**综合研讨课：（24学时）**

学生需利用课后时间完成综合设计题目，共10个，准备好相关的ppt以及程序，研讨课上随机抽取同学进行讲解，同时将自己编写的程序共享到群里，其他同学可以下载并在计算机上检查，每位同学至少演讲一次，同学讲解结束后进行讨论；演讲同学根据讲解质量和研讨情况给分，其他参与研讨的同学按照研讨参与情况给分，最后同学提交综合设计及研讨报告，综合上述表现，给出讨论课得分20分；

**讨论课内容和时间：**

1. 综合程序设计1-10
2. 共计约10个综合设计题目，确保每题目最多3学生讨论；
3. 每个学生讲解约20分钟，讨论约10分钟，每位同学参与必须讲解和讨论；
4. 学生利用课外上机时间

**讨论课过程**

1. 预先发布课程讨论题目，可在第1、2次课后全部发布讨论题目；
2. 学生通过QQ群等互动平台，与助教或教师确认自己讲解时间和内容；
3. 教师审阅学生的交流讨论程序、ppt等内容；
4. 上课时，学生讲解自己的程序设计思路，程序流程，实现方法，演示程序运行结果，并交流编程心得体会以及注意事项等内容；
5. 同学轮流提问并交流；
6. 教师点评并解答相关问题；
7. 助教全程关注每位同学的表现，并给出讨论课基本得分；
8. 每位同学根据互评表，给演讲的同学打分，满分100分。

## 综合程序设计一 学生成绩统计

**一、实验目的和任务：**

实验目的：

1. 练习结构、数组、指针、函数、文件等知识的综合运用；
2. 理解全局变量、局部变量、静态变量等内容。
3. 了解字符串函数的使用和预处理的作用。
4. 理解结构化程序设计方法，利用结构化程序设计方法，综合运用函数、数组、指针、结构等进行较复杂程序设计，能够实现一些简单的功能型程序。

实验任务：

1. 设计表示学生信息的结构；
2. 确定函数流程，清楚函数实现功能；
3. 熟悉文件的操作，编写函数从文件读取学生信息；
4. 编写统计函数，对读取的信息进行统计；
5. 编写输出函数，熟悉格式控制符的使用；
6. 设计测试文件，测试设计的结构和编写的函数；
7. 给出测试结果，程序评价。

**二、实验内容**

从文件中读取学生信息并打印。完成以下编程实验，调试运行程序并完成报告。

设每个学生包含如下信息：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **属性名** | 姓名 | 学号 | C成绩 | C实验成绩 | C总成绩 |
| **类型** | 字符串 | 字符串 | 整数 | 整数 | 整数（初始化为0） |

1. 设计一个表示学生信息的结构 struct student；
2. 编写函数从文件读入所有学生信息：
   1. **函数定义 int ReadStudentInfo**(const char\* filename, struct student\*\* pStudents); 函数返回值：返回学生个数，失败返回-1。输入参数const char\* filename：数据文件名；struct student\*\* pStudents：返回读取的学生信息，内存由ReadStudentInfo函数分配，main函数释放。
   2. **建议函数流程：**
      1. 打开数据文件，失败返回-1；
      2. 从头到尾扫描数据文件（即预读一遍）获得学生人数；
      3. 根据学生人数，利用malloc和sizeof等函数或运算符为struct student\*\* pStudents分配内存；
      4. 利用fseek函数，定位到文件头；
      5. 读取数据文件，并把每一条记录通过数组下标法或指针法保存到结构体数组pStudents中；
      6. 关闭文件；
      7. 返回学生人数。
   3. **数据文件格式说明：**

一行一条学生信息的记录，不同属性用空格隔开，输入数据中无总成绩记录。

1. **编写统计函数：**
   1. 函数定义void Count(struct student pStudents[], int num)。输入参数分别为保存学生的结构数组和学生人数。
   2. 功能为：统计所有学生的总成绩和每个班（根据学号判断班级）的平均成绩；输入参数包括学生信息结构数组，学生人数，计算的每个学生的总成绩直接填充到对应的输入结构数组对应字段中。统计信息包括每个班的总人数、平均成绩数组，用全局变量来表示。访问每个学生信息的时候，用指针来指向当前正在处理的学生。
   3. 设计一个宏，实现根据学号计算班级，并在函数Count中使用。
2. **编写信息输出函数：**

要求通过一定的格式控制符，使得输出美观。输出格式为：

|  |
| --- |
| 学生信息：  -----------------------------------------------------  序号 姓 名 学号班级C成绩 C实验成绩 总成绩  1 张三  2 ……  统计信息：  -----------------------------------------------------  序号 班级学生人数 平均成绩  1 1班 32 83  2 …… |

1. **编写main函数实现：**
   1. 定义变量，其中保存学生信息的数据文件直接在源代码中初始化；
   2. 读取学生信息，如果失败，打印失败信息并退出程序；
   3. 统计学生信息；
   4. 输出统计信息。
   5. 释放内存。

## 综合程序设计二 因式分解与完美数

**一、实验目的和任务：**

目的：掌握C语言综合编程技巧，运用C语言解决相关问题。

任务：

1. 熟悉C程序开发环境，熟悉C程序的编辑、编译、连接和运行的过程。
2. 掌握C语言综合程序设计过程，学习分析问题并将其用程序的方式进行转化。
3. 学会综合运用C语言中的各种知识完成程序设计。
4. 掌握较复杂程序的编译调试。
5. 掌握循环、函数调用、数组、指针等知识。
6. 学习通过算法优化来降低程序运算量。

**二、实验内容**

上机完成以下编程实验，调试运行程序并完成报告。

1. 输入一个数，使用质数表对其进行因式分解。

注：因式分解基本上就是使用小于输入数的数值当作除数，去除以输入数值，如果可以整除就视为因数，效率更高的解法是使用质数表的方法，该方法就是先求出小于该数的所有质数，并试试看是不是可以整除，从而完成因式分解。

1. 求10000以内的所有完美数，要求程序要有较高的运算效率。

说明：如果有一数n，其真因数（Properfactor）的总和等于n，则称之为完美数（PerfectNumber），例如以下几个数都是完美数：

6 = 1 + 2 + 3

28 = 1 + 2 + 4 + 7 + 14

496 = 1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248

程序基本上不难，第一眼看到时会想到使用循环求出所有真因数，再进一步求因数和，不过若n值很大，则此法会花费许多时间在循环测试上，十分没有效率，比如这个问题中，学生可编简单程序试试循环的运行效率。

提示：这个问题可利用前一个问题中的质数表来提高效率。

## 综合程序设计三 分析扑克牌

**一、实验目的和任务：**

目的：掌握C语言综合编程技巧，运用C语言解决相关问题。

任务：

1. 熟悉C程序开发环境，熟悉C程序的编辑、编译、连接和运行的过程。
2. 掌握C语言综合程序设计过程，学习分析问题并将其用程序的方式进行转化。
3. 学会综合运用C语言中的各种知识完成程序设计。
4. 掌握较复杂程序的编译调试。
5. 掌握循环、函数调用、数组、指针等知识。
6. 学习通过算法优化来降低程序运算量。

**二、实验内容**

上机完成以下编程实验，调试运行程序并完成报告。

1. 编写函数newcard，利用随机数产生函数任意发出5张牌（不包括大王和小王），必须满足在一副牌里的花色和点数，也就是说花色只能是：“红heart、黑spade、梅club、方diamond”四种，点数只能为：1~13；需要设计全局数据结构存放这5张牌，要求尽可能简洁。
2. 编写函数analyze，分析5张牌的情况，包括7种情况：同花顺flush（5张点数连续、花色相同的牌）、顺子straight（5张点数连续、花色不同的牌）、四炸four（有4张点数相同、花色不同的牌）、三拖二full\_house（5张牌中有3张点数相同和1个对子）、三炸three（3张点数相同、花色不同的牌）、两对two\_paires（有2个对子）、一对pair（有1个对子）；分析结果以英文字符串存放，以字符指针返回。
3. 编写main函数，先调用newcard发牌，再调用analyze分析，然后输出分析结果。设计简单的交互，使程序运行时可以按输入多次运行或结束运行。

## 综合程序设计四 字符串处理库

**一、实验目的和任务：**

目的：掌握C语言综合编程技巧，运用C语言解决相关问题。

任务：

1. 熟悉C程序开发环境，熟悉C程序的编辑、编译、连接和运行的过程。
2. 掌握C语言综合程序设计过程，学习分析问题并将其用程序的方式进行转化。
3. 学会综合运用C语言中的各种知识完成程序设计。
4. 掌握较复杂程序的编译调试。
5. 掌握循环、函数调用、数组、指针等知识。
6. 学习通过算法优化来降低程序运算量。

**二、实验内容**

上机完成以下编程实验，调试运行程序并完成报告。

1. 编写自己的字符串处理函数库，包括以下函数：计算字符串长度，比较两个字符串大小，字符串拷贝，字符串连接，查找子串（找到返回子串首字母的序号，序号从1开始；没找到返回0）。

注意库函数的健壮性，对不正确参数的适应性（例如拷贝字符串时，如果目的字符数组长度不够如何处理）等；要提供该函数库的头文件和使用说明。

1. 编写main函数，设计测试代码，分别调用各函数测试功能。设计简单的交互，使程序运行时可以按输入多次运行或结束运行。

## 综合程序设计五 文件比较器

**一、实验目的和任务：**

目的：掌握C语言综合编程技巧，运用C语言解决相关问题。

任务：

1. 熟悉C程序开发环境，熟悉C程序的编辑、编译、连接和运行的过程。
2. 掌握C语言综合程序设计过程，学习分析问题并将其用程序的方式进行转化。
3. 学会综合运用C语言中的各种知识完成程序设计。
4. 掌握较复杂程序的编译调试。
5. 掌握循环、函数调用、数组、指针等知识。
6. 学习通过算法优化来降低程序运算量。

**二、实验内容**

上机完成以下编程实验，调试运行程序并完成报告。

1. 编写blank函数，实现将一行字符中的连续多个空格改成1个空格的功能。
2. 编写main函数，读入两个文本文件，逐行比较是否相同。注意比较每行时需要先调用blank函数去掉多余空格，再进行比较。

如果两文件相同，输出ok；如果不同，输出所有不相同的行的行号和该行第一个不相同的字符的列号，假定行号和列号都从1开始编号。将比较结果同时输出到屏幕上和一个结果文件里。

## 综合程序设计六 大奖赛评分系统

**一、实验目的和任务：**

目的：掌握C语言综合编程技巧，运用C语言解决相关问题。

任务：

1. 熟悉C程序开发环境，熟悉C程序的编辑、编译、连接和运行的过程。
2. 掌握C语言综合程序设计过程，学习分析问题并将其用程序的方式进行转化。
3. 学会综合运用C语言中的各种知识完成程序设计。
4. 掌握较复杂程序的编译调试。
5. 掌握循环、函数调用、数组、指针等知识。
6. 学习通过算法优化来降低程序运算量。

**二、实验内容**

上机完成以下编程实验，调试运行程序并完成报告。

请为某大奖赛编写一个现场统分程序，在各评委打分后，及时通报评分结果。同时为了给评委一个约束，有利于竞赛评判的公正，要求增加给评委打分和排序的功能。

已知：某大奖赛有n个选手参赛，m（m>2）个评委依次为参赛的选手评判打分，最高10分，最低0分。统分规则为：在每个选手所得的m个得分中，去掉一个最高分，去掉一个最低分后，取平均分为该选手的最后得分，要求：

1. 根据n个选手的最后得分，从高到底排出名次表，以便确定获奖名单；
2. 根据各选手的最后得分与各评委给该选手所评分的差距，对每个评委评分的准确性给出一个定量的评价。
3. 编写自动测试代码，尽量测试各种情况，测试数据和结果分别存入文件备查。

## 综合程序设计七 保龄球游戏

**一、实验目的和任务：**

目的：掌握C语言综合编程技巧，运用C语言解决相关问题。

任务：

1. 熟悉C程序开发环境，熟悉C程序的编辑、编译、连接和运行的过程。
2. 掌握C语言综合程序设计过程，学习分析问题并将其用程序的方式进行转化。
3. 学会综合运用C语言中的各种知识完成程序设计。
4. 掌握较复杂程序的编译调试。
5. 掌握循环、函数调用、数组、指针等知识。
6. 学习通过算法优化来降低程序运算量。

**二、实验内容**

上机完成以下编程实验，调试运行程序并完成报告。

编写一个保龄球计分的程序，具体要求如下：

1. 保龄球一场共10局，每一局最多打两球，至少打一球。
2. 每一局共有10个瓶子，击倒一个瓶子得1分。若第1局到第9局的第一个球击倒10个 瓶子（全倒），则跳入下一局，不需要打第2个球。
3. 若第10局的第1个球击倒10个瓶子，或者第10局的2个球加起来击倒10个瓶子（补 中），则第10局可以打第3个球。
4. 保龄球10局最少打11个球，最多打21个球。
5. 计分方式若该局中第1球击倒10个瓶子，则本局分数加计下局的两球的分数；若一局 中第1球加第2球共击倒10个球则本局分数会加计下局第1球的分数；若一局2球没有击倒10个瓶子，该局只计击倒瓶子的分数。
6. 扩展要求（不要求，可选做）：可再编写一个玩家进行保龄球击打的模拟程序，规则可自行开脑洞设计，并配合计分程序完成保龄球游戏过程。

## 综合程序设计八 国际象棋

**一、实验目的和任务：**

目的：掌握C语言综合编程技巧，运用C语言解决相关问题。

任务：

1. 熟悉C程序开发环境，熟悉C程序的编辑、编译、连接和运行的过程。
2. 掌握C语言综合程序设计过程，学习分析问题并将其用程序的方式进行转化。
3. 学会综合运用C语言中的各种知识完成程序设计。
4. 掌握较复杂程序的编译调试。
5. 掌握循环、函数调用、数组、指针等知识。
6. 学习通过算法优化来降低程序运算量。

**二、实验内容**

上机完成以下编程实验，调试运行程序并完成报告。

在一个8\*8的国际象棋棋盘上放置8个皇后，放置要求：

1. 每个皇后两两之间不“冲突”，即没有一个皇后能“吃掉”任何一个其他皇后，简单说就是没有任何两个皇后占据棋盘上的同一行或同一列或同一对角线，即在每一横列、竖列、斜列都只有一个皇后。
2. 运行时任意输入一个位置，输出这种情况下的皇后放置情况；
3. 存储8\*8棋盘上所有可能的组合情况，输出可能情况的总数；
4. 对棋盘的大小进行扩展，皇后冲突规则不变，输出扩展后的可能情况总数。

## 综合程序设计九 矩阵运算1

**一、实验目的和任务：**

实验目的：

1. 练习结构、数组、指针、函数等知识的综合运用；
2. 培养学生综合运用C程序语言进行程序设计的能力；
3. 培养结构化程序设计能力；

实验任务：

1. 分析矩阵的属性和常规操作；
2. 定义矩阵的数据结构；
3. 确定矩阵的基本操作，画出程序结构图，给出函数原型和函数功能描述；
4. 编写矩阵的类型定义、初始化、以及各个函数模块；
5. 编写主控模块；
6. 设计测试数据，并调试这些函数和结构；
7. 给出测试结果，程序评价。

**二、实验内容**

上机完成以下编程实验，调试运行程序并完成报告。

1. 编写矩阵定义、初始化函数；
2. 编写矩阵加法函数；
3. 编写矩阵减法函数；
4. 编写矩阵乘法函数；
5. 编写求矩阵均值函数；
6. 编写求一个矩阵的子阵函数；
7. 编写矩阵输出函数；
8. 编写主控函数；

主要函数和结构参考原型如下：

typedefstruct{

double\*\*mat;

introw;

intcol;

}Matrix;

voidInitialMatrix(Matrix\*T,introw,intcol);//只分配空间不初始化；

voidInitialMatrixZero(Matrix\*T,introw,intcol);//初始化为0

voidInitialMatrixRand(Matrix \*T,int row, int col); //初始化为50以内随机正整数

voidInputMatrix(Matrix\*T); //键盘输入矩阵

voidDestroyMatrix(Matrix\*T); //释放矩阵空间

voidPrintfMatrix(Matrix\*T); //矩阵输出

intAddMatrix(Matrix\*A,Matrix\*B,Matrix\*C);//矩阵加

intMinusMatrix(Matrix\*A,Matrix\*B,Matrix\*C); //矩阵减

intMultiMatrix(Matrix\*A,Matrix\*B,Matrix\*C); //矩阵乘法

doubleMeanMatrix(Matrix\*T); //矩阵元素均值

intSubMatrix(Matrix\*T1,Matrix\*T2,intBeginRow,intBeginCol,intEndRow,intEndCol); //求T1的子矩阵T2;

void PrintMatrix(Matrix\*T); //矩阵输出

测试程序参考界面如下所示：

矩阵函数测试，请选择功能，输入对应的数字：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1：输入一个矩阵，求矩阵均值；

2：产生一个随机数矩阵，求矩阵均值；

3：输入两个个矩阵，求矩阵和；

4：输入两个个矩阵，求矩阵差；

5：输入两个个矩阵，求矩阵积；

6：产生两个个随机数矩阵，求矩阵和；

7：产生两个个随机数矩阵，求矩阵差；

8：产生两个个随机数矩阵，求矩阵积；

9：求矩阵的子阵，如矩阵的2-4行，1-3列的子阵；

0：结束!

## 综合程序设计十 矩阵运算2

**一、实验目的和任务：**

实验目的：

1. 练习结构、数组、指针、函数等知识的综合运用；
2. 培养学生综合运用C程序语言进行程序设计的能力；
3. 培养结构化程序设计能力；

实验任务：

1. 分析矩阵的属性和常规操作；
2. 定义矩阵的数据结构；
3. 确定矩阵的基本操作，画出程序结构图，给出函数原型和函数功能描述；
4. 编写矩阵的类型定义、初始化、以及各个函数模块；
5. 编写主控模块；
6. 设计测试数据，并调试这些函数和结构；
7. 给出测试结果，程序评价。

**二、实验内容**

上机完成以下编程实验，调试运行程序并完成报告。

1. 编写矩阵定义、初始化函数；
2. 编写矩阵输出函数；
3. 编写求矩阵最大值函数；
4. 编写矩阵元素归一化函数；
5. 编写矩阵转置函数；
6. 编写矩阵求逆函数；（注意要判断逆是否存在）
7. 编写主控函数；

主要函数和结构参考原型如下：

typedefstruct{

double\*\*mat;

introw;

intcol;

}Matrix;

voidInitialMatrix(Matrix\*T,introw,intcol);//只分配空间不初始化；

voidInitialMatrixZero(Matrix\*T,introw,intcol);//初始化为0

voidInitialMatrixRand(Matrix \*T,int row, int col); //初始化为50以内随机正整数

voidInputMatrix(Matrix\*T); //键盘输入矩阵

voidDestroyMatrix(Matrix\*T); //释放矩阵空间

voidPrintfMatrix(Matrix\*T); //矩阵输出

intMaxMatrix(Matrix\*T); //矩阵元素最大值

void TransMatrix(Matrix\*T); //矩阵转置

void NormMatrix(Matrix\*T); //矩阵归一化

void ReverseMatrix(Matrix\*A, Matrix\*B); //B是A的逆矩阵

测试程序参考界面如下所示：

矩阵函数测试，请选择功能，输入对应的数字：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1：输入一个矩阵，求矩阵元素的最大值；

2：产生一个随机数矩阵，求矩阵元素的最大值；

3：输入一个矩阵，求矩阵转置；

4：产生一个随机数矩阵，求矩阵转置；

3：输入一个矩阵，求矩阵归一化；

4：产生一个随机数矩阵，求矩阵归一化；

3：输入一个矩阵，求矩阵的逆；

4：产生一个随机数矩阵，求矩阵的逆；

0：结束!

## 综合程序设计十一 滤波器设计和使用

**一、实验目的和任务：**

实验目的：

1. 练习随机数、静态局部变量、滤波器等知识的综合运用；
2. 培养学生综合运用C程序语言进行程序设计的能力；
3. 培养结构化程序设计能力；

实验任务：

实验内容：

1. 完成随机数产生函数：double RandDouble(double Max)，即生产0到Max之间的随机数，且返回的随机数带2位小数，如6.28等；
2. 编写一函数，double filter(double x)；实现如下功能：输入*xi*，输出*yi，*其中*xi*的定义如5-1所示，即*i*大于0时，为随机数，小于0时为0，*i*为整数；*xi*和*yi*的关系如5-2所示。

(5-1)

(5-2)

c0,c1,c2,c3,c4的值分别为常数：0.01，0.1, 0.78, 0.1, 0.01，请用宏定义c0到c4的值；rand()为随机数函数，产生随机数整数；

1. 写一主函数，在主函数中调用随机数生成函数，产生序列x，并求出序列y的值，输出x和y的值。
2. 设Cn为某低通滤波器系数，Cn的值见后。改写double filter(double x)函数，实现滤波功能。编写主函数，读取一个被噪声污染的语音文件，并进行滤波处理，把处理后的文件保存下来，并进行试听，观察处理效果。

CN[101] = {-0.004808919657970,

-0.001547991286813,

-0.001484119075616,

-0.001180626535065,

-0.000638610828712,

0.000104137685310,

0.000976202943056,

0.001875083276187,

0.002678747868229,

0.003258340856584,

0.003497633328625,

0.003310022925111,

0.002656136525684,

0.001555093457088,

0.000095231399231,

-0.001579126147775,

-0.003267487272886,

-0.004745069573984,

-0.005784364014156,

-0.006183187126704,

-0.005799711899067,

-0.004575941863646,

-0.002560489731262,

0.000086215276100,

0.003095941733918,

0.006123608643356,

0.008773450587629,

0.010645725379523,

0.011369883796541,

0.010707969201656,

0.008534221934017,

0.004906086245703,

0.000084923610347,

-0.005500640536805,

-0.011244701868107,

-0.016442739558196,

-0.020332177216741,

-0.022189237562184,

-0.021388750352397,

-0.017496131711919,

-0.010310727721149,

0.000074954684212,

0.013283855664295,

0.028642402846373,

0.045253402410963,

0.062043759242903,

0.077862333668155,

0.091580983706917,

0.102191837897796,

0.108897201849955,

0.111190826178577,

0.108897201849955,

0.102191837897796,

0.091580983706917,

0.077862333668155,

0.062043759242903,

0.045253402410963,

0.028642402846373,

0.013283855664295,

0.000074954684212,

-0.010310727721149,

-0.017496131711919,

-0.021388750352397,

-0.022189237562184,

-0.020332177216741,

-0.016442739558196,

-0.011244701868107,

-0.005500640536805,

0.000084923610347,

0.004906086245703,

0.008534221934017,

0.010707969201656,

0.011369883796541,

0.010645725379523,

0.008773450587629,

0.006123608643356,

0.003095941733918,

0.000086215276100,

-0.002560489731262,

-0.004575941863646,

-0.005799711899067,

-0.006183187126704,

-0.005784364014156,

-0.004745069573984,

-0.003267487272886,

-0.001579126147775,

0.000095231399231,

0.001555093457088,

0.002656136525684,

0.003310022925111,

0.003497633328625,

0.003258340856584,

0.002678747868229,

0.001875083276187,

0.000976202943056,

0.000104137685310,

-0.000638610828712,

-0.001180626535065,

-0.001484119075616,

-0.001547991286813,

-0.004808919657970

};

提示：

wav文件格式的为文件头+数据，其中文件头的定义如下：

typedef struct \_tagMsWavPcmHeader44{

unsigned char ChunkID[4]; // "RIFF"; The "RIFF" the mainchunk;

unsigned long ChunkSize; // FileSize - 8; The size following this data

unsigned char Format[4];

// "WAVE"; The "WAVE" format consists of two subchunks: "fmt " and "data"

unsigned char SubChunk1ID[4]; // "fmt "

unsigned long SubChunk1Size;

// 16 for PCM. This is the size of the rest of the subchunk which follows this data.

unsigned short AudioFormat; // 1 for PCM. Linear quantization

unsigned short NumChannels; // 1->Mono, 2->stereo, etc..

unsigned long SampleRate; // 8000, 11025, 16000, 44100, 48000, etc..

unsigned long ByteRate; // = SampleRate \* NumChannels \* BitsPerSample/8

unsigned short BlockAlign; // = NumChannels \* BitsPerSample / 8

unsigned short BitsPerSample; // 8->8bits, 16->16bits, etc..

unsigned char SubChunk2ID[4]; // "data"

unsigned long SubChun2Size;

// = NumSamples \* NumChannels \* BitsPerSample / 8. The size of data

} wav\_pcm\_header44;