**练习 1-1** 在你自己的系统中运行“hello, world”程序。再有意去掉程序中的部分

内容，看看会得到什么出错信息。

**练习 1-2** 做个实验，当printf 函数的参数字符串中包含\c（其中c 是上面的转义

字符序列中未曾列出的某一个字符）时，观察一下会出现什么情况。

**练习 1-3 参考教程1.2节**修改温度转换程序，使之能在转换表的顶部打印一个标题。

**练习 1-4** 编写一个程序打印摄氏温度转换为相应华氏温度的转换表。

**练习 1-5** 修改温度转换程序，要求以逆序（即按照从300 度到0 度的顺序）打印温

度转换表。

**练习 1-6** 验证表达式getchar() != EOF的值是0还是1。

**练习 1-7** 编写一个打印EOF值的程序。

**练习 1-8** 编写一个统计空格、制表符与换行符个数的程序。

**练习 1-9** 编写一个将输入复制到输出的程序，并将其中连续的多个空格用一个空格代

替。

**练习 1-10** 编写一个将输入复制到输出的程序，并将其中的制表符替换为\t，把回退符

替换为\b，把反斜杠替按为\\。这样可以将制表符和回退符以可见的方式显示出来。

**练习 1-11** 你准备如何测试单词计数程序？如果程序中存在某种错误，那么什么样的输

入最可能发现这类错误呢’

**练习 1-12** 编写一个程序，以每行一个单词的形式打印其输入。

**练习 1-13** 编写一个程序，打印输入中单词长度的直方图。水平方向的直方图比较容易

绘制，垂直方向的直方图则要困难些。

**练习 1-14** 编写一个程序，打印输入中各个字符出现频度的直方图。

**练习 1-15** 重新编写1.2节中的温度转换程序，使用函数实现温度转换计算。

**练习 1-16** 修改打印最长文本行的程序的主程序main，使之可以打印任意长度的输入

行的长度，并尽可能多地打印文本。

**练习 1-17** 编写一个程序，打印长度大于80 个字符的所有输入行。

**练习 1-18** 编写一个程序，删除每个输入行末尾的空格及制表符，并删除完全是空格的

行。

**练习 1-19** 编写函数 reverse(s)，将字符串s 中的字符顺序颠倒过来。使用该函数

编写一个程序，每次颠倒一个输入行中的字符顺序。

**练习 1-20** 编写程序 detab，将输入中的制表符替换成适当数目的空格，使空格充满到

下一个制表符终止位的地方。假设制表符终止位的位置是固定的，比如每隔n 列就会出现一个制表符终止位。n应该作为变量还是符号常量呢？

**练习 1-21** 编写程序 entab，将空格串替换为最少数量的制表符和空格，但要保持单词

之间的间隔不变。假设制表符终止位的位置与练习1-20 的detab程序的情况相同。当使用一个制表符或者一个空格都可以到达下一个制表符终止位时，选用哪一种替换字符比较好？

**练习 1-22** 编写一个程序，把较长的输入行“折”成短一些的两行或多行，折行的位置

在输入行的第n 列之前的最后一个非空格之后。要保证程序能够智能地处理输入行很长以及在指定的列前没有空格或制表符时的情况。

**练习 1-23** 编写一个删除C语言程序中所有的注释语句。要正确处理带引号的字符串与

字符常量。在C语言中，注释不允许嵌套。

**练习 1-24** 编写一个程序，查找C语言程序中的基本语法错误，如圆括号、方括号、花

括号不配对等。要正确处理引号（包括单引号和双引号）、转义字符序列与注释。（如果读者想把该程序编写成完全通用的程序，难度会比较大。）

**练习 2-1** 编写一个程序以确定分别由signed及unsigned 限定的char、short、

int与long类型变量的取值范围。采用打印标准头文件中的相应值以及直接计算两种方式实现。后一种方法的实现较困难一些，因为要确定各种浮点类型的取值范围。

**练习 2-2** 在不使用运算符&&或||的条件下编写一个与上面的for 循环语句等价的循

环语句。

**练习2-3** 编写函数 htoi(s)，把由十六进制数字组成的字符串（包含可选的前缀0x

或0X）转换为与之等价的整型值。字符串中允许包含的数字包括：0～9、a～f以及A～F。

**练习 2-4** squeeze(s1, s2)，将字符串s1 中任何与字符串s2 中字符匹配的字符

都删除。

**练习 2-5** 编写函数 any(s1, s2)，将字符串s2中的任一字符在字符串s1中第一次

出现的位置作为结果返回。如果s1中不包含s2中的字符，则返回-1。（标准库函数strpbrk具有同样的功能，但它返回的是指向该位置的指针。）

**练习 2-6** 编写一个函数 setbits(x, p, n, y)，该函数返回对x执行下列操作后

的结果值：将x中从第p位开始的n个（二进制）位设置为y中最右边n位的值，x的其余

各位保持不变。

**练习 2-7** 编写一个函数 invert(x, p, n)，该函数返回对x 执行下列操作后的结

果值：将x中从第p位开始的n个（二进制）位求反（即，1 变成0，0 变成1），x的其余各

位保持不变。

**练习 2-8** 编写一个函数rightrot(x, n)，该函数返回将x循环右移（即从最右端

移出的位将从最左端移入）n（二进制）位后所得到的值。

**练习 2-9** 在求对二的补码时，表达式x &= (x – 1)可以删除x中最右边值为1 的

一个二进制位。请解释这样做的道理。用这一方法重写bitcount函数，以加快其执行速度。

**练习2-10** 重新编写将大写字母转换为小写字母的函数lower，并用条件表达式替代其

中的if-else结构。

**练习 3-1** 在上面有关折半查找的例子中，while循环语句内共执行了两次测试，其实

只要一次就足够（代价是将更多的测试在循环外执行）。重写该函数，使得在循环内部只执行一次测试。比较两种版本函数的运行时间。

**练习 3-2** 编写一个函数escape(s, t)，将字符串t 复制到字符串s 中，并在复制

过程中将换行符、制表符等不可见字符分别转换为\n、\t等相应的可见的转义字符序列。要求使用swich语句。再编写一个具有相反功能的函数，在复制过程中将转义字符序列转换为实际字符。

**练习3-3** 编写函数 expand(s1, s2)，将字符串s1 中类似于a-z 一类的速记符号在字符串s2中扩展为等价的完整列表abc…xyz。该函数可以处理大小写字母和数字，并可

以处理a-b-c、a-z0-9与-a-z等类似的情况。作为前导和尾随的-字符原样排印。

**练习 3-4** 在数的对二的补码表示中，我们编写的itoa函数不能处理最大的负数，即

n等于-2 字长-1的情况。请解释其原因。修改该函数，使它在任何机器上运行时都能打印出正确的值。

**练习 3-5** 编写函数 itob(n, s, b)，将整数n转换为以b为底的数，并将转换结果以字符的形式保存到字符串s中。例如，itob(n, s, 16)把整数n格式化成十六进制整数

保存在s中。

**练习 3-6** 修改itoa函数，使得该函数可以接收三个参数。其中，第三个参数为最小

字段宽度。为了保证转换后所得的结果至少具有第三个参数指定的最小宽度，在必要时应在所得结果的左边填充一定的空格。

**练习 4-1** 编写函数 strindex(s, t)，它返回字符串t在s中最右边出现的位置。

如果s中不包含t，则返回-1。

**练习 4-2** 对atof函数进行扩充，使它可以处理形如123.45e-6的科学表示法，其中，浮点数后面可能会紧跟一个e或E以及一个指数（可能有正负号）。

**练习4-3** 在有了基本框架后，对计算器程序进行扩充就比较简单了。在该程序中加入

取模（%）运算符，并注意考虑负数的情况。

**练习 4-4** 在栈操作中添加几个命令，分别用于在不弹出元素的情况下打印栈顶元素；

复制栈顶元素；交换栈顶两个元素的值。另外增加一个命令用于清空栈。

**练习 4-5** 给计算器程序增加访问sin、exp 与pow 等库函数的操作。有关这些库函

数的详细信息，参见附录B.4节中的头文件<math.h>。

**练习 4-6** 给计算器程序增加处理变量的命令（提供26 个具有单个英文字母变量名的

变量很容易）。增加一个变量存放最近打印的值。

**练习 4-7** 编写一个函数ungets(s)，将整个字符串s 压回到输入中。ungets 函数

需要使用buf和bufp吗？它能否仅使用ungetch 函数？

**练习 4-8** 假定最多只压回一个字符。请相应地修改getch与ungetch这两个函数。

**练习 4-9** 以上介绍的getch与ungetch函数不能正确地处理压回的EOF。考虑压回

EOF时应该如何处理？请实现你的设计方案。

**练习 4-10** 另一种方法是通过getline 函数读入整个输入行，这种情况下可以不使用

getch与ungetch函数。请运用这一方法修改计算器程序。

**练习 4-11** 修改getop 函数，使其不必使用ungetch 函数。提示：可以使用一个

static类型的内部变量解决该问题。

**练习 4-12** 运用printd函数的设计思想编写一个递归版本的itoa函数，即通过递归

调用把整数转换为字符串。

**练习 4-13** 编写一个递归版本的reverse(s)函数，以将字符串s倒置。

**练习 4-14** 定义宏swap(t, x, y）以交换t类型的两个参数。（使用程序块结构会对

你有所帮助。）

**练习 5-1** 在上面的例子中，如果符号+或-的后面紧跟的不是数字，getint函数将把符号视为数字0 的有效表达方式。修改该函数，将这种形式的+或-符号重新写回到输入流中。

**练习 5-2** 模仿函数getint 的实现方法，编写一个读取浮点数的函数getfloat。

getfloat函数的返回值应该是什么类型？

**练习 5-3** 用指针方式实现第2 章中的函数strcat。函数strcat(s, t)将t 指向

的字符串复制到s指向的字符串的尾部。

**练习 5-4** 编写函数 strend(s, t)。如果字符串t 出现在字符串s 的尾部，该函数

返回1；否则返回0。

**练习 5-5** 实现库函数strncpy、strncat 和strncmp，它们最多对参数字符串中的前n 个字符进行操作。例如，函数strncpy(s, t, n)将t 中最多前n 个字符复制到s中。更详细的说明请参见附录B。

**练习 5-6** 采用指针而非数组索引方式改写前面章节和练习中的某些程序，例如getline（第1、4 章），atoi、itoa以及它们的变体形式（第2、3、4 章），reverse（第3 章），strindex、getop（第4 章）等等。

**练习 5-7** 重写函数readlines，将输入的文本行存储到由main 函数提供的一个数组中，而不是存储到调用alloc分配的存储空间中。该函数的运行速度比改写前快多少？

**练习 5-8** 函数 day\_of\_year和month\_day中没有进行错误检查，请解决该问题。

练习 5-9 用指针方式代替数组下标方式改写函数day\_of\_year和month\_day。

**练习 5-10** 编写程序 expr，以计算从命令行输入的逆波兰表达式的值，其中每个运算

符或操作数用一个单独的参数表示。例如，命令expr 2 3 4 + \* 将计算表达式2 × (3 + 4)的值。

**练习 5-11** 修改程序entab和decab（第1 章练习中编写的函数），使它们接受一组作

为参数的制表符停止位。如果启动程序时不带参数，则使用默认的制表符停止位设置。

**练习 5-12** 对程序entab和detab的功能做一些扩充，以接受下列缩写的命令：

entab –*m* +*n* 表示制表符从第*m*列开始，每隔*n* 列停止。选择（对使用者而言）比较方便的默认行为。

**练习 5-13** 编写程序tail，将其输入中的最后*n* 行打印出来。默认情况下，*n* 的值为

10，但可通过一个可选参数改变*n* 的值，因此，命令 tail –*n* 将打印其输入的最后*n* 行。无论输入或*n* 的值是否合理，该程序都应该能正常运行。编写的程序要充分地利用存储空间；输入行的存储方式应该同5.6 节中排序程序的存储方式一样，而不采用固定长度的二维数组。

**练习 5-14** 修改排序程序，使它能处理-r 标记。该标记表明，以逆序（递减）方式排

序。要保证-r和-n能够组合在一起使用。

**练习 5-15** 增加选项-f，使得排序过程不考虑字母大小写之间的区别。例如，比较a

和A时认为它们相等。

**练习 5-16** 增加选项-d（代表目录顺序）。该选项表明，只对字母、数字和空格进行比

较。要保证该选项可以和-f组合在一起使用。

**练习 5-17** 增加字段处理功能，以使得排序程序可以根据行内的不同字段进行排序，每

个字段按照一个单独的选项集合进行排序。（在对本书索引进行排序时，索引条目使用了-df 选项，而对页码排序时使用了-n选项。）

**练习5-18** 修改dcl程序，使它能够处理输入中的错误。

**练习 5-19** 修改undcl 程序，使它在把文字描述转换为声明的过程中不会生成多余的

圆括号。

**练习 5-20** 扩展 dcl 程序的功能，使它能够处理包含其它成分的声明，例如带有函数

参数类型的声明、带有类似于const限定符的声明等。

**练习6-1** 上述getword 函数不能正确处理下划线、字符串常量、注释及预处理器控

制指令。请编写一个更完善的getword函数。

**练习 6-2** 编写一个程序，用以读入一个C语言程序，并按字母表顺序分组打印变量名，要求每一组内各变量名的前6个字符相同，其余字符不同。字符串和注释中的单词不予考虑。

请将6 作为一个可在命令行中设定的参数。

**练习 6-3** 编写一个交叉引用程序，打印文档中所有单词的列表，并且每个单词还有一

个列表，记录出现过该单词的行号。对the、and等非实义单词不予考虑。

**练习 6-4** 编写一个程序，根据单词的出现频率按降序打印输入的各个不同单词，并在

每个单词的前面标上它的出现次数。

**练习6-5** 编写函数 undef，它将从由lookup和install维护的表中删除一个变量

及其定义。

**练习 6-6** 以本节介绍的函数为基础，编写一个适合C 语言程序使用的#define 处理

器的简单版本（即无参数的情况）。你会发现getch和ungetch函数非常有用。

**练习 7-1** 编写一个程序，根据它自身被调用时存放在argv[0]中的名字，实现将大

写字母转换为小写字母或将小写字母转换为大写字母的功能。

**练习 7-2** 编写一个程序，以合理的方式打印任何输入。该程序至少能够根据用户的习

惯以八进制或十六进制打印非图形字符，并截断长文本行。

**练习7-3** 改写minprintf函数，使它能完成printf函数的更多功能。

**练习 7-4** 类似于上一节中的函数minprintf，编写scanf函数的一个简化版本。

**练习 7-5** 改写第4 章中的后缀计算器程序，用scanf 函数和（或）sscanf 函数实现输入以及数的转换。

**练习7-6** 编写一个程序，比较两个文件并打印它们第一个不相同的行。

**练习7-7** 修改第5章的模式查找程序，使它从一个命名文件的集合中读取输入（有文

件名参数时），如果没有文件名参数，则从标准输入中读取输入。当发现一个匹配行时，是否应该将相应的文件名打印出来？

**练习 7-8** 编写一个程序，以打印一个文件集合，每个文件从新的一页开始打印，并且

打印每个文件相应的标题和页数。

**练习 7-9** 类似于isupper 这样的函数可以通过某种方式实现以达到节省空间或时间的目的。考虑节省空间或时间的实现方式。

**练习 8-1** 用read、write、open和close系统调用代替标准库中功能等价的函数，重写第7 章的cat程序，并通过实验比较两个版本的相对执行速度。

**练习 8-2** 用字段代替显式的按位操作，重写fopen 和\_fillbuf 函数。比较相应代码的长度和执行速度。

**练习 8-3** 设计并编写函数\_flushbuf、fflush和fclose。

**练习 8-4** 标准库函数int fseek(FILE \*fp, long offset, int origin)类似于函数lseek，所不同的是，该函数中的fp是一个文件指针而不是文件描述符，且返回值是一个int类型的状态而非位置值。编写函数fseek，并确保该函数与库中其它函数使用的缓冲能够协同工作。

**练习 8-5** 修改fsize程序，打印i结点项中包含的其它信息。

**练习 8-6** 标准库函数 calloc(n, size)返回一个指针，它指向n 个长度为size

的对象，且所有分配的存储空间都被初始化为0。通过调用或修改malloc函数来实现calloc函数。

**练习 8-7** malloc 接收对存储空间的请求时，并不检查请求长度的合理性；而free

则认为被释放的块包含一个有效的长度字段。改进这些函数，使它们具有错误检查的功能。

**练习 8-8** 编写函数 bfree(p, n)，释放一个包含n个字符的任意块p，并将它放入

由malloc 和free 维护的空闲块链表中。通过使用bfree，用户可以在任意时刻向空闲块链表中添加一个静态或外部数组。