

1. 删数问题

【问题描述】输入一个高精度的大正整数 S (S 最长可达 240 位)，去掉其中任意 N 位数字后剩下的数字按原次序组成一个新的正整数 S' 。编程对给定的 N 和 S ，寻找一种方案使得剩下的数字组成的新数 S' 最小。

【输入形式】输入有两行：

1. 第一行是大整数 S 。其中 S 最长可达 240 位。
2. 第二行是整数 N 。 S 、 N 均以非 0 数字开头。

【输出形式】输出有一行，是在 S 中删除 N 位后所得的最小数字 S' 。

【样例输入 1】

178543

4

【样例输出 1】 13

【样例输入 2】

1002

1

【样例输出 2】 002

【样例说明】样例 1 中输入整数 $S=178543$ ， $N=4$ ，要求在 178543 中删除 4 位，使剩下的数字最小。正确答案为 $S'=13$ 。样例 2 中输入整数 $S=1002$ ， $N=1$ ，删完一位后 $S'=002$ ，而不是 2，即 2 之前的 0 也必须输出。

【运行时限】程序一次运行的最长时间限制在 15 秒内，超出则认为程序错误。

【算法提示】将整数看作字符串形式读入；删数时，从前往后寻找第一个比后一个数字大的数字，然后删除之，按照这种方法删除 N 个数字即得最小数字。

【评分标准】该题要求输出一个大整数的各位数字。结果完全正确得 20 分，每个测试点 4 分。上传 C 语言文件名为 `delete.c`。

2. 扩展字符

【问题描述】

编写一函数 `expand(s1,s2)`，用以将字符串 $s1$ 中的缩记符号在字符串 $s2$ 中扩展为等价的完整字符，例如将 `a-d` 扩展为 `abcd`。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。在 `main` 函数中测试该函数：从键盘输入包含缩记符号的字符串，然后调用该函数进行扩展，输出扩展结果。

(教材 P63 : Exercise 3-3)

注意：

1. 待扩展字符串中有可能包含空格，例如：`a-d x-z` 应扩展成：`abcd xyz`。所以读入待扩展字符串时，应能够读入包含空格的字符串。
2. 只要缩记符号之后的字符比之前的字符的 ASCII 码值大，就要将它们之间的所有字符扩展出来，例如：`Z-a` 之间的字符也要扩展出来；
3. 特殊情况：`a-b-c` 将被扩展为：`abc`。`a-a` 将被扩展为：`a-a`。

【输入形式】

从键盘输入包含扩展符的字符串

【输出形式】

输出扩展后的字符串

【输入样例】

a-c-u-B

【输出样例】

abcdefghijklmnopqrstu-B

【样例说明】

扩展输入 a-c-u 为：abcdefghijklmnopqrstu ，而 B 比 u 值小，所以无法扩展，直接输出。

【评分标准】

结果完全正确得 20 分，每个测试点 4 分，提交程序文件 `expand.c` 。

3. [字母频率统计](#)

【问题描述】

编写程序从标准输入中读入一段英文，统计其中小写字母出现次数，并以柱状图的形式显示其出现次数。

【输入形式】

在标准输入上输入一段英文文章（可能有一行，也可能有多行），在新的一行的开头输入 `ctrl+z` 键表示结束。

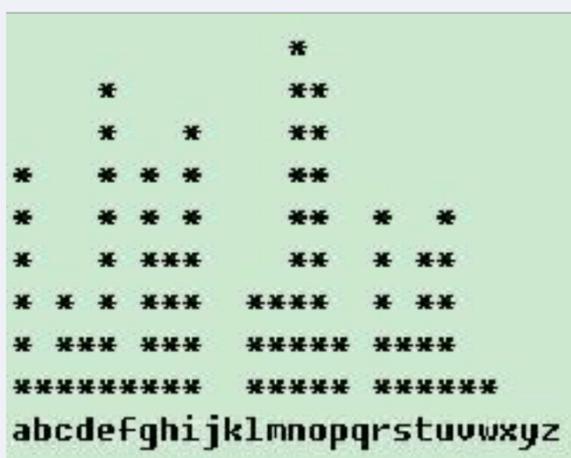
【输出形式】

在屏幕上依次输出表示每个小写字母出现次数的柱状图（以“*”符号表示柱状图，空白处用空格字符表示，某个小写字母出现多少次，就显示多少“*”符号；柱状图的高度以出现最多的字母次数为准），在最后一行依次输出 26 个小写字母。

【样例输入】

The computing world has undergone a
revolution since the publication of
The C Programming Language in 1978.

【样例输出】



【样例说明】

在输入的英文短文中，小写字母 a 出现了 6 次，所以其上输出了 6 个字符 。出现次数最多的是字母 n，所以柱状图的高度为 9 个字符。字母 j 没有出现，所以其上都为空格字符。

【评分标准】

该题要求输出柱状图表示的字母出现次数，共有 5 个测试点。上传 C 语言文件名为 `bar.c` 。

4. [矩阵运算](#)

【问题描述】

对于多个 N 阶矩阵，依次进行加、减运算。

【输入形式】

从标准输入读取输入。第一行只有一个整数 N ($1 \leq N \leq 10$)，代表矩阵的阶数。

接下来是一个矩阵，是 N 行，每行有 N 个整数（可能是正、负整数），是矩阵的所有元素。

然后一行只含一个字符“+或-”，代表加、减操作。

然后用同样的方式输入另一个矩阵。

后续仍然是运算符和矩阵。直至运算符为“#”时停止计算，将结果输出。

【输出形式】

向标准输出打印矩阵的操作结果。输出 N 行，每行对应矩阵在该行上的所有元素，每一行末均输出一个回车符。每个元素占 5 个字符宽度（包括负号），向右对齐，不足部分补以空格。

【输入样例】

```
3
1 -2 7
2 8 -5
3 6 9
+
3 5 7
-1 2 6
3 7 10
-
1 -2 7
2 8 -5
3 6 9
#
```

【输出样例】

（下图中“ ”代表空格）

```
####3####5####7
###-1####2####6
####3####7###10
```

【评分标准】

本题不准使用数学库函数。运行时限 1 秒，完全正确 20 分，每个测试点 4 分。提交程序文件名为 `matrix.c`。

5. [文件拷贝 2](#)**【问题描述】**

写一程序将一个文件 `fcopy.in` 拷贝至另一个文件 `fcopy.out`，其中在所拷贝的文件中，多个连续空白符（包括空格符、制表符）只拷贝一个空格符，其它字符不变。

【输入形式】

源文件名和目标文件名分别为 `fcopy.in` 和 `fcopy.out`，程序将从当前目录下读取 `fcopy.in` 文件。

【输出形式】

将 `fcopy.in` 文件内容拷贝至当前目录下的 `fcopy.out` 文件中。在所拷贝的文件中，多个连续空白符（包括空格符、制表符）只拷贝一个空格符，若非空白符之间有一个制表符，则该制表符也要替换为空格符，其它字符不变。

【输入样例】

假如文件 `fcopy.in` 中内容如下：

`Alcatel provides end-to-end solutions.`

【输出样例】

输出文件 `fcopy.out` 中内容为：

`Alcatel provides end-to-end solutions.`

【样例说明】

将文件 `fcopy.in` 拷贝到 `fcopy.out`，同时做适当的转换。

【评分标准】

其中在所拷贝的文件中，多个连续空白符（包括空格符、制表符）只拷贝一个空格符，其它字符不变，完全符合要求得 20 分，每个测试点 4 分。提交程序名为 `copy.c`。

6. [括号匹配](#)

【问题描述】

假设一个输入字符串中包含圆括号、方括号和花括号三种类型的括号，以及其它一些任意字符。编写程序，判别串中的括号是否正确匹配，即：

1. 各种左、右括号的个数要一致；
2. 不能先出现右括号；
3. 其它规则暂不考虑，例如：`(ad[ce)ef]` 认为是正确的。

【输入形式】

从当前目录下 `correct.in` 文件中读入一行字符串。字符串最大长度 80，不含空格。

【输出形式】

输出到当前目录下 `correct.out` 文件中。输出只有一个单词，如果括号匹配则输出“True”到文件中，否则输出“False”。在输出末尾要有一个回车符。

【输入样例】

设输入文件内容如下：

`rhe+[35(fje)w-wr3f[efe{feofds}]`

【输出样例】

输出文件内容为：

`False`

【样例说明】

输入字符串为 `rhe+[35(fje)w-wr3f[efe{feofds}]`，在式中“{”“}”个数不一致，不符合嵌套规则，故输出为“False”。

【评分标准】

结果正确得 20 分，每个测试点 4 分，提交程序名为 `c0802.c`。

1. [凸多边形面积](#)

【问题描述】 给出平面上一组顶点的坐标，计算出它们所围成的凸多边形的面积。

【输入形式】从标准输入读取顶点坐标。格式为：第一行是点的个数 N ($3 \leq N \leq 15$)，后面紧接着 N 行，每行两个数字（由空格隔开），分别表示该点的 X 、 Y 坐标 ($0 \leq X, Y \leq 32767$)。所有点的坐标互不相同，且按顺时针次序给出。

输入数据确保该多边形是一个凸多边形。

【输出形式】向标准输出打印一个浮点数，是该多边形的面积。该浮点数保留两位小数。

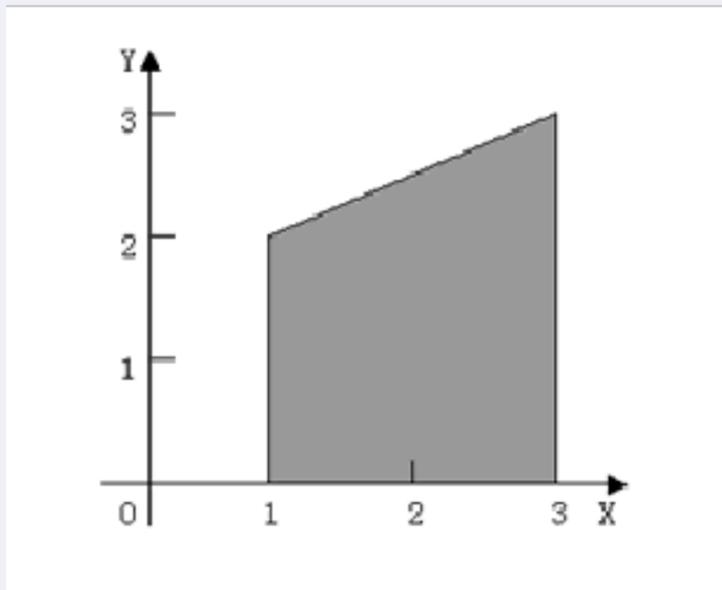
【输入样例】 4

3 3

3 0

1 0

1 2



【输出样例】

5.00

【样例说明】输入数据表示了如图所示的四边形。其面积为 5.00。

提示：求三角形面积可用海伦公式，求平方根可用 `<math.h>` 头文件中定义的 `sqrt` 函数。

【评分标准】结果完全正确得 20 分，每个测试点 4 分。提交程序名为：`points.c`。

2. [整数的 N 进制字符串表示](#)

【问题描述】编写函数 `itob(n,s,b)`，用于把整数 n 转换成以 b 为基的字符串并存储到 s 中。编写程序，使用函数 `itob(n,s,b)` 将输入的整数 n ，转换成字符串 s ，将 s 输出。转换后的字符串从最高的非零位开始输出。如果 n 为负数，则输出的字符串的第一个字符为 `'-`。 b 为大于 1 小于 37 的任意自然数值。当 $b=2$ 时，输出字符只可能是 `'0'` `'1'`；当 $b=16$ 时，输出字符串中可能含有字符为 `'0'` `'9'` `'a'` `'f'` (字母以小写输出)。 b 还可以是其它数值。比如输入 $n=33, b=17$ ，则输出 33 的 17 进制值为 。

【输入形式】控制台输入整数 n 和 b ，其中 n 可以为负数。 n 和 b 以空格分隔。

【输出形式】控制台输出转化后的字符串 s 。

【样例输入】 5 2

【样例输出】 101

【样例说明】5 的二进制就是 101

【评分标准】结果完全正确得 20 分，每个测试点 4 分。提交程序名为：`itob.c`

3. [求两组整数的异或集](#)

【问题描述】

从标准输入中输入两组整数(每行不超过 20 个整数，每组整数中元素不重复)，合并两组整数，去掉在两组整数中都出现的整数，并按从大到小顺序排序输出（即两组整数集“异或”）。

【输入形式】

首先输入第一组整数，以一个空格分隔各个整数；然后在新的一行上输入第二组整数，以一个空格分隔，行末有回车换行。

【输出形式】

按从大到小顺序排序输出合并后的整数集（去掉在两组整数中都出现的整数，以一个空格分隔各个整数）。

【样例输入】

```
5 1 4 32 8 7 9 -6
5 2 87 10 1
```

【样例输出】

```
87 32 10 9 8 7 4 2 -6
```

【样例说明】

第一组整数为 5 1 4 32 8 7 9 -6 第二组整数分别为 5 2 87 10 1。将第一组和第二组整数合并（去掉在两组整数中都出现的整数 5 和 1），并从大到小顺序排序后结果为 87 32 10 9 8 7 4 2 -6。

【评分标准】该题要求输出两组整数的异或集，共有 5 个测试点，提交程序文件名为 `xor.c`。

4. [字符串中字符排序](#)

【问题描述】编写一个程序，从键盘接收一个字符串，然后按照字符顺序从小到大进行排序，并删除重复的字符。

【输入形式】用户在第一行输入一个字符串。

【输出形式】程序按照字符(ASCII)顺序从小到大排序字符串，并删除重复的字符进行输出。

【样例输入】badacgegfacb

【样例输出】abcdefg

【样例说明】用户输入字符串 badacgegfacb，程序对其进行按从小到大(ASCII)顺序排序，并删除重复的字符，最后输出为 abcdefg

【评分标准】结果完全正确得 20 分，每个测试点 4 分。提交源程序名为 `stringsort.c`

5. [超长正整数的减法](#)

【问题描述】

编写程序实现两个超长正整数（每个最长 80 位数字）的减法运算。

【输入形式】

从键盘读入两个整数，要考虑输入高位可能为 0 的情况（如 00083）。

1. 第一行是超长正整数 A;
2. 第二行是超长正整数 B;

【输出形式】

输出只有一行，是长整数 A 减去长整数 B 的运算结果，从高到低依次输出各位数字。要求：若结果为 0，则只输出一个 0；否则输出的结果的最高位不能为 0，并且各位数字紧密输出。

【输入样例】

234098
134098703578230056

【输出样例】

-134098703577995958

【样例说明】

进行两个正整数减法运算， $234098 - 134098703578230056 = -134098703577995958$ 。

【评分标准】

完全正确得 20 分，每个测试点 4 分，提交程序文件名为 `subtract.c`。

6. [字符串替换（新）](#)

【问题描述】

编写程序将一个指定文件中某一字符串替换为另一个字符串。要求：（1）被替换字符串若有多个，均要被替换；（2）指定的被替换字符串，大小写无关。

【输入形式】

给定文件名为 `filein.txt`。从控制台输入两行字符串（不含空格，行末尾都有回车换行符），分别表示被替换的字符串和替换字符串。

【输出形式】

将替换后的结果输出到文件 `fileout.txt` 中。

【样例输入】

从控制台输入两行字符串：

in

out

文件 `filein.txt` 的内容为：

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
    FILE * IN;
```

```
    {
```

```
        ,
```

```
        return;
```

```
    }
```

```
    fclose(IN);
```

```
}
```

【样例输出】

文件 `fileout.txt` 的内容应为：

```
#outclude <stdio.h>
```

```
void maout()
```

```
{
```

```
    FILE * out;
```

```

    {
        ,

    return;
    }
fclose(out);
}

```

【样例说明】

输入的被替换字符串为 `in`，替换字符串为 `out`，即将文件 `filein.txt` 中的所有 `in` 字符串（包括 `iN`、`In`、`IN` 字符串）全部替换为 `out` 字符串，并输出保存到文件 `fileout.txt` 中。

【评分标准】

该题要求得到替换后的文件内容，共有 5 个测试点。上传 C 语言文件名为 `replace.c`。

7. 最长升序子串（选做，不计分）

【问题描述】 输入一行字符串，该字符串只由小写英文字母 `a-z` 组成，且其中的字符可以重复，最长不超过 10000 个字符。

从该字符串中按顺序挑选出若干字符（不一定相邻）组成一个新串，称为“子串”。如果子串中每两个相邻的字符或者相等，或者后一个比前一个大，则称为“升序子串”。编程求出输入字符串的最长升序子串的长度。

例如，由输入字符串 `abdbch` 可以构成的升序子串有：`abd`、`abch`、`bbch`、`abbch` 等。其中最长的升序子串是 `abbch`，其长度为 5。

【输入形式】 从标准输入读取一行字符串，该串不含空格，以回车符结束。

【输出形式】 向标准输出打印一个正整数，是字符串中最长的升序子串的长度，在行末要输出一个回车符。

【输入样例】 `abdbch`

【输出样例】 `5`

【样例说明】 `abdbch` 中最长子串是 `abbch`，长度是 5。

【评分标准】 结果完全正确得 20 分，每个测试点 4 分。上传 c 语言源程序为 `up.c`。

1. **合并字符串** **【问题描述】**

编写一个函数 `char * str_bin(char* str1, char* str2)`，`str1`、`str2` 是两个有序字符串（其中字符按 ASCII 码从小到大排序），将 `str2` 合并到字符串 `str1` 中，要求合并后的字符串仍是有序的，允许字符重复。在 `main` 函数中测试该函数：从键盘输入两个有序字符串，然后调用该函数，最后输出合并后的结果。

【输入形式】

分行从键盘输入两个有序字符串（不超过 100 个字符）

【输出形式】

输出合并后的有序字符串

【输入样例】

`aceg`

`bdfh`

【输出样例】

`abcdefgh`

【样例说明】

输入两个有序字符串 `aceg` 和 `bdfh`，输出合并后的有序字符串 `abcdefgh`

【评分标准】

结果完全正确得 20 分，每个测试点 4 分，提交程序文件名为 `combine.c`。

2.删除子串【问题描述】编写一个程序，当在一个字符串中出现子串时就删除它。

【输入形式】用户在第一行输入一个字符串，用户在第二行输入一个子串。

【输出形式】程序在下一行输出删除其中所有子串后的字符串。如果字符串不包含子串则输出原字符串本身。

【样例输入】

I am a boy!

a

【样例输出】

I m boy!

【样例说明】用户首先输入字符串 I am a boy!，然后输出子串 a，程序会寻找字符串中的子串删除它，最后 将删除后的结果输出:I#m##boy! #表示空格。

【评分标准】结果完全正确得 20 分，每个测试点 4 分。提交程序名为：delsubstring.c 。

3.小数形式与科学计数法转换（简）【问题描述】

编写一个程序，将用小数表示的浮点数，转换成科学计数法的形式输出。输入的数据没有符号，小数点前后必有数字，且全为有效数据，即小数点后的末尾数字不为 0；小数点前若只有一位数字，可以为 0，否则小数点前的最高位数字不为 0。

提示：以字符串形式保存相关数据。

【输入形式】

从控制台输入一小数，最后有回车换行符，所有输入的字符数不会超过 100。

【输出形式】

以科学计数法形式输出数据。输出的数据由以下几部分构成：

1.底数部分是一个小数或整数，若为小数，则小数点前后必有数字，而且都为有效数字。即：小数点前只有一位大于 0 的数字，小数点后的末尾数字不能为 0。若为整数，则只有一位数字，不带小数点。

2.必有小写字母“e”。

3.指数部分是一个整数，若大于等于 0，则不带正号“+”若小于 0，则需要带负号“-”，且整数的最高位数字不为 0。

【输入样例 1】

0.000000000000002

【输出样例 1】

2e-15

【输入样例 2】

8.9845623489651700659

【输出样例 2】

8.9845623489651700659e0

【输入样例 3】

367298599999093453490394859509568659795603.4

【输出样例 3】

3.672985999990934534903948595095686597956034e41

【样例说明】

以小数形式输入数据，然后转换成科学计数法形式输出。

【评分标准】

该题要求以科学计数法形式输出数据，提交程序文件名为 notation.c 。

4.旋转魔方阵(文件)【问题描述】

), 要求输出如下的魔方阵, 即边长为 $N*N$, 元素取值为 1 至 $N*N$, 1 在左上角, 呈顺时针方向依次放置各元素。

$N=3$ 时:

```
1  2  3
8  9  4
7  6  5
```

【输入形式】

从标准输入读取一个整数 N 。

【输出形式】

将结果输出到文件文件 `file.out`。输出符合要求的方阵, 每个数字占 5 个字符宽度, 向右对齐, 在每一行末均输出一个回车符。

【输入样例】

4

【输出样例】 输出文件 `file.out` 内容为:

```
1  2  3  4
12 13 14 5
11 16 15 6
10  9  8  7
```

【评分标准】

本题不准使用数学库函数。结果正确得 20 分, 每个测试点 4 分, 提交程序文件名为 `magic.c`。

5. 全排列数的生成 **【问题描述】** 输入整数 N , 生成从 1~ N 所有整数的全排列。

【输入形式】 输入整数 N 。

【输出形式】 输出有 $N!$ 行, 每行都是从 1~ N 所有整数的一个全排列, 各整数之间以空格分隔。各行上的全排列不重复。输出各行遵循“小数优先”原则, 在各全排列中, 较小的数尽量靠前输出。如果将每行上的输出看成一个数字, 则所有输出构成升序数列。具体格式见输出样例。

【样例输入 1】 1

【样例输出 1】 1

【样例说明 1】 输入整数 $N=1$, 其全排列只有一种。

【样例输入 2】 3

【样例输出 2】

```
1 2 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1
```

【样例说明 2】 输入整数 $N=3$, 要求整数 1、2、3 的所有全排列, 共有 $N!=6$ 行。且先输出 1 开头的的所有排列数, 再输出 2 开头的的所有排列数, 最后输出 3 开头的的所有排列数。在以 1 开头的的所有全排列中同样遵循此原则。

【样例输入 3】 10

【样例输出 3】

```
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 10 9
1 2 3 4 5 6 7 9 8 10
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/178134067010006106>