

千锋教育 C 语言大纲

- 大纲以问题方式呈现。它不单单是知识点的简单堆砌，更重要的是它还能体现出讲师的授课思路以及知识的结构脉络，希望大家能细细品味。
- 希望大家能提前预习，做到带着问题听课。
- 如果你能正确回答出本书上的所有问题，你几乎可以应对所有的 C 语言理论面试！

内部资料 版权所有

肺腑之言

我们的教学理念

我们认为衡量基础语言教学成败的唯一标准是学员能够真正独立的编写出程序来！

我们的授课模式

几乎每天的授课都由理论和实战两个部分组成。理论主要集中于语法的讲解，理论授课会很紧凑，有时会让部分同学感觉到吃力，但理论难学不意味着我要花大量的时间去学习它。因为当你觉得不好理解时，不是因为它有多么难，而是不知道它有什么用以及如何使用它。真正事半功倍的学习方式是尝试着使用它们。所以我们每天的编程实战时间会很多。

希望大家搞清楚的

“语言学习”不是“语法和理论学习”，我们的教学不会让学生沉浸在晦涩的语法和艰深的理论中。我们会将更多的时间放在教会大家如何写程序上。

希望大家能做到的

动起来！多写多实践，充分调动你的双手和大脑。想学会编程，只靠眼睛和耳朵的静态学习模式是根本不行的。

克服惰性和浮躁！很多人的努力程度之低还远没有到拼天赋的地步。一时的偷懒或者懈怠给你带来的只不过是片刻的并不持久的“欢愉”。但是代价可能是无法跟上正常上课节奏的压力和痛苦，而这种痛苦是会累计的，直到将你的信心磨灭。所以一定要自制和自律。学习的过程其实也是在“修心”，而修炼时的最大的敌人就是浮躁。希望在这个阶段，大家能让自己沉静下来，养成勤奋踏实而又不浮躁的态度，克服自身惰性养成良好的学习习惯。

充分利用时间但注意劳逸结合！编程技能的培养是需要时间保证的。根据目前统计，在学习日，学员平均只有 2 个小时左右的独立编程时间，整个阶段下来也只不过 30 个小时，对于学习编程来讲这是不够的。学员应该争取每个学习日有 3 个小时的编程时间，周末两天保证有 12 个小时的练习时间。如果本阶段能保证有接近 80 个小时的编程练习的话，对于绝大部分人来讲，编程入门是没有问题的。但是过犹不及，不应该随意加大学习量。如果每天晚上都学习到很晚才睡觉的话，那么肯定会因睡眠不足而影响第二天的学习。每个人都应根据自身的情况制定学习目标，不要因为追求达不到的目标而牺牲休息时间。

不要怪我们“不讲理”

每个人在幼儿时期，都需要父母几乎 24 小时的照顾和“约束”。基础语言学习阶段犹如人的幼儿时期，也需要教学相关人员对大家更加细致的照顾。因此尽管大家都是现实生活中的成人，但又好像是程序开发世界中的“幼儿”。如果要想成为一名合格的程序员，就需要接受来自教学人员的各种“刁难”，这可能包括晚上或者周末的强制学习、罚抄等“限制人身自由”或接近“体罚”的教学方式。总之，这个阶段不要把自己当作一个有自尊的“人”。生活很现实，如果要想得到别人的尊重，你要有让别人尊敬你的资本。我们这么做的目的也是希望你拥有将来作为程序开发人员让别人尊敬的资本。

培养终身受用的好习惯

树如果一开始就长歪了，那再正过来的代价将会非常大。同样，错误的理念会使初学者陷入并迷失在泥潭中不能自拔，坏习惯也将会是你一路的绊脚石。而不幸的是，对于绝大部分人来讲，这两者远比其正确的方式更容易被人学会和接受。所以初学者应该多听一听老师的建议并在学习初始阶段就树立起正确的学习观念，努力克服惰性和浮躁并养成良好的习惯，将好的东西持之以恒下去，只有这样才能越学越有信心，最终到达成功的彼岸！

Day1-熟悉 Mac 程序开发环境

一 Mac 操作系统

1.1 Mac 常识

Mac 是一种类 Unix 的图形化界面操作系统，有两种开发环境：图形化 IDE 和终端 Terminal。

1.2 常用快捷键

- a) **Command+Tab**: 任意情况下切换应用程序
- b) **Command+~**: 同一应用程序多窗口间切换
- c) **Command+F**: 呼出大部分应用程序的查询功能
- d) **Command+C/V/X**: 复制/粘贴/剪切
- e) **Command+N**: 新建应用程序窗口
- f) **Command+Q**: 退出当前应用程序
- g) **Control+Space**: 呼出 Spotlight，右上角的查询图标。
- h) **Command+Space**: 切换输入法

二 常用终端命令

2.1 有关文件和目录的基本概念

- a) 目录层次结构
Mac 系统没有分区概念，整个计算机文件系统从根目录 `/` 开始，由一层层的目录 (windows 里的文件夹) 组成，文件根据功能安装在不同的目录里。整个系统目录结构从根目录 `/` 开始，形成一个倒树形结构。
- b) home 目录
每个 Mac 用户都有一个 **home** 目录，该用户的所有操作，例如创建的个人目录和文件，通常放在该目录里。通常打开终端 Terminal 后，自动进入当前用户的 **home** 目录。
- c) 当前目录
当前用户执行操作的工作目录，当前目录可以通过 **cd** 命令改变，可以通过 **pwd** 命令显示当前目录的绝对路径。当前目录可以用标识符 `.` 引用。
- d) 父目录
除根目录 `/` 外，每个目录都有一个上级目录，通常称为父目录。父目录可以用标识符 `..` 引用。
- e) 绝对路径
绝对路径是从根目录 `/` 开始，计算的某一文件/文件夹的路径。
- f) 相对路径
相对路径是从当前路径开始，计算的某一文件/文件夹的路径。

2.2 常用命令

使用规范：命令名 [命令选项] [命令行参数]

- a) 显示当前目录的绝对路径
 - 1) 命令名称：pwd
 - 2) 使用方式：pwd
- b) 进入某一目录的命令
 - 1) 命令名称：cd
 - 2) 使用方式：cd dst_dir, dst_dir 可以省略，进入当前用户 home 目录。

- c) 显示某一目录内容
 - 1) 命令名称: `ls`
 - 2) 使用方式: `ls dst_dir`, `dst_dir` 可以省略, 显示当前目录内容, 不包括隐藏文件, 如果要显示隐藏文件, 使用 `ls -al`。
- d) 创建一个文件
 - 1) 命令名称: `touch`
 - 2) 使用方式: `touch new_file`。
- e) 复制文件或目录
 - 1) 命令名称: `cp`
 - 2) 使用方式: `cp src dst`
- f) 创建一个新目录
 - 1) 命令名称: `mkdir`
 - 2) 使用方式: `mkdir new_dir`, 在当前目录创建新目录 `new_dir`。
- g) 删除文件或目录
 - 1) 命令名称: `rm`
 - 2) 使用方式: `rm files_or_dir`, 通常使用参数 `-rf`。
- h) 移动一个文件或目录
 - 1) 命令名称: `mv`
 - 2) 使用方式: `mv src dst`, 把文件或目录 `src`, 移动到新目录 `dst`。

2.3 作业:

- (1) 显示当前目录里的文件
- (2) 进入当前用户 `home` 目录, 创建一个新目录, 进入该新目录, 在该目录创建一个新目录和二个新文件。
- (3) 接着习题(2), 返回用户 `home` 目录, 把(2)中创建的目录复制一份放在当前目录, 然后把旧目录移动到 `/tmp` 目录。
- (4) 进入 `/tmp` 目录, 删除(3)中我们移动过来的目录。

三 使用 vim 编辑器

`vim` 的三类模式: 正常模式、编辑模式和可视模式。

正常模式中可以执行命令, 默认情况下, `vim` 启动后会进入这个模式。

编辑模式中可以插入和编辑文本。

可视模式中可以可视化地选择一段文本, 然后仅针对这段文本执行命令或操作。

通常在正常模式和编辑模式间交替使用 `vim`。在其他模式里, 可以按下 `ESC` 键切换到正常模式。

进入 `vim` 程序: `vim`, 可以在编辑模式下输入文本, 并保存成新文件。

使用 `vim` 打开一个文件: `vim filename`。

3.1 正常模式下的常用命令

3.1.1 转换到编辑模式

"`i`"命令可以在当前光标之前插入文本。

"`a`"命令可以在当前光标之后插入文本。

"`o`"命令可以在当前行的下面另起一行, 并使当前模式转为插入模式。

"`O`"命令(注意是大写的字母 `O`)将在当前行的上面另起一行。

3.1.2 删除

要删除一个字符, 只需要将光标移到该字符上按下"`x`"。

删除一整行内容使用"dd"命令，删除后下面的行会移上来填补空缺。

3.1.3 撤销

如果你误删了过多的内容，可以使用命令"u"撤销上一次的操作。

3.1.4 保存和退出

使用":w"保存已编辑的内容。

丢弃所有的修改并退出，用命令":q!"。

使用":wq"保存已编辑内容然后退出。

用":e!"命令放弃所有修改并重新载入该文件的原始内容。

3.1.5 光标移动

可以使用方向键移动光标，也可以使用下面的快捷方式。

"\$"将光标移动到当前行行尾。

"^"将光标移动到当前行的第一个非空白字符上。

"G"将光标定位到最后一行上。

"gg"将光标移动到第一行。

":number"将光标移动到第 number 行。

3.1.6 行号

"set number"在每行的前面显示一个行号。

":set nonumber"关闭显示行号。

3.1.7 简单搜索

"/string"命令可用于搜索一个字符串。要查找上次查找的字符串的下一个位置,使用"n"命令。

3.2 复制粘贴

复制或者剪切的行是从光标所在的行开始。粘贴的位置是从光标所在的下一行开始。

复制一行: yy 复制多行: xyy(x 表示数量, 比如 3yy)

剪切一行: dd 剪切多行: xdd(x 表示数量, 比如 3dd)

粘贴: p

四 gcc 编译器简单使用

(1) gcc 是常用的 C 语言编译链接工具，能够从 C 源码文件生成可执行程序。

假如有 C 源代码文件 helloworld.c，我们可以通过下述命令生成可执行程序：

```
gcc -o helloworld helloworld.c
```

生成的可执行程序名为 helloworld，可以通过下述命令执行：

```
./helloworld
```

Day2-数据和操作

一 了解计算机和计算机语言

1.1 简述计算机的基本工作原理。

1.2 你如何理解计算机语言？

1.3 什么是程序，它的本质？

二 一切都是数据

2.1 怎么理解程序的本质就是处理各种数据？你又如何理解代码也是“数据”的？

2.2 计算机中数据是哪种进制存储的？为什么用该进制？

2.3 掌握进制转换原理，试着将 885 转换成二进制，再将 1010110 转换成十进制。

2.4 了解八进制，试着将八进制 072 转换成十进制和二进制。

2.5 掌握十六进制和二进制的转换以及与十进制之间的相互转换，试着将 0xCC88 转换成二进制和十进制。

2.6 二进制如何快速转换成十六进制？十六进制的优势是什么？

2.7 掌握正负数的原码 / 反码 / 补码，将十六进制的 -0x100 的二进制原码 / 反码 / 补码写出来。

2.8 程序中的数据最终都会被翻译成二进制数据，这个数据是以哪种码表示呢？

2.9 补码的作用有哪些？

三 如何访问数据以及表达数学意义上“数据”

3.1 程序中的数据来源于哪里或者说存储在哪里？字节和存储单元的对应关系？

3.2 为什么我们很难直接通过内存地址来访问数据？

3.3 想象一下，可不可以给某个内存地址起个名字，然后通过这个名字访问呢？

3.4 简述变量和常量的概念，变量的实质？

3.5 C 语言中变量名称必须符合标识符规范，标识符规定了哪些原则？C 语言的保留字可以当作变量名吗？

3.6 为什么变量需要指定数据类型？基本数据类型都有哪些？

3.7 整型根据长度和有无符号又可以细分为哪几种？修饰的关键字是什么？它们之间的区别是什么？

3.8 举例表达字符常量？字符类型和整型有什么关系，区别是什么？什么是 ASCII，它的作用是什么？`char a = 'a'`和`char a = 97`等价吗？

3.9 实数类型有几种？区别是什么？实型常量默认是哪种类型？

四 数据操作，赋值和算数运算符（数学运算）

4.1 计算机的 CPU 能做哪些具体操作？C 语言通过什么来描述这些操作呢？

4.2 运算符构成表达式，表达式一定有一个确定的值吗？

4.3 赋值运算符的左侧可以是常量吗？有什么要求？赋值表达式作为一个表达式，它的值是什么？

4.4 你觉得赋值的本质是什么？

4.5 常用的算数运算符都有哪些？

4.6 求余“%”运算符的操作数可以是实数类型吗？其他算术运算符对操作数有要求吗？

4.7 整数和实数的除法运算的区别是什么？

4.8 自增自减运算符的作用，对操作数有什么要求？以及它们在操作数左侧和右侧的区别是什么？

Day3-数据和操作

一 复合数据类型—数组

- 1.1 当我们需要在程序中模拟现实中的同类集合时，比如具体点儿的 100 个整数，抽象一些的 10 个人的年龄等数据模型的时候，我们上一天学的内容是不是有一些捉襟见肘呢？你有什么办法来解决吗？
- 1.2 请解释什么是数组？它有什么特点？
- 1.3 数组定义时有 3 要素包括（元素类型、数组名、元素数量），请解释这 3 个要素的作用？
- 1.4 数组名和普通变量名一样吗？它有什么特别的含义吗？数组中的元素跟其类型相同的变量有区别吗？
- 1.5 如何引用数组的某个成员？数组的第一个和最后一个元素的下标是什么？
- 1.6 数组的初始化可以放在声明的时候同时进行，如果只想对部分成员初始化是否可以？如何做？剩余未被初始化的成员的值是什么？
- 1.7 访问数组成员时如果下标越界，程序是否在编译时报错？
- 1.8 数组的长度（元素个数）可以是变量吗？
- 1.9 给定一个数组名，如何知道数组的大小（所占内存）呢？又如何确定数组的长度呢？

二 字符数组和字符串

- 2.1 我们的 C 程序有能够直接处理文字的能力，那如何在程序中表达诸如 “apple” 这样的单词或者 “Good Night!” 这样的语句呢，你有什么方法吗？
- 2.2 什么是字符数组？它和普通数组有什么区别吗？
- 2.3 什么是字符串？C 语言中有字符串类型吗？”a”和’a’一样吗？
- 2.4 `char string[] = “hello world”;`这种写法是否可以？如果可以 `string` 的长度是多少？
- 2.5 字符串常量为什么需要尾零(‘\0’)?

三 复合数据类型—结构体（了解内容）

- 3.1 简单定义一个描述 “书” 的结构体，成员需要有书的价格，书名、作者、出版日期等信息。

3.2 根据上面的“书”的结构体，定义一个具体的书的对象并初始化里面的成员，并输出这本书的信息。

四 数据操作，关系和逻辑运算符（bool 运算）

4.1 简述关系运算符的作用并列举所有的关系运算符。

4.2 关系运算符构成的表达式叫做关系表达式，关系表达式的值是什么？

4.3 C 语言中如何规定“真”和“假”的？

4.4 简述逻辑运算符的作用并列举所有的逻辑运算符。

4.5 逻辑运算表达式的值是什么？

4.6 数学中如果要表示某个数在 70 到 80 之间（不包括 70 和 80）可以用 $70 < a < 80$ 来表示，那么 C 语言中可不可以采用这种表达方式？应该怎么表示？

五 复杂的表达式—运算符的优先级和结合性

由于 C 语言中运算符的操作数既可以是普通的变量或者常量，也可以是其它任意运算符构成的表达式，这样多个不同的运算符可以以某种方式组合和嵌套在一起，但是任意一个表达式的值必须是唯一的，表达式不能是有歧义的，比如 $3+5*3$ 的结果是 24 还是 18 呢？

- (1) 数据对象和不同的运算符嵌套组合在一起可以构成更加复杂的表达式，但不管表达式有多复杂，也要有一个确定的值，计算机所谓的执行指令，就是去计算整个表达式的值。在计算最终的值的时候，中途可能要计算很多子表达式的值。
- (3) C 语言没有明确规定多个运算符参与的表达式的计算顺序。计算顺序取决于两个因素优先级和结合性。
- (3) C 语言规定运算符是有级别的，优先级用来标识运算符在表达式中的运算顺序的，高优先级的运算符所在的表达式将会被优先执行，比如表达式 $3+5*3$ 中，由于“*”的优先级高于“+”，所以操作数 5 被优先分配给 $5*3$ 这个表达式。
- (4) 同种优先级的运算符的运算顺序取决于运算符的结合性，优先级相同的运算符其结合性也肯定相同，大部分运算符的结合性都是从左到右的叫做左结合性，只有少数的运算符结合性是从右到左的右结合性。我们将在以后的运算符讲解中将会针对特定的运算符再详细分析。举例：`int a, b = 2; a = b = 3;` 则 a 的值是 2 还是 3 呢？思路：不存在优先级只考虑结合性，“=”运算符是右结合性，因此先执行 `b=3`, 再执行 `a=b`，所以 a 的值是 3，整个 `a=b=3` 表达式的值是 3。
- (5) 如果忘记运算符优先级的话，那么可以通过加（）的方式来表达你希望的运算顺序，括号中的表达式将被优先执行，比如表达式 $(3+5)*8$ 中， $3+5$ 将会被先执行。给某个子表达式加（）的方法是解决优先级的万能方法。

Day4-编程

一 程序的结构

程序从根本上来讲是由前两天学到的那些基本运算符构成的表达式组成的，表达式加上分号构成语句，程序的执行是自上而下按照语句的先后顺序一步一步执行的。根据我们目前所学，可以想象程序中所有的语句都有机会被执行，并且一旦被执行过，将再也没有机会执行第二次。

程序是用来解决实际问题的，而人类在思考解决问题的方法的时候，常常会根据判断做出决断，比如医生在给人看病时，会根据病人的实际情况制定不同的治疗方案，这叫做“对症下药”。在比如交警处理酒驾时他的思路是这样的，（1）凭借经验或者随机拦下过往车辆让司机进行酒精测试。（2）根据测试结果选择是放行还是进一步处理。

如果我们的程序中要模拟上面的情况应该怎么写呢？很显然，我们在程序中需要做出判断，并根据判断的结果来决定接下来执行什么。而程序的默认顺序执行方式是不可能做到选择性的执行某些语句的。

再来看另外一种情形，比如生活中每个人每天的大部分活动实际上都是对昨天以及过往的重复，如果我们的程序模拟这些话，那么模拟程序中的一部分代码其实在执行相同的操作，比如我们要重复执行 30 天的“吃饭和睡觉”，根据我们目前所学，我们的代码中要有 30 个吃饭睡觉的操作代码顺序放置，这样代码中将会有大量的重复操作，试想一下如果让我们模拟一年的吃饭睡觉，我们的代码是个什么样子呢？

为了解决上面两类问题，C 语言中引入了两种新的程序结构，条件执行和循环执行，再加上顺序执行就形成了 3 种程序的基本结构，实际上程序的基本结构就只有这 3 种类型，而这 3 种类型有机的组合在一起可以让我们的程序表达我们所有的想法。

二 条件执行

2.1 列举 if 语句的 3 种基本结构。

2.2 if 语句嵌套都需要注意些什么？

2.3 switch 语句相比 if else if 语句有什么优势？是否都可以用 if 语句代替？

2.4 switch(表达式)，对表达式有什么要求？case 后面可以是变量吗？

2.5 switch 语句中 break 的作用，如果没有 break 会出现什么样的后果？

三 循环执行

3.1 while 和 do while 有什么区别？

3.2 代码如何实现循环不退出？

3.3 for 循环能完全代替 while 循环吗？for 循环的优势是什么？

3.4 break 和 continue 在循环中的作用分别是什么？

Day5-编程

一 本周还未讲过的语法

1.1 隐式类型转换

举例说明什么是隐式类型转换？

类型转换或者说类型提升的原则是什么？

1.2 强制类型转换

强制类型转换的语法是？

强制类型转换会改变被转换的操作数的类型吗？

1.3 逗号运算符

你觉得逗号运算符有什么作用？

1.4 ?:运算符

请用 if else 来实现相同的功能。

1.5 简述+=/=/*/等运算符的含义以及优先级。

二 编程实战

2.1 思考一下，你为什么不能写出程序来？

2.2 你觉得怎么做才能写出程序来？

2.3 编程的技巧

- (1) 什么是编程思维？
- (2) 知己知彼，探讨计算机的优势和缺陷。
- (3) 如何编写一个程序。

Day6-代码抽象和复用（函数）

一 代码块和局部变量

1.1 什么是代码块？

1.2 代码块中声明的变量都有哪些局限？

二 如何使代码块复用

2.1 如果将代码块当作是放在内存中的某些二进制数据，那么如何引用它们呢？

2.2 如何跟代码块交流？

三 函数初步

3.1 函数的定义都有哪几个要素，分别是什么，都有哪些作用？

3.2 程序中如何引用一个函数？实参传递需要注意什么？

3.3 调用一个函数时，函数出现在程序中的位置是随意的吗？如果函数出现在某一个算术表达式中，那么该函数得具备什么样的条件？

3.4 请总结一下函数的作用。

四 函数的调用过程

4.1 程序中的函数是不是一定会在某个时刻被执行？

4.2 分析一下函数的调用过程。

4.3 函数执行期间用到的数据在函数调用结束之后还存在吗？分析一下函数里面声明的局部变量的生命周期。

4.4 static 修饰的局部变量有什么特点？

4.5 什么是栈？有什么作用？栈的生命周期是什么？

4.6 main 函数的栈跟普通函数的栈有区别吗？请分析一下。

五 程序中引用已定义实现的函数

5.1 我们之前使用过的 printf 等函数并不是由我们定义实现的，为什么我们可以直接使用

它？

5.2 为什么需要函数声明？

六 在函数中调用函数本身（递归）

6.1 函数中可以调用其他函数，能否调用它本身呢？如果能调用函数本身，在写程序时需要注意些什么呢？

6.2 你能试着分析以下函数的整个调用过程吗？

```
int big(int a[], int n)
{
    if (n == 1)
        return a[0];
    int b = big(a+1, n-1);
    return a[0] > b ? a[0] : b;
}
```

6.3 什么是递归终止条件？

6.4 分析一下递归的优缺点。

七 编程实战

7.1 初学者经常搞不清楚函数参数的作用而出现下面的代码片段。请分析出现该错误的原因。

```
int add(int a, int b)
{
    a = 3;
    b = 5;
    return a+b;
}
```

7.2 设计一个运算器能实现加减乘除运算。

7.3 设计一个函数，找出某个整形数组中的最大值。

7.4 对一个给定的数组进行排序。

7.5 设计一个函数，求某个字符串中某个字符出现的次数（通过返回值返回结果）。

7.6 设计一个类似于 `atoi` 的函数，该函数能将一个数字构成的字符串转化为 10 进制的数，例如将“12345”转化为 12345。

7.7 递归实现斐波那契数列。

7.8 递归求阶乘合，比如输入 5，则求解 $1! + 2! + 3! + 4! + 5!$ 。

Day7-指针（初步）

前言

任何我们之前学过的变量或者数组在内存中都有一块内存区域与之对应，但是我们并不需要了解这一细节，因为我们不会直接跟内存打交道，而是通过变量名称或者数组名等间接方式实现对内存的访问。但是有时候我们需要知道变量或者数组对应的地址。比如我们在某个函数中需要对包含 1000 个整数的数组进行操作，那么我们在实际传参的时候有两种传递方式，一种是将整个数组的内容拷贝给这个函数，另外一种就是简单的将数组的地址传递给函数。很显然第一种在空间和时间上的效率都比较低，实际上系统在对待函数中出现的“数组声明”参数时，是将其对应为“数组首地址”来看待的，我们会在接下来具体分析这个问题。

一 数据的地址

1.1 如何获取基本类型变量的地址？

1.3 数组的地址？数组成员的地址？

1.4 函数在内存中也是“一组数据”，它的地址如何获取？

1.5 如果有整型变量 a，那么 a 和 &a 的区别是什么？a 的类型是整型，那么 &a 的类型呢？

二 指针类型和指针变量

2.1 如何通过某个变量的地址值来引用这个变量？

2.2 地址值在 C 语言中也可以认为是有类型的叫做指针类型，那么指针类型的变量有没有，应该如何定义？

2.3 什么是 void *指针，它和 int *有什么区别？

2.4 作为变量指针变量跟普通变量有区别吗？

2.5 声明完一个指针变量后它的初始值是什么？就比如 int a;a 的值是什么？int *p;p 的值又是什么？

2.6 如果对指针求 sizeof，它的值是什么？

三 通过指针变量来访问数据

3.1 举例说明指向基本数据类型的指针变量？

3.2 指针是否可以指向一个数组？如果可以，应如何声明和初始化？

3.3 指针都可以执行哪些运算？举例说明。

3.4 如果 `int *p = 0x1001`;那么 `p+1==0x1002`; `p-1==0x1000` 这两个表达式的值是什么?

四 编程实战

- 4.1 函数的形参类型是否可以是指针类型, 跟普通类型有什么区别?
- 4.2 数组作为函数的行参时, 表示的是数组本身还是数组的地址?
- 4.3 函数 `void test(int a[100], int n)`声明中。如果对 `a` 求 `sizeof`, 它的值是什么? 数组个数 100 有意义吗?
- 4.4 输入某个字符串, 计算该字符串的长度。
- 4.5 实现一个同 `strcpy` 函数相同功能的函数 `mystrcpy`。
- 4.6 实现两个函数分别实现数组的插入和删除操作。
- 4.7 采用指针操作的方式对某个整形数组进行选择排序。
- 4.8 实现一个字符串压缩函数, 压缩策略是将连续出现的字符转换成字符+次数的形式, 例如: 字符串 “`abcccddeaaaaa`” 压缩后变成: “`abc3d2ea5`”, 再编写一个相反的解压函数。

Day8-指针（复杂的指针声明）

一 复杂的指针声明

- 1.1 字符串除了可以用字符数组来引用，还可以通过指针来引用，应该怎么声明呢？
- 1.2 指针变量也是变量，它也有地址，如果要使用某个变量来存储它的地址，该变量应该如何声明？
- 1.3 对于 `char a[3][5]`，如何声明一个指针指向它？
- 1.4 对于函数 `void func(int a, char *b)`，如何声明一个指针指向它？
- 1.5 请用 `typedef` 简化 1.2、1.3 和 1.4 的指针声明。

二 `const` 修饰指针

- 2.1 `const int *a` 和 `int const *a` 有区别吗？
- 2.2 `const int *a` 和 `int * const a` 有区别吗？
- 2.3 下面写法是否正确
 - (1) `const int *a; int b; a = &b; *a = 100;`
 - (2) `int *const a; int b; a = &b; *a = 100;`
 - (3) `int b; int *const a = &b; *a = 100;`
- 2.4 思考一下 `const int *const a` 表示什么意思？
- 2.5 如果有 `const int *a; int *b;` 那么语句 `a = b;` 是否可行？反过来 `b = a;` 又是否可行？请思考 `const` 修饰指针的作用？

三 指针数组

- 3.1 数组的元素也可以是指针类型，应该如何声明？
- 3.2 有 `char *array[10]` 和 `int *array1[10]`，请问 `sizeof(array)` 和 `sizeof(array1)` 的大小分别是什么？

四 编程实战

- 4.1 仿照 `strcmp` 函数实现一个 `mystrcmp` 函数。
- 4.2 对若干个字符串按照 `strcmp` 函数的比较结果进行排序（指针数组）。
- 4.3 利用命令行参数实现一个计算器，能执行 `+-*/` 操作。比如终端输入 `:3 + 5`，输出 `8`。

Day9-指针和内存管理

一 动态内存分配

- 1.1 什么是静态内存分配，举例说明，它的优点和缺点是什么？
- 1.2 什么是动态内存分配？它有什么积极意义？
- 1.3 C 语言中如何动态申请和释放内存？申请的内存存在什么地方？
- 1.4 动态内存分配的缺陷是什么？

二 程序的内存分布和管理

- 2.1 什么是全局变量？它放在什么地方？全局变量的缺点是什么？
- 2.2 请分析全局变量，static 变量、局部临时变量以及动态分配的数据对象它们的作用域和生命周期。
- 2.3 请分析并对比栈内存管理和堆内存管理这两种模式的不同特点。

三 编程实战

- 3.1 仿造 strdup 函数实现一个 mystrdup 函数
- 3.2 输入两个字符串，将它们拼接在一起，然后返回新的字符串的地址。
- 3.3 仿照 strstr 实现一个 mystrstr。
- 3.4 仿照 qsort 实现一个快速排序 myqsort 函数。

Day10-预处理

一 预处理—头文件

1.1 什么是预处理？

1.2 头文件的作用？头文件的内容一般是什么？

1.3 `#include <file.h>`和`#include "file.h"`有什么区别？

1.4 头文件是否可以被重复包含？为什么？

二 预处理—宏

2.1 宏的作用和意义是什么？

2.2 带参宏和函数的区别？

2.3 `typedef unsigned char * p0` 和 `#define p1 unsigned char *`，它们有什么区别？

三 预处理—条件编译

3.1 什么是条件编译？

3.2 `#if` 对应的条件编译和 `if` 对应的条件执行的区别是什么？

3.3 如何用条件编译来防止头文件重复包含？

四 代码分散在多个文件中

4.1 如果代码工程中的代码量比较大或者需要多个人共同参与开发时，如果把所有的代码都写在同一个 C 文件中是不现实也不科学的，那么代码分散的原则是什么？如何引用在其他文件中定义的全局变量或者函数？

4.2 多文件如何使用 GCC 来编译？

4.3 `static` 修饰的函数有什么具体含义？

五 编程实战

编写一个简易的字典，实现查找功能，字典的单词和释义保存在一个全局的数组中，格式：

```
char *dic[] = {  
    "apple:苹果", "bad:坏的", ... ... "ball: 球" };
```

项目要求：声明放在头文件中，实现放在多个 C 文件中,其中字符串操作的函数放在 `strop.c` 中，排序函数放在 `sort.c` 中，这两个文件各自都有自己的头文件。

Day11-抽象数据模型

一 结构体深入

- 1.1 结构体是否可以整体赋值？
- 1.2 结构体可否构成数组？
- 1.3 对结构体成员的引用可以通过指针和变量两种方式，它们语法上的区别是什么？
- 1.4 结构体的成员如果是数组应该如何引用整个数组？如果引用数组的某个成员应该怎么写？
- 1.5 结构体的成员如果是指针，它的引用方式跟其他类型的成员有什么区别吗？
- 1.6 结构体作为函数参数的时候跟数组作为函数参数时有什么不同？
- 1.7 结构体定义是否可以嵌套？如果是指向它自身的指针是否可以？
- 1.8 如何确定结构体大小？（了解）

二 联合体和枚举

- 2.1 联合体跟结构体有什么区别？
- 2.2 下面联合体的 size 是多少？如果有 Test x; x.b = 0x12345678;那么 x.a[0]的值是什么？

```
typedef union {  
    char a[8];  
    int b;  
} Test;
```

- 2.2 什么是枚举，它有什么作用？
- 2.3 下面枚举中 Blue 和 White 对应的值分别是什么？

```
enum {  
    Red,  
    Blue,  
    Black = -1;  
    White  
};
```

三 位操作

- 3.1 列举所有的位运算符。位运算对操作对象有什么要求？

四 编程实战

4.1 请用宏分别来实现对某个整数的某一位的下列操作：

- a 某位清 0。
- b 某位置 1。
- c 对某一位反置（如果是 0 就变成 1，反之一样）。

4.2 使用位操作的方式，实现两个整数变量值的交换。

4.3 输出某个 10 进制整数的 2 进制版本，比如输入 10，输出 0000 1010。

4.4 利用动态结构体数组，实现一个学生成绩管理系统，支持信息录入和按照成绩排序两个功能。

Day12-C 标准库的使用

一 字符串操作

- 1.1 学会使用 man 手册来学习系统函数的使用。
- 1.2 列出常用的字符串操作函数。

二 文件操作

- 2.1 文件操作的流程是什么？
- 2.2 什么是标准输出 / 输入 / 报错？打开一个文件的模式都有哪些？如何在指定的文件位置开始操作？

三 编程实战

用普通文件来模拟数据库，实现一个学生成绩管理系统，增加数据的文件存储功能。

Day13-数据结构初步（选）

一 常见数据结构

1.1 什么是栈？它的特点是什么？栈的操作都有哪些？

1.2 什么是队列？有什么特点？它和栈的区别是什么？队列的操作都有哪些？

1.3 什么是链表？跟数组有什么区别？常见的操作有哪些？

二 编程实战

（1）用数组实现栈的“出栈”和“入栈”的操作。

（2）用数组实现循环队列的“出队列和入队列”操作。

（3）实现一个单链表，能够进行“增删查改”操作。（选择掌握）

编程实战

编程实战说明：

编程实战题目根据完成时间的顺序分为 3 部分：

第一部分：编程思维训练

目的：主要是用来训练学员对 C 语言运算符的驾驭能力，学员必须能非常熟练的用运算符构成的 C 表达式来表达自己的想法。

练习时间：day2-day3

第二部分：综合练习基础部分

目的：本部分的练习题必须全部完成，本阶段的题目逻辑上比较简单，但是可以充分的练习语法，同时培养编程思维和编程感觉，哪怕是再没基础的同学，如果能独立的完成这 60 道题目，那么肯定能够顺利通过本阶段的考核。

练习时间：day4-day5+day6+day7（第 6，7 天如果是周末则表示是周末，如果不是周末则时间顺延），要求所有同学都必须做完 60 道题目。

第三部分：综合练习高级部分

目的：本部分的题目逻辑稍微复杂一些，能够进一步提升学员的 C 语言编程水平。对于基础差的同学，如果没有时间和能力可以先不用做这部分题目。

练习时间：教学前 10 天内。

一、编程思维训练（学会用 C 语言运算符来表达你的想法）

用 C 表达式来表达下面的题目，比如判断一个整数 a 是不是等于 9，对应的表达式为 $a == 9$ 。

数学运算

1. 将数学表达式 $c = 3a + 5b$ 翻译成 C 的表达式。
2. 已知某个圆的半径为 a ，表示圆的面积和圆的周长。
3. 已知字符变量 a 的值对应的是某大写字母，将其转换为对应的小写字母。
4. 获取某个整数 a 的个位数值（比如 75 是 5，109 是 9）。
5. 获取某个大于 3 位的 10 进制整数 a 的百位数值（比如 1923 是 9）。
6. 假设整数 $a=1, b=5$; 表达式 $a++ + ++b$ 的值是？
7. 不借助第三个变量，用一组表达式来交换两个整数 a 和 b 。
8. 设变量 m, n, a, b, c, d 均为 0，执行 $(m=a==b) || (n=c==d)$ 后， m, n 的值是？
9. $\text{int } a=8, b=5, c;$ ，执行语句 $c=a/b+0.4$; 后， c 的值为？
10. 给定一个浮点数，要求将其转化为只保留小数点后 2 位的小数，其中需要四舍五入，比如 $1.23678 \rightarrow 1.24$ 。

Bool 运算

1. 假设 $a=1$ ，则表达式 $a=1$ 和 $a==1$ 的值分别是？
2. 假设 $a=1$ ；表达式 $!(a-2)$ 的值是？
3. 表达式 $-1 \leq 3 \leq -5$ 的值是？
4. 假设 $a=3, b=4, c=5$ ；表达式 $a+b > c \&\& b==c$ 的值是？
5. 假设 $a=3, b=4, c=5$ ；表达式 $!(a > b) \&\& !c || 1$ 的值？
6. 对于整数 a ，如果表达式 $(a \% 5 == 3) \&\& (a \% 7 == 4) \&\& (a \% 5 == 3) \&\& (a < 20)$ 为真，那么 a 的值是？
7. 判断某个整数 a 是否能被 7 整除。
8. 判断某个边长为 a 的正方形的面积是否比某个半径为 b 的圆的面积大。
9. 判断某个正整数 a 的个位数值在 3，8 的范围之内。比如 16 满足，12、19 不满足。
10. 判断某个字符变量 a 是不是一个英文字符。
11. 判断某个字符变量 a 是不是一个数字字符。
12. 判断某个正整数 a 是不是一个质数（只能被自己和 1 整除的数）。
13. 给定整数 a ，判断它是否能同时被 7 和 3 整除。
14. 给定整数 a ，判断它是否能被 7 或者被 3 整除。
15. 给定字符 a 和 b ，判断它们是否有一个字符是阿拉伯数字符号。
16. 任意给定某一年为 a ，判断它是否为闰年。

逗号运算符

1. 如果有 $\text{int } a$; $\text{int } i = 2$; 则执行完 $a = (i++, i++, i++)$ 后; a 和 i 的值分别是？

条件运算符

1. 有 3 个整数 a, b, c ，不用 if 只用基本表达式来找到其中的最大值。
2. 已知: $\text{int } n, i=1, j=2$; 执行语句 $n=i < j ? i++ : j++$; 则 n 、 i 和 j 的值是？

二、综合练习基础部分（所有同学必做）

分支练习部分

1. 通过 `scanf` 输入一个整数，判断它是否是一个偶数？如果是，则输出这个数。
2. 通过 `scanf` 输入一个整数，输出这个整数对应的绝对值。
3. 通过 `scanf` 输入一个字符，判断它是不是一个大写的英文字符，如果是，则输出这个字符。
4. 通过 `scanf` 输入一个字符，判断它是不是一个小写英文字符，如果是则输出这个字符的大写，如果不是则原样输出。
5. 通过 `scanf` 输入一个字符，判断其是不是一个阿拉伯数字字符，如果是则打印 YES 否则打印 NO。
6. 通过 `scanf` 输入代表某一年的整数，如果该年是闰年则输出 YES 否则输出 NO。
7. 通过 `scanf` 输入两个整数，将其中较大的数输出。
8. 通过 `scanf` 输入三个整数，将其中较小的数输出。
9. 通过 `scanf` 输入三个整数，按照从小到大的顺序输出。（采用多重分支）
10. 通过 `scanf` 输入一个 3 位整数，判断这个数是不是一个对称数，比如 757，626 等都是。
11. 给出一个百分制成绩，要求输出成绩等级 'A'、'B'、'C'、'D'、'E'。90 分以上输出 'A'，80~89 分输出 'B'，70~79 分输出 'C'，60~69 分输出 'D'，60 分一下输出 'E'。
12. 有一个函数：
 $y = 1; (x \leq 1)$
 $y = x; (x > 1 \ \&\& \ x < 10)$
 $y = 2x + 1; (x \geq 10)$
终端输入 `x` 值，编程实现求解该函数的值。
13. 输入一个时间，输出它的下一秒时间。比如输入 12:30:59 秒下一秒是 12:31:00。

综合练习部分

1. 在屏幕上输出 10 行内容，每行的内容都是 “*”。
2. 在屏幕上输出 10 行内容，每行的内容都是 “*****”。
3. 在屏幕上输出 10 行内容，每行的内容都不一样，第 1 行一个星号，第 2 行 2 个星号，依此类推第 10 行 10 个星号。
4. 在屏幕上输出 10 行内容，每行的内容都是 “1”。
5. 在屏幕上输出 10 行内容，每行的内容都不一样，第 1 行输出 “1”，第 2 行输出 “2”，依此类推第 10 行输出 “10”。
6. 在屏幕上输出以下内容：
A
AB
ABCD
ABCDE
ABCDEF
7. 在屏幕上输出以下内容：

- 12345
1234
123
12
1
8. 计算 10 个 99 相加后的值并输出。
 9. 计算从 1 加到 100 的值并输出。
 10. 计算 10 的阶乘 ($1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8 \times 9 \times 10$)。
 11. 计算 2 的 20 次方。
 12. 计算从 1 到 1000 以内所有奇数的合并输出。
 13. 计算从 1 到 1000 以内所有能被 3 或者 17 整除的数的合并输出。
 14. 计算从 1 到 1000 以内所有能同时被 3, 5 和 7 整除的数的合并输出。
 15. 计算 1 到 100 以内能被 7 或者 3 整除但不能同时被这两者整除的数的个数。
 16. 计算 1 到 100 以内能被 7 整除但不是偶数的数的个数。
 17. 计算从 1 到 100 之间临近两个整数的合并并依次输出。比如第一次输出 $3(1+2)$, 第二次输出 $5(2+3)$, 最后依次输出 $199(100+99)$ 。
 18. 计算从 1 加到 100 中途的所有数值的合, 比如第一次输出 1, 第二次输出 1+2 的合, 第 3 次输出 1+2+3 的合, 最后一次输出 1 到 100 所有数相加之后的合。
 19. 判断 1077 是不是一个质数 (质数是只能被 1 和它自身整除的数)。
 20. 一球从 100 米高度自由落下, 每次落地后反跳回原高度的一半; 再落下, 求它在第 10 次落地时, 共经过多少米?
 21. 将某个 8 位的整数所有位的数值加在一起并输出。
 22. 给定一个 5 位的整数, 将该数按照 10 进制位逆置, 例如给定 12345 变成 54321, 12320 变成 2321。
 23. 求 $s=a+aa+aaa+aaaa+aa...a$ 的值, 其中 a 是一个数字 (1-9 之间) 计算的数据的个数是 5。例如 $2+22+222+2222+22222$ 。
 24. 给定一个正整数 n 按照下面的公式计算 S(浮点类型)的值。公式:
$$S=1+1/(1+2)+1/(1+2+3)+.....+1/(1+2+3+4+.....+n)$$
 25. 给定某个字符数组, 统计数组中所有英文字符的个数, 比如 “123fdd” 中有 3 个。
 26. 给定某个字符数组, 统计数组中所有英文字符和阿拉伯数字的个数, 比如 “123fdd” 中有英文字符有 3 个, 数字 3 个。
 27. 给定某个拥有 5 个元素的字符数组, 数组的成员都有阿拉伯字符构成, 试着将该数组转换成一个整数, 比如字符数组的内容是: {'1','2','3','3','2'} 则将被转换成 12332。
 28. 给定一个完全由英文字符构成的数组, 将数组中的小写字母转换成大写字母, 大写字母转换成小写字母并输出。例如 “abcGGG” 转化为 “ABCggg”。
 29. 给定一个完全由英文字符构成的数组, 将数组中下标为偶数的字符都转换为大写 (如果原来是大写则不变)。
 30. 给一个完全由英文字符构成的字符数组加密, 加密原则如下, 除了字符 ‘Z’ 和 ‘z’ 之外, 每个字符变成 ASCII 码值比它大 1 的字符, 也就是 ‘A’ 变成 ‘B’。‘Z’ 或者 ‘z’ 转化为 ‘A’ 或者 ‘a’。
 31. 计算某个由英文、数字以及标点符号构成的数组的总宽度, 其中英文字符的宽度为 1cm, 数字宽度为 0.5cm、标点符号宽度为 0.8cm。

32. 接上题，如果规定行的宽度为 10cm，将某个字符长度超过 50 的字符串截断，恰好使 10cm 宽的行能容纳。输出这个被截断的子数组。
33. 给定某个整型数组，计算该数组所有偶数的合。
34. 给某个整型数组赋值，赋值规律如下，下标能被 3 整除的都赋值为 1，能被 5 整除的都赋值为 2，能被 7 整除的都赋值为 3，能被 3、5、7 任意两个或者 3 个都能整除的数赋值为 8，其余都赋值为 0。
35. 通过终端输入 10 个整数并将其保存在一个整型数组中，数字保存在数组中的顺序与下标正好相反，也就是第一个被输入的数放在数组最后一个元素中，最后一个输入的数字放到第一个元素中。
36. 通过终端输入 10 个整数，计算 10 个整数中所有能被 3 整除的数的合。
37. 给定一个 5 个元素构成的整型数组，每个元素的值都在 0-9 之间，按照位置将其组成一个 5 位数并输出，例如 `int a[5] = {1,2,2,3,7};` 则输出 73221。
38. 给定 2 个大小一样的整型数组，将某个数组作为源数组，另一个作为目的数组，然后将源数组的内容拷贝到目的数组。
39. 给定一个整型数组，将第一个跟最后一个元素的内容交换。
40. 给定一个整型数组，从第 1 个元素开始将相邻的两个元素分别相互交换。交换完后，第 1 个元素将变成最后一个元素，其余元素都前进一位。
41. 给定一个有 10 个整形数的元素，将前 5 个元素跟后 5 个元素做整体交换，比如 `{1,1,1,1,1,2,3,2,2,2}->{2,3,2,2,2,1,1,1,1,1}`。
42. 判断一个整型数组是否是对称数组，例如 `{1,2,3,3,2,1}` 和 `{1,6,8,1,8,6,1}` 都是对称数组。
43. 给定两个大小一样的整型数组，交换这两个数组的内容。
44. 给定两个大小一样的整型数组，将两个数组中下标一样的元素两两相加，相加后的结果作为这两个数组对应下标的新值，也就是操作完毕后两个数组的内容完全相同。
45. 给定一个能容纳 10 个元素的整型数组，现有 9 个元素，现在第 5 个元素的位置插入一个数字 88，后面的数字顺序后移。
46. 给定一个 10 个元素的整型数组，现在将第 2 个元素删除，后面的数组顺序前移。
47. 给定一个有 100 个元素的数组，查询数组中是否有元素的值等于某个数 n。
48. 给定一个整型数组，求该数组元素中最大值的下标。
49. 给定一个整型数组，求该数组中第二大的数的下标。
50. 给定一个整型数组，求该数组中数值小于 10 的元素的个数。
51. 给定一个整型数组，计算大于该数组平均值的元素的个数。
52. 给定一个整型数组，找到数组中的最小值，并将其放到数组的首元素中，原来首元素的内容放到最小值所在的元素中。
53. 给定一个整型数组，统计某个整数在数组中出现的次数。
54. 给定一个英文句子，单词之间用 1 个空格分开，求出第 2 个单词的偏移位置。例如 “Professor du comes from Korea” 的偏移位置是 10。
55. 给定一个英文句子，单词之间用 1 个空格分开，求其中所有单词的数量。
56. 给定两个字符数组，将这两个拼接起来放在第一个数组中(假定第一个数组足够长)，比如 “abc” 和 “123” 构成 “abc123”。
57. 将一个字符数组循环右移 2 位。比如 “12345”->“45123”，假定字符数组中字符的数量大于 2。
58. 给定一个整型数组，数组的长度为 N (N>3)，从数组中寻找一个连续的长度为 3 的子数组，要求该子数组的合最大。

59. 给定两个长度一样的整型数组，判断两个数组是否相同，相同的原则是数组中的每一个相互对应的元素的“合值”相同，“合值”是指元素对应的整数所有位的合，例如：a[0]的值是 1112，b[0]的值是 23，则这两个元素“相同”。
60. 给定两个字符数组，比较这两个字符数组的大小，比较的原则是字符数组中所有字符的 ASCII 值相加后的合值，合值越大则字符数组越大。

综合练习高级部分（基础差者选择性做，要求所有题目需要用函数来实现）

1. 在屏幕上输出以下图形：

```
*  
***  
*****
```

2. 在屏幕上输出以下图形：

```
0  1  1  1  1  
-1 0  1  1  1  
-1 -1 0  1  1  
-1 -1 -1 0  1  
-1 -1 -1 -1 0
```

3. 在屏幕上输出以下图形：

```
*  
***  
*****  
***  
*
```

4. 任意给定一个位数不超过 9 的整数将其所有位的数值加在一起并输出。
5. 任意给定一个不超过 9 位的整数，将其高低位翻转，例如给定 12345 变成 54321，12320 变成 2321。
6. 有 1、2、3、4 个数字，输出所有由这 4 个数字组成的互不相同且无重复数字的三位数？
7. 对一个整数进行质因数分解，例如 60 可以分解为 $60 = 2 * 2 * 3 * 5$ 。
8. 计算 N 的阶乘后 0 的个数。
9. 打印出所有的“水仙花数”，所谓“水仙花数”是指一个三位数，其各位数字立方和等于该数。
10. 有这样一个数列，这个数列中前两个数为 1 和 1，从第三个数开始，每个数都等于前两项之和，使用循环输出这个数列的前 20 项
例如: 1 1 2 3 5 8 13 21 35
11. 一个数如果恰好等于它的因子(除了自身以外的约数)之和，这个数称为完数，例如 6 的因子是 1,2,3，而 $6=1+2+3$ ，因此 6 是完数，请找出 1-1000 之内的完数。
12. 猴子吃桃问题：猴子第一天摘下若干个桃子，当即吃了一半，还不过瘾，又多吃了一个，第二天早上又将剩下的桃子吃掉一半，又多吃了一个。以后每天早上都吃了前一天剩下的一半零一个。到第 10 天早上想再吃时，见只剩下一个桃子了。求第一天共摘了多少？
13. 任意给定一个整型数组，将数组里的元素的顺序倒转过来。
14. 任意给定一个整型数组，将某一个数字插入到数组的某个位置上（插入后原来数组最后一个元素丢弃）。

15. 任意给定一个整型数组，删除数组中某一个元素，删除元素后面的元素要依次往前递补。
16. 给定一个按照由小到大排好序的整型数组，快速定位数组中是否有某个整数 n 。
17. 给定一个排序（升序）整型数组，将某个数 n 插入到该数组中，要求新数组也必须是有序的。
18. 给定一个整型数组，数组中的元素的值在 1-9 之间（包括 1 和 9），统计数组中每个元素对应的值出现的次数。比如数组 {2,3,2,3,5,6}，2 和 3 分别出现了 2 次，5 和 6 出现了 1 次。
19. 将两个已排序（升序）的数组合并成一个新数组，合并后的数组也是排好序的。
20. 将一个整数列按奇数在前偶数在后的顺序重新排列，并要求奇偶两部分分别有序。
21. 将某个字符数组中 ASCII 为偶数的字符全都删除，删除后形成一个新的数组。
22. 给定一个英文句子，单词之间用 1 个或者多个空格分开，求其中所有单词的数量。
23. 将任意一个字符数组循环右移 N 位。
24. 给定一个整数和一个整数数组，从数组里找两个元素，他们的合等于给定的整数。
25. 在一个字符串中找到第一个只出现一次的字符。如输入 abaccdeff，则输出 b。
26. 给定两个排好序的数组 A, B，大小分别为 n, m 。给出一个高效算法查找 A 中的哪些元素存在 B 数组中。
27. 输入两个字符串，从第一个字符串中删除第二个字符串中所有的字符。例如，输入 "They are students." 和 "aeiou"，则删除之后的第一个字符串变成 "Thy r stdnts."。
28. 给定一个字符数组，判断某个字符串是否在这个字符数组中，比如 "abdefghj78" 中包含 "defg"。
29. 给定一个整型数组，统计数组中每个元素对应的值出现的次数。比如数组 {2,3,2,3,5,6}，2 和 3 分别出现了 2 次，5 和 6 出现了 1 次。
30. 两个乒乓球队进行比赛，各出三人，甲队为 A,B,C 三人，乙队为 X,Y,Z 三人，已抽签决定比赛名单，有人向队员打听比赛的名单，A 说他不和 X 比，C 说他不和 X,Z 比，请编写程序找出三队赛手的名单。
31. 返回一个整数数组的连续子数组，该子数组的和在所有连续子数组里和最大。例如，[-2,1,-3,4,-1,2,1,-5,4]，子数组 [4,-1,2,1] 是符合该要求的最大子数组。
32. n 个数字 (0,1,...,n-1) 形成一个圆圈，从数字 0 开始，每次从这个圆圈中删除第 m 个数字（第一个为当前数字本身，第二个为当前数字的下一个数字）。当一个数字删除后，从被删除数字的下一个继续删除第 m 个数字。求出在这个圆圈中剩下的最后一个数字。