt 文件，然后编程，通过程序读取汉字点阵 txt 文件中的信息，根据点阵信息决定向 BMP 文件中写入红点或白点数据，从而制作出自己的一枚电子印章。

编程前准备：利用隶书 56 点阵汉字字库文本文件 LiShu56.txt（在老师提供的软件包中）制作印章字库文本文件 XXX.txt（用记事本软件即可制作完成），要求与印章的摆放顺序一致。其中 **XXX 指自定义的任意名称**。如图 2 所示，是编者自己制作的字库文件文件 xds.txt。

编程时的注意事项：

- 1) 建立存储印章的文件 XXX.bmp，然后按照 BMP 文件的格式往 XXX.bmp 文件中写入文件头、信息头。
- 2) 然后根据 XXX.txt 文件中的内容确定颜色数据写入 XXX.bmp 文件中。XXX.txt 文件中的一个字符对应于 XXX.bmp 文件中一个彩色点的数据（包括蓝、绿、红三个字节数据，若是红色点，一般置 B=0、G=0、R=255）。构成字形的数据对应写入红色点数据，未构成字形的数据对应写入白色点数据。
- 3) 检查文件头、信息头、彩色数据是否符合 BMP 文件的格式规范，若规范，则在 windows 环境下可以正常浏览和使用该电子印章图片了。
- 4) 印章字库文本文件 XXX.txt 和生成的印章文件 XXX.bmp 通过命令行参数给定。假如程序编译连接后生成的可执行文件名为 mySeal.exe，则用 xds.txt 制作印章 xds.bmp 的命令为：mySeal xds.txt xds.bmp。由图 2 生成的印章如图 3 所示。

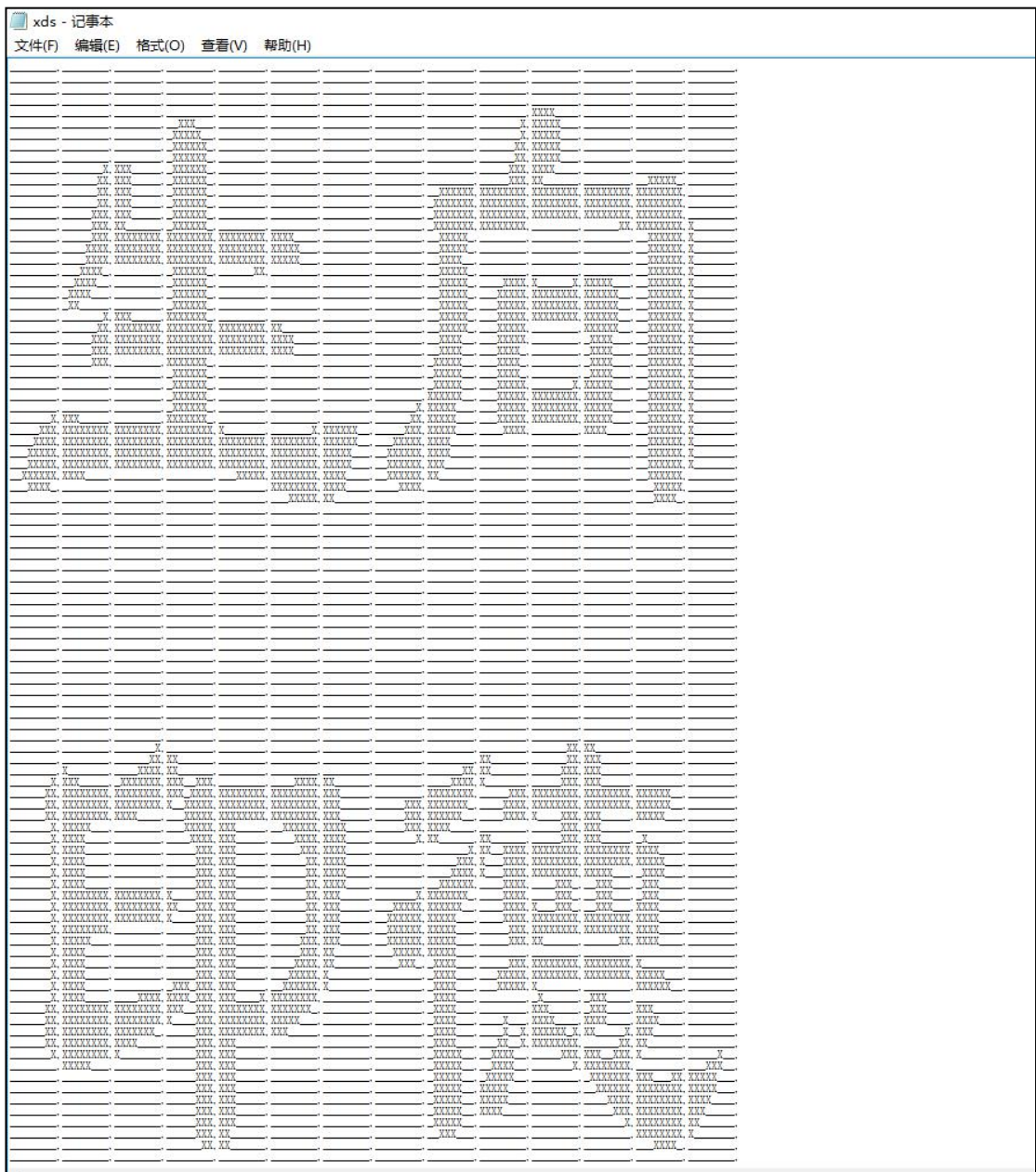


图 2 通过记事本浏览 xds.txt 文件的部分内容



图 3 由图 2 生成的 xds.bmp 印章

题目 2：编程实现根据输入的汉字制作不同的电子印章

具体要求如下：

- 1) 从键盘输入 4 个汉字，根据汉字的编码在隶书 56 点阵汉字字库文本文件 LiShu56.txt 中查找相应汉字的字型点阵信息，并将点阵信息存入数组。
- 2) 建立印章文件 XXX.bmp，按照 BMP 文件的格式写入文件头、信息头。
- 3) 然后根据数组中的汉字点阵信息确定颜色数据写入 XXX.bmp 文件中。XXX.txt 文件中的一个字符对应于 XXX.bmp 文件中一个彩色点的数据（包括蓝、绿、红三个字节数据，若是红色点，一般置 $B=0$ 、 $G=0$ ， $R=255$ ）
- 4) 检查文件头、信息头、彩色数据是否符合 BMP 文件的格式规范，若规范，则在 windows 环境下可以正常浏览和使用该电子印章图片了。
- 5) 隶书点阵文本文件 LiShu56.txt 和生成的印章文件 XXX.bmp 通过命令行参数给定。假如程序编译连接后生成的可执行文件名为 mySeal.exe，则用 LiShu56.txt 制作印章 XXX.bmp 的命令为：mySeal LiShu56.txt XXX.bmp，执行该命令后输入“令狐冲印”4 个汉字生成的红色印章如图 4(a)所示。采用其它字体点阵文件生成的印章如图 4(b)-图 4(d)所示，

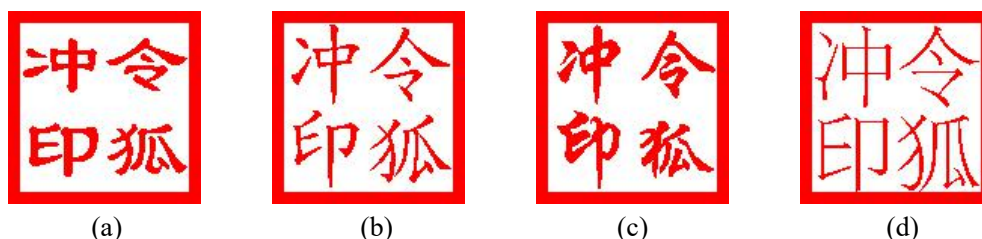


图 4 从左至右分别是由隶书、楷体、华文行楷、宋体点阵 txt 文件生成的印章

备注：有兴趣的可与指导老师议定增加功能，比如生成不同规格形状的电子印章等。

4.5 关键问题释疑

4.5.1 彩色图像的显示原理

对于彩色图像，它的显示必须从三原色 RGB 概念说起。众所周知，自然界中的所有颜色都可以由红（R）、绿（G）、蓝（B）三原色组合而成。有的颜色含有红色成分多一些，其它成分少一些。针对含有红色成分的多少，可以人为地分成 0 到 255 共 256 个等级，0 级表示不含红色成分，255 级表示含有 100% 的红色成分。同样，绿色和蓝色也可以被分成 256 级。这样，根据红、绿、蓝各种不同的组合我们就能表示出 $256 \times 256 \times 256$ （约 1600 万）种颜色。表 1 是常见的一些颜色的 RGB 组合值。当一幅图中每个像素被赋予不同的 RGB 值时，就能呈现出五彩缤纷的颜色了，这就形成了彩色图像。

真彩色图像（又称 24 位色图像）的颜色种类高达 $256 \times 256 \times 256 = 2^{24} = 16777216$ 种，也就是包含上述提到的 R、G、B 颜色表示方法中所有的颜色。真彩色图像是说它具有显示所有颜色的能力，即最多可以包含所有的颜色。通常，在表示真彩色图时，每个像素直接用 R、G、B 这 3 个分量字节来表示，而不采用调色板技术。原因很简单：如果使用调色板，表示一个像素颜色在调色板中的索引要用 24 位（因为共有 2^{24} 种颜色，即调色板有 2^{24} 行），这和直接用 R、G、B 这 3 个分量表示用的字节数一样，不但没有节省任何空间，还要加上一个 $256 \times 256 \times 256 \times 3 = 3 \times 2^{24}$ 字节的大调色板。所以真彩色图直接用 R、G、B 这 3 个分量表示。

表 1 常见颜色的 RGB 组合

颜色	红色成分	绿色成分	蓝色成分
黑色	0	0	0
白色	255	255	255
红色	255	0	0
绿色	0	255	0
蓝色	0	0	255
青色	0	255	255
紫色	255	0	255
黄色	255	255	0
灰色	128	128	128
银色	192	192	192

4.5.2 BMP 彩色图像的文件格式

BMP 图像文件格式，是微软公司为其 WINDOWS 环境设置的标准图像格式，并且内含了一套图像处理的 API 函数。随着 WINDOWS 在世界范围内的普及，BMP 文件格式越来越多地被各种应用软件所支持。BMP 图像文件是位图文件，位图表示的是将一幅图像分割成栅格，栅格的每一点称为像素，每一个像素具有自己的 RGB 值，即一幅位图是由一系列像素点构成的点阵。一个 BMP 文件包括位图文件头 BITMAPFILEHEADER、位图信息头 BITMAPINFOHEADER、调色板 PALETTE 和位图像素数据 4 个部分，如图 5 所示。其中，前面 3 部分的结构在 windows.h 中进行了定义。下面对它们进行详细说明。

位图文件头BITMAPFILEHEADER
位图信息头BITMAPINFOHEADER
调色板PALETTE
实际的位图像素数据

图 5 BMP 文件结构示意图

一、位图文件头

位图文件头结构 BITMAPFILEHEADER 包含位图文件的类型大小信息和版面信息。结构如下：

```
typedef struct tagBITMAPFILEHEADER    // bmfh
{
    WORD    bfType;
    DWORD   bfSize;
    WORD    bfReserved1;
    WORD    bfReserved2;
    DWORD   bfOffBits;
} BITMAPFILEHEADER;
```

这个结构的长度是固定的，为 14 个字节，各个域的说明如下：

- bfType：指定文件类型，必须是 0x4D42，即字符串“BM”。也就是说所有 BMP 文件的头两个字节都是“BM”。

- **bfSize:** 指定整个文件的大小（以字节为单位）。
- **bfReserved1:** 保留，一般为 0。
- **bfReserved2:** 保留，一般为 0。
- **bfOffBits:** 指定从文件头到实际的位图像素数据首部的字节偏移量。即图 5 中前 3 个部分的长度之和。

二、位图信息头

位图信息头结构 BITMAPINFOHEADER 包含图像本身的属性。其定义如下：

```
typedef struct tagBITMAPINFOHEADER    // bmih
{
    DWORD   biSize;
    LONG    biWidth;
    LONG    biHeight;
    WORD    biPlanes;
    WORD    biBitCount
    DWORD   biCompression;
    DWORD   biSizeImage;
    LONG    biXPelsPerMeter;
    LONG    biYPelsPerMeter;
    DWORD   biClrUsed;
    DWORD   biClrImportant;
} BITMAPINFOHEADER;
```

这个结构的长度是固定的，为 40 个字节，各个域的说明如下：

- **biSize:** 指定 BITMAPINFOHEADER 结构的长度，为 40 个字节。
- **biWidth:** 指定位图的宽度（以像素为单位）。
- **biHeight:** 指定位图的高度（以像素为单位）。
- **biPlanes:** 指定目标设备的位面数。这个成员变量的值必须为 1。
- **biBitCount:** 指定每个像素的位数。常用的值为 1（黑白二色图）、4（16 色图）、8（256 色图）、24（真彩色图）。
- **biCompression:** 指定压缩位图的压缩类型。有效的值为 BI_RGB(0), BI_RLE8(1), BI_RLE4(2), BI_BITFIELDS(3)。用得不多，在 24 位格式中，该变量被设置为 0。
- **biSizeImage:** 指定图像的大小（以字节为单位）。如果位图的格式是 BI_RGB，则将此成员变量设置为 0 是有效的。该值可以根据 biWidth'和 biHeight 的乘积计算出来。要注意的是：上述公式中的 biWidth'必须是 4 的整倍数（所以计算乘积时写的是 biWidth'，表示大于或等于 biWidth 的、离 4 最近的整倍数。例如，若 biWidth=240，则 biWidth'=240；若 biWidth=241，则 biWidth'=244）。
- **biXPelsPerMeter:** 为位图指定目标设备的水平分辨率（以“像素/米”为单位）。
- **biYPelsPerMeter:** 为位图指定目标设备的垂直分辨率（以“像素/米”为单位）。
- **biClrUsed:** 指定位图实际用到的颜色数。如果该值为 0，则用到的颜色数为 2 的 biBitCount 次方。

- biClrImportant: 指定对位图的显示有重要影响的颜色数。如果此值为 0, 则所有颜色都很重要。

三、调色板（注意：印章要求采用 24 位色，不需要调色板，所以这部分可不用考虑）

对于 2 色、16 色和 256 色位图，需要调色板。调色板中的各个元素规定了第 4 部分（实际位图数据）对应的颜色值。对于 24 位色的真彩色图，不需要调色板，第 2 部分（信息头 BITMAPINFOHEADER）后直接是位图数据。

调色板实际上是一个数组，共有 biClrUsed 个元素（如果此值为 0, 则有 2 的 biBitCount 次方个元素）。数组中每个元素的类型是一个 RGBQUAD 结构（4 个字节），定义如下：

```
typedef struct tagRGBQUAD
{
    BYTE    rgbBlue;           //该颜色的蓝色分量
    BYTE    rgbGreen;          //该颜色的绿色分量
    BYTE    rgbRed;            //该颜色的红色分量
    BYTE    rgbReserved;       //保留值
} RGBQUAD;
```

四、图像数据

对于用到调色板的位图，图像数据就是该像素值在调色板中的索引值，对于 24 位色的真彩色图，图像数据就是实际的 R、G、B 值。下面对 2 色、16 色、256 色位图和真彩色位图分别加以介绍。

- 对于 2 色位图，用 1 位就可以表示该像素的颜色（一般 0 表示黑，1 表示白），所以 1 个字节可以表示 8 个像素。
- 对于 16 色位图，用 4 位表示 1 个像素的颜色，所以 1 个字节可以表示 2 个像素。
- 对于 256 色位图，1 个字节刚好可以表示 1 个像素。
- 对于 24 位色真彩色图（无调色板），3 个字节才表示 1 个像素。

注意：

(1) BMP 文件按从下到上，从左到右的顺序存储图像数据。即从文件中最先读到的是图像最下面一行的左边第一个像素，然后是左边第二个像素……接下来是倒数第二行左边第一个像素，左边第二个像素……依次类推，最后得到的是最上面一行的最右一个像素。

(2) 对于 24 位色真彩色位图而言，数据的排列顺序以图像的左下角为起点，从左到右、从下到上，每连续 3 个字节便描述图像一个像素点的颜色信息，这三个字节分别代表蓝、绿、红三基色在此像素中的亮度，若某连续三个字节为：00H, 00H, FFH, 则表示该像素的颜色为红色。24 位真彩色位图中每个像素的 RGB 数据可以定义成如下结构：

```
typedef struct tagRGBDATA{
    BYTE rgbBlue;
    BYTE rgbGreen;
    BYTE rgbRed;
} RGBDATA;
```

五、BMP 文件示例

24 位色真彩色位图文件组成。用编辑软件（用 **HEdit 软件查看**，老师的软件包中会提供。或用教材 235 页编程题第 6 题的可执行程序查看，其源程序可参考实践教程 113 页）查看 24 位色真彩色位图文件 Lena.bmp 的组成，有如下 3 部分：

1. 24 位色真彩色位图的文件头。如图 6 深色部分所示，从文件存放首地址 0x0000 开始，共 14 个字节。

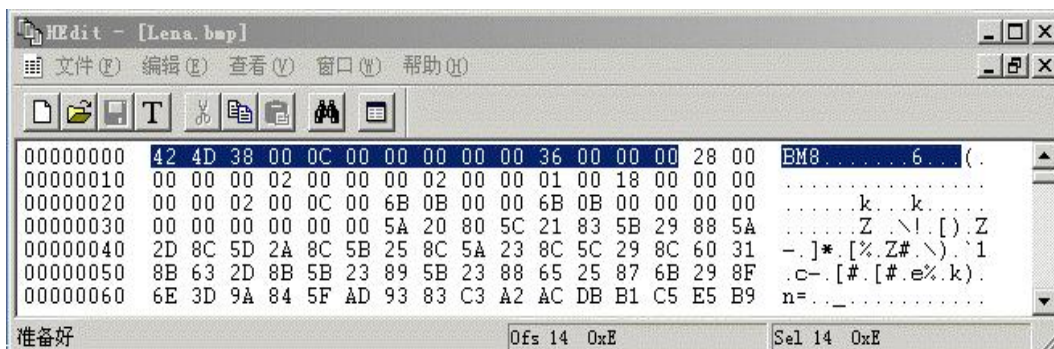


图 6 24 位色真彩色位图的文件头

注：在图 6 中，**BMP 文件的数据看中间这一块数据即可**，每 2 位是一个十六进制数字，对应一个字节。左边的这一列是对应行中首字节在文件中的相对位置（编号），右边的这一块数据只是将中间块数据用其对应的 ASCII 码字符展示而已（不可见字符显示为点号）。

2. 24 位色真彩色位图的信息头。如图 7 深色部分所示，从文件存放地址 0x000E 开始，共 40 个字节。

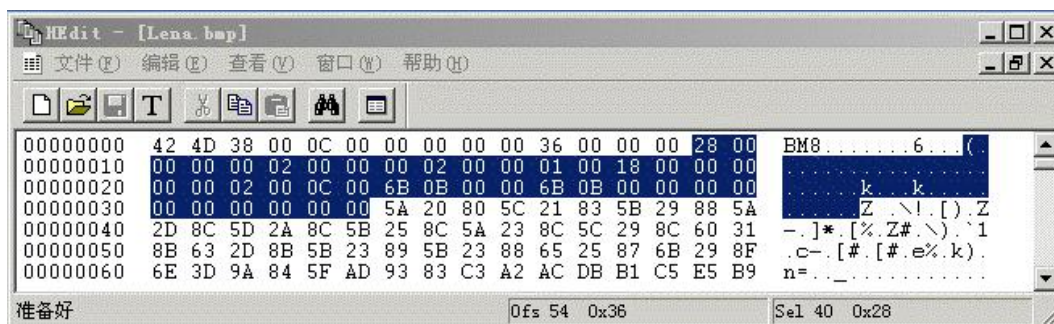


图 7 24 位色真彩色位图的信息头

3. 24 位色真彩色位图的实际像素数据。如图 8 深色部分所示，从文件存放地址 0x0036 开始，共 $512 \times 512 \times 3 = 786432$ 个字节。

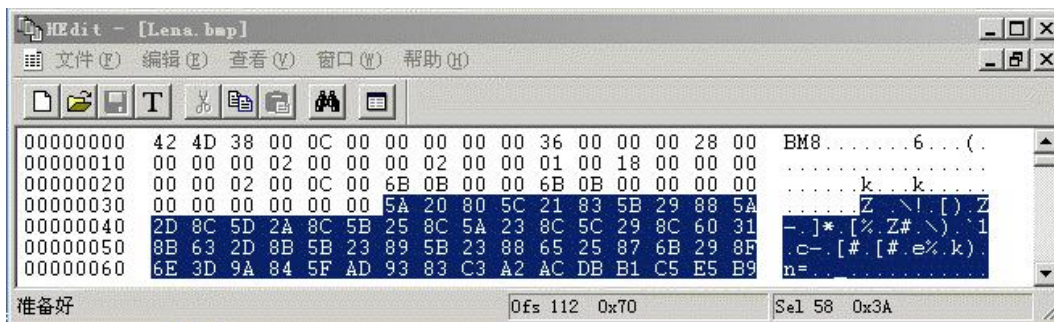


图 8 24 位色真彩色位图的实际像素数据

通过以上实例，我们熟悉了 BMP 彩色图像的格式，然后我们按格式创建 BMP 文件，并将

印章点阵数据写入，则可制作出自己的印章 BMP 图像。

4.5.3 如何制作 BMP 文件

制作一个可在 windows 中正常浏览和使用的 BMP 文件，主要包括如下四个步骤：

第一步：用命令行中给出的文件名新建一 BMP 文件，此时还是一个空文件；

第二步：置文件头数据并写入 BMP 文件；

第三步：置信息头数据并写入 BMP 文件；

第四步：置图像 RGB 数据并写入 BMP 文件。

```
/******  
**** FileName:  demo.c  
**** Function:  真彩色 bmp 图片文件的创建  
**** Usage:    demo  xxxx.bmp  
*****/  
  
#include <CONIO.H>  
#include <STDLIB.H>  
#include <STDIO.H>  
#include <MATH.H>  
#include <MALLOC.H>  
#include <string.h>  
  
typedef unsigned char BYTE;  
typedef unsigned short WORD;  
typedef unsigned long DWORD;  
  
/**** The file header of bmp file 文件头****/  
#include <pshpack2.h> //This file turns 2 byte packing of structures on, then  
sizeof(BITMAPFILEHEADER)=14, otherwise sizeof(BITMAPFILEHEADER)=16  
typedef struct tagBITMAPFILEHEADER {  
    WORD bfType;  
    DWORD bfSize;  
    WORD bfReserved1;  
    WORD bfReserved2;  
    DWORD bfoffBits;  
} BITMAPFILEHEADER;  
#include <poppack.h> //This file turns packing of structures off  
  
/**** The information header of bmp file 信息头****/  
typedef struct tagBITMAPINFOHEADER {  
    DWORD biSize;
```



```

        DWORD biWidth;
        DWORD biHeight;
        WORD   biPlanes;
        WORD   biBitCount;
        DWORD biCompress;
        DWORD biSizeImage;
        DWORD biXPelsPerMeter;
        DWORD biYPelsPerMeter;
        DWORD biClrUsed;
        DWORD biClrImprotant;
    } BITMAPINFOHEADER;

    /**** The RGB data of bmp file 图像 RGB 数据***/
    typedef struct tagRGBDATA{
        BYTE rgbBlue;
        BYTE rgbGreen;
        BYTE rgbRed;
    } RGBDATA;

    int main(int argc, char *argv[])
    {
        RGBDATA *bmpData=NULL; //图像数据指针
        FILE *fp;              //BMP 文件指针
        long i, j, k;
        long width=300;        //图像宽度
        long height=300;       //图像高度
        long dataSize=width*height;
        BITMAPFILEHEADER bmfHeader;
        BITMAPINFOHEADER bmiHeader;
        if(argc<2)
        {
            printf("\n    请指定您要生成的 BMP 文件名! \n");
            printf("\n    方法 1: 在 cmd 窗口输入:  demo  xxxx.bmp  后回车执行! \n");
            printf("\n    方法 2: 在 VC 的工程设置中添加参数:  xxxx.bmp  后编译运行! \n");
            printf("\n    功能:  根据设定红绿蓝三色分量的值, 控制显示不同的彩色条纹! \n\n");
            exit(0);
        }

        printf("\n    *****\n");

```

```

printf("\n          感谢您使用 BMP 图片文件制作 DEMO 程序          \n");
printf("\n          作者：向德生          \n");
printf("\n          日期：2010 年 5 月 12 日          \n");
printf("\n          目的：编制印章生成程序时，供参考！          \n");
printf("\n          功能：根据设定红绿蓝三色分量的值，控制显示不同的彩色条纹！\n\n");
printf("\n          *****\n\n");
//第一步：用命令行中给出的文件名新建一 BMP 文件，此时还是一个空文件
if((fp=fopen(argv[1], "wb+"))==NULL)
{
    printf("Cannot open BMP file!");
    exit(0);
}

//第二步：置文件头数据并写入 BMP 文件
bmfHeader.bfType=0x4d42;
bmfHeader.bfSize=14+40+width*height*3;
bmfHeader.bfReserved1=0;
bmfHeader.bfReserved2=0;
bmfHeader.bfoffBits=0x36;
fwrite(&bmfHeader, sizeof(BITMAPFILEHEADER), 1, fp);

//第三步：置信息头数据并写入 BMP 文件
bmiHeader.biSize=40;
bmiHeader.biWidth=width;
bmiHeader.biHeight=height;
bmiHeader.biPlanes=1;
bmiHeader.biBitCount=24;
bmiHeader.biCompress=0;
bmiHeader.biSizeImage=width*height*3;
bmiHeader.biXPelsPerMeter=0;
bmiHeader.biYPelsPerMeter=0;
bmiHeader.biClrUsed=0;
bmiHeader.biClrImportant=0;
fwrite(&bmiHeader, sizeof(BITMAPINFOHEADER), 1, fp);

//第四步：置图像数据并写入 BMP 文件
//分配足够内存，让 bmpData 指向这块内存，用于存放图像各像素点的 RGB 分量值
if((bmpData=(RGBDATA*)malloc(width*height*3))==NULL)

```

```

{    printf("bmpData memory malloc error!");
}

//先在 bmpData 所指内存中置图像 RGB 数据，然后将所有数据写入 BMP 文件

for(i=0;i<height;i++)
{    for(j=0;j<width;j++)
    {
        k=(height-i-1)*width + j;//计算第 i 行第 j 列图像数据在 bmpData[] 数组中的位置
        if(i<100)//上 100 行只置红色分量值，显示为红色
        {
            bmpData[k].rgbBlue=0;
            bmpData[k].rgbGreen=0;
            bmpData[k].rgbRed=255;//红色分量值为： 255
        }
        else
            if(i<200)//中 100 行只置绿色分量值，显示为绿色
            {
                bmpData[k].rgbBlue=0;
                bmpData[k].rgbGreen=255;//绿色分量值为： 255
                bmpData[k].rgbRed=0;
            }
            else
                if(i<300)//下 100 行只置蓝色分量值，显示为蓝色
                {
                    bmpData[k].rgbBlue=255;//蓝色分量值为： 255
                    bmpData[k].rgbGreen=0;
                    bmpData[k].rgbRed=0;
                }
    }
}

fwrite(bmpData, sizeof(RGBDATA), dataSize, fp);//将 bmpData 所指的 RGB 数据一次性写入 BMP 文件

printf("\n    恭喜您！BMP 文件已经成功生成！\n");
printf("\n    请在当前目录下查看生成的 BMP 文件%s\n\n", argv[1]);
free(bmpData);    //释放 bmpData 所指的内存空间
bmpData=NULL;    //置 bmpData 为空指针
fclose(fp);    //关闭 fp 所指文件
}

```

该程序生成的大小为 300×300 的 BMP 图片，可在运行本程序的目录下查看。

重要提示：任务 2 的编程与本示例编程的不同主要是在第四步。

4.5.4 如何根据输入的汉字确定其编码

我们看到屏幕上显示的汉字的字型有两种表达方式：一种称为矢量方式，一种称为点阵方式。其中的点阵方式较为简单，其原理就是好比：铺地砖。有的铺为白色，有的铺为黑色，只要精心安排，就会组成我们希望的图案，当然也可以是汉字。

瓷砖越多，铺出的图案效果越细腻，这就是点阵的规模。56 点阵汉字用 $56 \times 56 = 3136$ 个像素点来描绘。

当我们需要记录一个汉字信息的时候，当然不是存储汉字的字型，而是存储它的编码。汉字可以有多种编码的规则，其中 GB2312 的编码规则是：一个汉字用两个字节表示，前一个字节表示区号，后一个表示区中的偏移序号。

每个区有 94 个汉字。区号和序号的编码都是从 0xA1 开始（为了避免和西文冲突）。已知某个汉字的 GB2312 编码，就可以计算出它在二进制字模文件中所在的绝对位置。我们在本次课程设计中，不用二进制字模文件。而是用文本文件 LiShu56.txt，因为其中的点阵信息非常形象直观，适合我们这些 C 的初学者。

下面程序能从键盘获得一个或多个汉字的编码，如输入“向”，则其编码为 CF F2，有了这个编码，我们就可以在 LiShu56.txt 文件中查找其点阵信息。

```
#include<stdio.h>
int main(void)
{
    int i, len;
    unsigned char BM[100];
    printf("请输入一个或多个汉字：");
    scanf("%s", BM);
    for(len=0; BM[len]!='\0'; len++);
    for(i=0; i<len; i++)
        printf("BM[%d]=%x\n", i, BM[i]); //输入"向", 则其编码为 CF F2
    return 0;
}
```

4.5.5 如何将编码变成字符串？

我们知道，“向”的编码为 2 个字节，分别是 CF 和 F2，如何将其变成字符串（包括字符‘C’、‘F’、‘F’、‘2’，占 4 个字节）存入字符数组中，以便于在 LiShu56.txt 文件中查找其点阵信息。

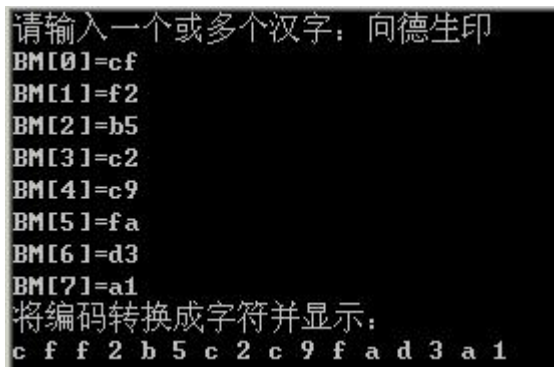
```
#include<stdio.h>
int main(void)
{
    int i, len;
    unsigned char BM[100];
    char str[200];
    printf("请输入一个或多个汉字：");
    scanf("%s", BM);
```

```

for(len=0;BM[len]!='\0';len++);
for(i=0;i<len;i++)
    printf("BM[%d]=%x\n",i,BM[i]); //输入"向", 则其编码为 CF    F2
printf("将编码转换成字符并显示: ");
for(i=0;i<len;i++)
    sprintf(str+2*i,"%x",BM[i]); //编码->字符, 并存入 str 数组
for(i=0;i<2*len;i++)
    printf("%c ",str[i]);
return 0;
}

```

例如, 我想制作自己的印章, 先输入 4 个印章上显示的汉字“向德生印”, 通过上面的程序就可以得到相应的编码(十六进制表示)。其中, “向”字的编码为 cff2, “德”字的编码为 b5c2, “生”字的编码为 c9fa, “印”字的编码为 d3a1。然后将这 4 个汉字的十六进制编码转换成对应的字符“c f f 2”、“b 5 c 2”、“c 9 f a”、“d 3 a 1”。执行效果如下图所示。



```

请输入一个或多个汉字: 向德生印
BM[0]=cf
BM[1]=f2
BM[2]=b5
BM[3]=c2
BM[4]=c9
BM[5]=fa
BM[6]=d3
BM[7]=a1
将编码转换成字符并显示:
c f f 2 b 5 c 2 c 9 f a d 3 a 1

```

当我们输入要生成印章的汉字后, 首先将编码转换成“字符”, 然后就可以利用这些字符在 [LiShu56.txt](#) 文件中查找匹配其点阵信息。原理就是: 利用这些字符, 在文本文件 LiShu56.txt 中搜索, 有匹配的编码, 就将其后的点阵信息读出并写入图像文件, 则可生成印章图片。

注意: 4 个汉字处在图片的不同位置, 在将各个汉字的点阵信息写入图像文件中时, 要通过适当的算法对各汉字的位置进行调整。