

**华侨大学 信息科学与工程学院 通信工程系**

**2023 级《面向对象程序设计》课程设计**

**——彭盛亮**

说明：

1. 五题中任选一题完成即可；
2. 每题均有三个任务，最低完成第一个任务，成绩优秀必须三个任务都完成；
3. 请在规定的时间内完成并验收；
4. 验收之后，撰写并提交设计报告；
5. 最终的成绩，根据平时实验、课后作业、课程设计验收、以及设计报告撰写情况综合评定。

1. BMP 图像处理

一个 BMP 图像文件包括四个区域，如下表所示。对于 24 位 BMP 图像，关键是图像宽度、图像高度、数据区偏移量、数据区大小四个参数，以及图像数据区域。

文件区域	变量名	地址偏移	大小	作用说明
文件头	bfType	0000h	2Bytes	文件标识符，必须为"BM"
	bfSize	0002h	4Bytes	整个图像文件的字节数
	bfReserved1	0006h	2Bytes	保留
	bfReserved2	0008h	2Bytes	保留
	bfOffBits	000Ah	4Bytes	文件开始到图像数据区的字节偏移量
信息头	biSize	000Eh	4Bytes	信息头区字节数
	biWidth	0012h	4Bytes	图像宽度（列数）
	biHeight	0016h	4Bytes	图像高度（行数）
	biPlanes	001Ah	2Bytes	位面数
	biBitCount	001Ch	2Bytes	一个像素点比特数（BMP24 图像为 24）
	biCompression	001Eh	4Bytes	图像数据压缩类型
	biSizelImage	0022h	4Bytes	图像数据字节数
	biXPelsPerMeter	0026h	4Bytes	水平方向每米像素数
	biYPelsPerMeter	002Ah	4Bytes	垂直方向每米像素数
	biClrUsed	002Eh	4Bytes	实际使用的调色板索引数
	biClrImportant	0032h	4Bytes	对显示有重要影响的颜色索引数
调色板		0036h	不定	BMP24 图像中无此区域
图像数据		bfOffBits	biSizelImage	BGR 数据按顺序排列

在图像数据区，每个像素点包含 3 个字节，分别为 BGR 值；图像所有像素点按照从左下到右上的顺序排列，最先是倒数第 0 行；每行须满足 4 字节对齐规则，如果其字节数不是 4 的整数倍，则补 0 使之满足。下图举例说明了宽度和高度均为 6 的 24 位 BMP 图像的数据区。

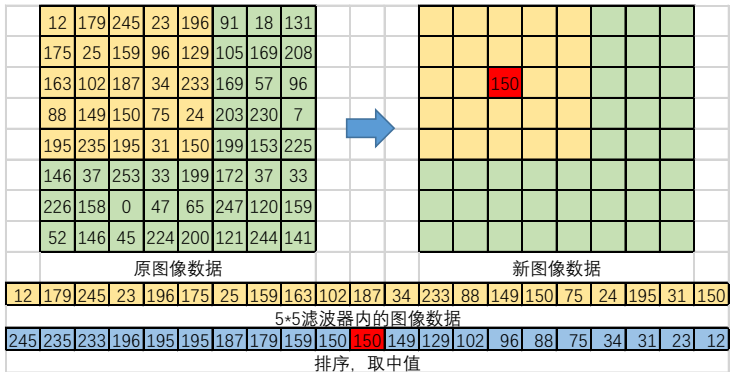
	第0列	第1列	第2列	第3列	第4列	第5列	补零
倒数第0行	B G R	B G R	B G R	B G R	B G R	B G R	
倒数第1行	B G R	B G R	B G R	B G R	B G R	B G R	
倒数第2行	B G R	B G R	B G R	B G R	B G R	B G R	
倒数第3行	B G R	B G R	B G R	B G R	B G R	B G R	
倒数第4行	B G R	B G R	B G R	B G R	B G R	B G R	
倒数第5行	B G R	B G R	B G R	B G R	B G R	B G R	
遵照4字节对齐规则，每行补2个字节，实际字节数为20							

推而广之, 对于高度为h宽度为m的 24 位 BMP 图像, 其每行的字节数为 num=ceil(m\*3/4)\*4。若数据区的首地址为 p，则倒数第 i 行、第 j 列像素点的地址为 p+i\*num+j\*3。

基于以上知识，开发一款 BMP 图像处理软件，具有简单的操作界面，能对 24 位 BMP 图像文件进行各种处理。

(1) 中值滤波:

中值滤波能够有效滤除图像中的椒盐噪声。利用以下 5\*5 滤波器，对图像进行中值滤波（以 5\*5 区域内 25 个数值的中位数，作为区域中心处的新值），输出滤波后的图像文件。



<a> 中值滤波原理



<b> 原图像



<c> 滤波后图像

(2) 图像缩小:

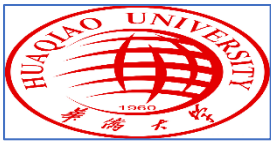
将图像的宽度或者高度缩小，输出缩小后的图像文件。



<a> 原图像



<b> 宽度缩小 1/2



<c> 高度缩小 1/2



<d> 宽高度均缩小 1/2

(3) 图像旋转:

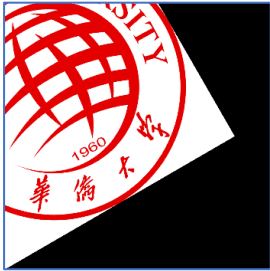
将图像以某一点为中心，旋转某个角度，输出旋转后的图像文件。



<a> 原图像



<b> 绕中心旋转 30°



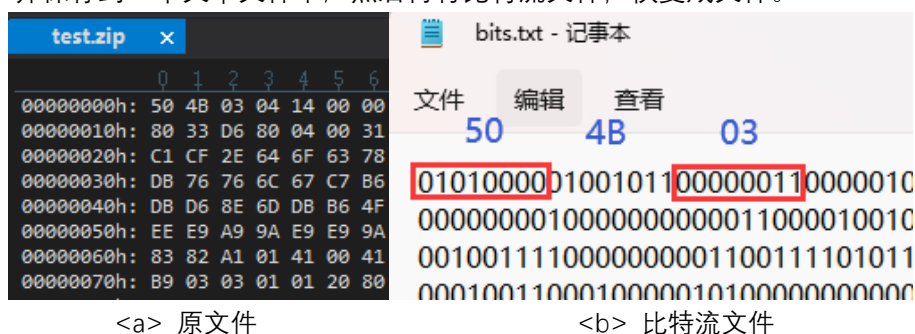
<c> 绕左下角旋转 30°

## 2. 通信传输流程

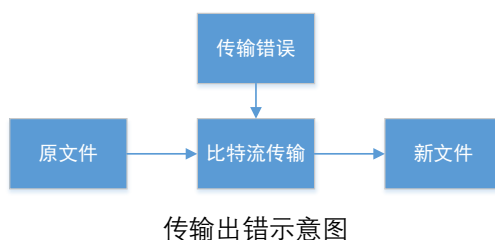
利用 C/C++ 编写一个软件，模拟通信中的数据传输流程，文件保存每一步的结果。

### (1) 比特流：

在通信过程中，数据以比特流的形式进行传输。读取一个要传输的文件，将其转换成比特流的形式，并保存到一个文本文件中；然后再将比特流文件，恢复成文件。

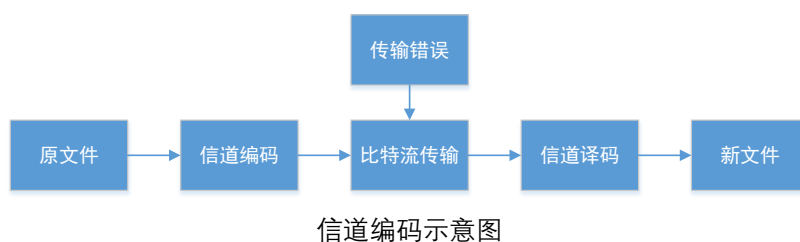


受到干扰和噪声的影响，比特流在传输时可能发生错误。手动修改比特流文件，将其中一些比特取反，来模拟传输出错。此时，恢复出的新文件是否能够正常打开？



### (2) 信道编码：

为了克服传输错误，可以在传输前对比特流进行信道编码，传输后再对比特流进行译码。



(7,4)汉明码是一种经典的信道编码，其编码和译码具体流程如下：

信道编码过程：

输入 4 个信息比特：a6 a5 a4 a3

计算 3 个冗余比特如下（xor 为异或）：

➤ a2 = a6 xor a5 xor a4;

➤ a1 = a6 xor a5 xor a3;

➤ a0 = a6 xor a4 xor a3;

输出 7 个比特：a6 a5 a4 a3 a2 a1 a0

例如：

对于比特流 01010000 ……：

1. 每 4 比特为一组

2. 0101 的编码结果为 0101101

0000 的编码结果为 0000000

3. 编码后的比特流为

01011010000000 ……

信道译码过程：

输入 7 个比特：a6 a5 a4 a3 a2 a1 a0

计算校验子 S2 S1 S0

➤  $S2 = a2 \text{ xor } a6 \text{ xor } a5 \text{ xor } a4$ ;

➤  $S1 = a1 \text{ xor } a6 \text{ xor } a5 \text{ xor } a3$ ;

➤  $S0 = a0 \text{ xor } a6 \text{ xor } a4 \text{ xor } a3$ ;

根据校验子判断是否有错（如下图）

➤ 如果有错码，则相应比特取反，进行纠错，再输出 a6 a5 a4 a3

➤ 否则直接输出 a6 a5 a4 a3

例如：

有错码的待译码比特流 01001010000000.....

1. 每 7 比特为一组

2. 0100101 的校验子为 011，

代表 a3 有错，a3 取反变 1，进行纠错，

纠错后的译码结果为 0101

3. 0000000 的校验子为 000，

代表无错误，无需纠错，

译码结果直接为 0000

4. 译码后的比特流：01010000.....

校验子 S2 S1 S0	0 0 0	0 0 1	0 1 0	1 0 0	0 1 1	1 0 1	1 1 0	1 1 1
错误位置	无错误	a0	a1	a2	a3	a4	a5	a6

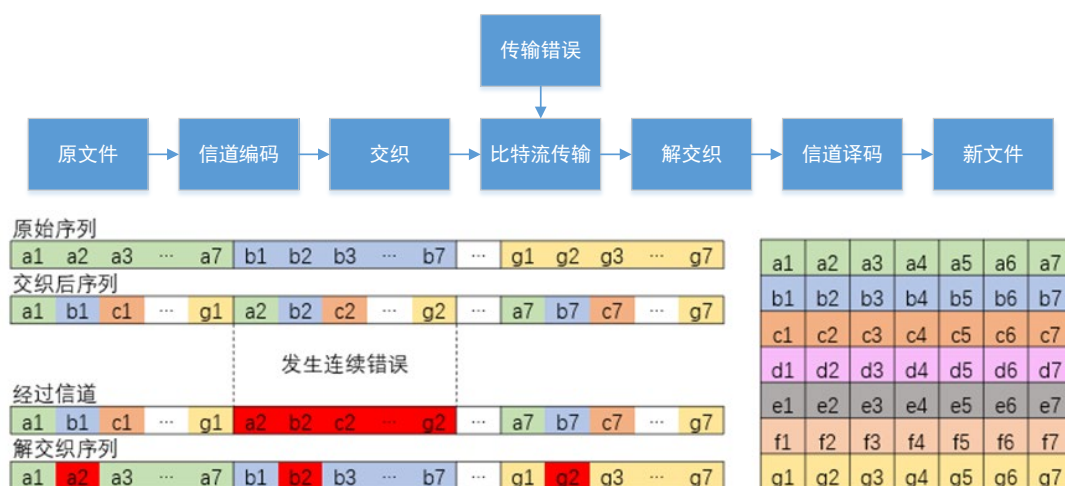
校验子含义

利用(7,4)汉明码对比特流进行信道编码和译码，将编码和译码后的比特流分别保存到不同的文本文件中。

在编码后的比特流文件，将 7 个比特中某 1 个比特取反，然后译码，最后恢复出的文件是否能够正常打开？如果 7 个比特中有多个取反呢？

### (3) 交织技术：

(7,4)汉明码最多能够纠正 7 个比特中的 1 个错误；当出现连续多个错误时，它就无能为力了。此时，可以利用交织技术，将连续的多个错误，打散到不同的 7 比特分组中，使得每个分组中的错误不超过 1 个，如下图所示。



利用 7\*7 交织器，对比特流进行交织和解交织，处理完的比特流存入不同的文本文件中。

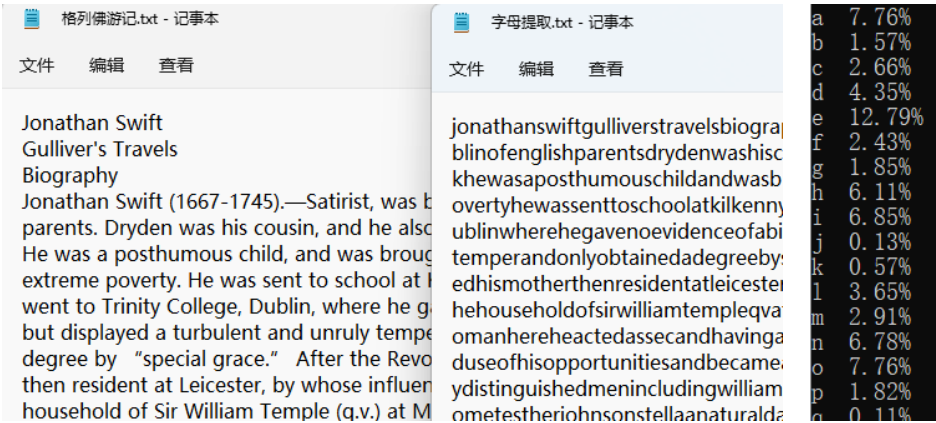
在交织后的比特流文件中，将连续不超过 7 个比特取反，然后解交织和解码，最后还原出文件的文件能正常打开吗？

3. 文本文件压缩

设计一款加密及压缩软件，具有简单的操作界面，可对文本文件进行加密和压缩。

(1) 字母提取：

给定一个文本格式的英文文档，将其中的全部字母转成小写并提取出来，存入另外一个文本文件中；并统计文中各个字母出现的概率。



(2) 等长压缩：

在通常的文本文件中，字符均按照 ASCII 码存储，占用一个字节（8 比特）空间。但在上述提取出文件中，全部都是小写字母，总共只有 26 种，理论上 5 比特就可以表示（5 位二进制最多表示 32 种不同的符号）。

请用 5 比特的二进制 00001-11010 分别表示字母 a-z，将上述只包含小字母的文件，压缩成一个新的二进制文件，文件的后缀名为自己姓名的首字母（例如张三的文件后缀名为“.zs”）。对比压缩后的文件与原文件的大小。

例如：

对于 jonathan……

1. 字母序号：10, 15, 14, 1, 20, 8, 1, 14……

2. 5 比特编码：'01010' '01111' '01110' '00001' '10100' '01000' '00001' '01110'……

3. 8 比特重组：'01010011' '11011100' '00011010' '00100000' '00101110'……

4. 按字节依次写入新的压缩文件

同时完成相应的解压缩功能。

(3) 变长压缩：

在上述定长压缩中，所有字母的编码长度（码长）都是 5 比特，没有考虑各个字母的差异。变长压缩的思想是，由于字母出现的概率不一样，可以用短码表示出现概率高的字母，用长码表示出现概率低的字母，这样可以使得所有字母的平均码长降低，达到压缩的目的。

Huffman 编码是一种经典的变长压缩编码。根据大数据统计出的字母出现概率，Huffman 编码中各个字母的编码结果见下表（字母 e 出现概率最高，码长最短为 3；字母 z 出现概率最低的，码长最长为 9）。

Huffman 编码结果

字符	出现概率	huffman 编码	码长
<b>e</b>	12.70%	011	3
<b>t</b>	9.06%	111	3
<b>a</b>	8.17%	0001	4
<b>o</b>	7.51%	0010	4
<b>i</b>	6.97%	0100	4
<b>n</b>	6.75%	0101	4
<b>s</b>	6.33%	1000	4
<b>h</b>	6.09%	1001	4
<b>r</b>	5.99%	1010	4
<b>d</b>	4.25%	00000	5
<b>l</b>	4.03%	00001	5
<b>c</b>	2.78%	10110	5
<b>u</b>	2.76%	10111	5
<b>m</b>	2.41%	11000	5
<b>w</b>	2.36%	11001	5
<b>f</b>	2.23%	11010	5
<b>g</b>	2.02%	001100	6
<b>y</b>	1.97%	001101	6
<b>p</b>	1.93%	001110	6
<b>b</b>	1.49%	001111	6
<b>v</b>	0.98%	110111	6
<b>k</b>	0.77%	1101100	7
<b>j</b>	0.15%	110110101	9
<b>x</b>	0.15%	110110100	9
<b>q</b>	0.10%	110110110	9
<b>z</b>	0.07%	110110111	9

利用以上编码结果，对上述只包含小字母的文件，进行压缩和解压。并比较压缩后的文件，与原文件、以及定长压缩文件的大小。

## 4. 推箱子游戏

设计一款推箱子游戏，控制小人将所有箱子推到目的地。



### (1) 基本功能:

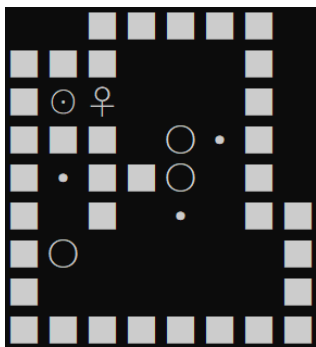
以“■”代表墙，以“♀”代表小人，以“.”代表目的地，以“○”代表箱子未到达目的地，以“⊙”代表箱子已到达目的地。注意：地图上的空位置有两种，空格为普通的空位置，“.”为箱子目的地的特殊空位置。

通过键盘，控制小人上下左右移动。

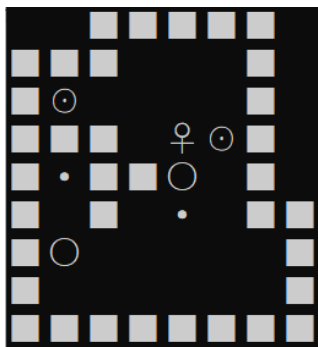
如果小人的下一位置为空位置，则小人可以进入下一位置。此时，原位置恢复成空格或者“.”，下一位置变为小人，同时更新小人位置。

如果小人的下一位置为箱子（“○”或“⊙”），且下下位置为空位置，则小人可以推动箱子。此时，原位置恢复成空格或者“.”，下一位置变为小人；下下位置变为箱子（“○”或“⊙”），同时更新小人以及箱子的位置。

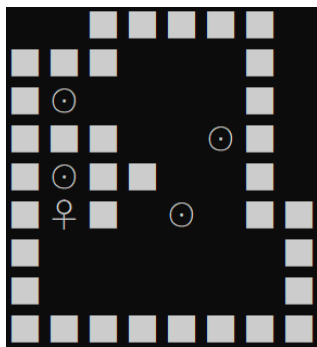
当所有箱子都处于目的地位置时，判定游戏成功。



<a> 游戏开始



<b> 游戏中

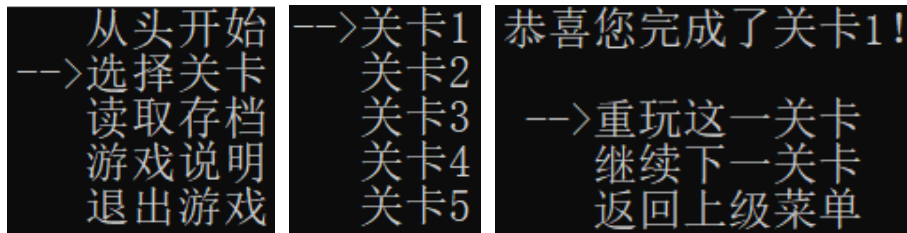


<c> 任务完成

### (2) 游戏关卡:

设计至少 5 个游戏关卡，每个关卡的地图及任务均不同。开始游戏时，可选择“从头开始”，即从第一个关卡开始；也可以“选择关卡”，然后进入子菜单，从特定的关卡开始。每个关卡任务完成之后，提示“重玩这一关卡”、“继续下一关卡”和“返回上级菜单”。





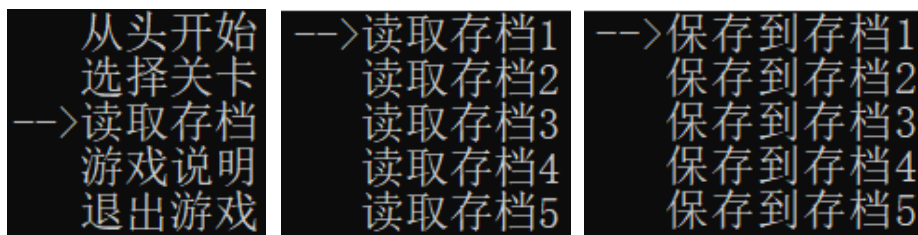
<a> 主菜单

<b> 选择关卡子菜单

<c> 关卡完成子菜单

### (3) 游戏存档:

设计至少 5 个游戏存档。在主菜单选择“读取存档”时，可进入读档子菜单，然后选择特定的存档继续游戏；游戏中，按下 m 或者 M 键，可进入存档子菜单，然后选择特定的存档保存游戏数据。



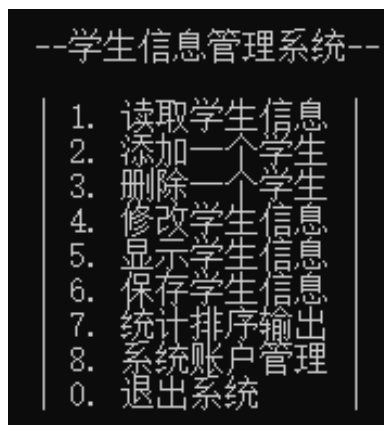
<a> 主菜单

<b> 读取存档子菜单

<c> 保存存档子菜单

## 5. 学生信息管理系统

设计一个学生信息管理系统, 每个学生的信息包括: 学号(id, 不超过 10 位数字)、姓名(name, 不超过 20 个字符)、年龄 (age, 整数)、性别 (gender, F 或 M)、成绩 (score, 浮点数) 和电话 (phone, 不超过 20 位数字), 系统以菜单方式工作, 例如:



系统功能菜单

具体功能要求如下:

### (1) 学生信息维护:

- 读取学生信息: 从文件中读取保存的学生信息;
- 添加一个学生: 要求学号要唯一, 如果添加了重复学号的记录时, 则提示数据重复并取消添加;
- 删除一个学生: 根据输入的学号删除学生信息; 如果学号不存在, 则提示错误;
- 修改学生信息: 首先输入学号, 再输入待修改的信息项, 然后输入新值; 如果学号或者信息项不存在, 则提示错误;
- 显示学生信息: 屏幕显示所有学生的信息;
- 保存学生信息: 将所有学生信息保存到文件。

### (2) 统计排序输出:

- 依次根据学号、年龄、性别以及成绩, 统计出所有满足条件的学生信息。例如, 对于学号条件, 首先提示“是否限定学号范围”; 如果是, 则输入学号的起始和终止范围; 如果否, 则跳过此条件;
- 统计之后, 提示“是否排序输出”。如果是, 则依次输入排序的关键字以及排序规则。例如, 首先输入“score descend”, 再输入“age ascend”, 意思是先按成绩从高到低排序, 成绩相同时再按年龄从小到大排序;
- 最后输出统计排序后的学生信息到文本文件。

### (3) 系统账户管理:

- 系统运行首先进入账户登录界面, 账号密码验证无误, 才能进入系统;
- 账户有管理员、教师和学生三类;

- c) 学生账户进入系统后，只有“读取学生信息”、“显示学生信息”和“统计排序输出”功能；
- d) 教师账户在学生账户的基础上，还有“添加一个学生”、“删除一个学生”、“修改学生信息”，“保存学生信息”功能；
- e) 管理员账户在教师账户的基础上，还有“系统账户管理”功能。每个账户的信息包括：用户名（username，不超过 20 个字符），密码（password，不超过 10 个字符）和账户类型（type，T 或者 S）；管理员账户可以读取、添加、删除、修改、显示和保存账户信息。管理员账户唯一，用户名固定为“admin”，初始密码“123456”，账户类型为 M。

用户登录

用户名：

密 码：

系统登录