

1 - 判断可逆素数

封装实现判断素数的函数跟逆置的函数即可。

2 - 矩阵相交

编程实现提示给的思路即可：

输入的两点可以是矩形任一对角线上的端点，求相交的面积可以先求矩形在 X 轴和 Y 轴上的交集。

矩形在 X 轴上的交集可以按照如下算法进行求解：

假设 AX1 和 AX2 中的较大值为 MAX_AX，较小值为 MIN_AX；BX1 和 BX2 中的较大值为 MAX_BX，较小值为 MIN_BX。用 MAX_AX 和 MAX_BX 中的较小者减去 MIN_AX 和 MIN_BX 中的较大者，结果为正表示两矩形在 X 轴上的交集，若为负则表示不相交。

Y 方向其实也同理。

3 - 求差集

先把集合 A 读进来，并求出 A 中元素的个数 `cnt`。读入 B 集合中的元素 x 时，遍历 A，如果 `A[i] = x`，则标记 `A[i] = -1`。最后输出 A 中所有不等于 -1 的元素。

4 - 矩阵运算

读入符号和进行矩阵计算的过程没什么特殊的，最后右对齐补空格要使用格式化字符串 `%5d`（如果在前面补 0 则是 `%05d`）。

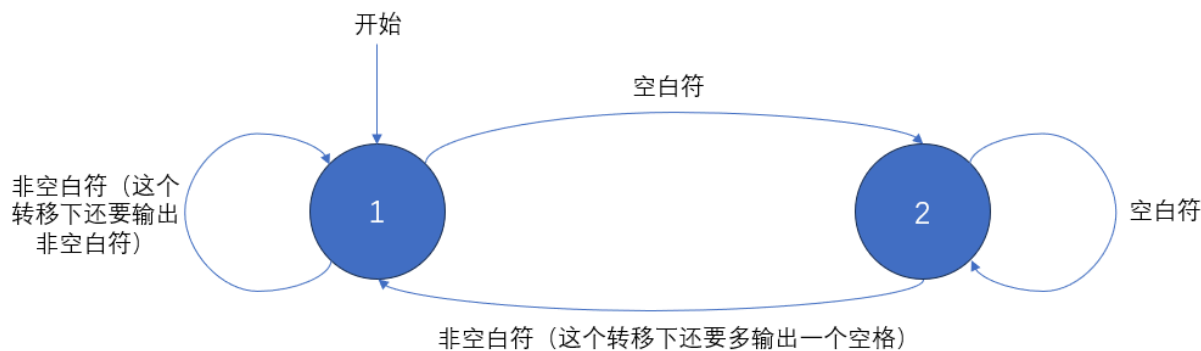
5 - 文件拷贝 2

我们可以用状态转移的思想来解决这个问题，状态转移思想的关键就在于**定义状态以及状态之间的转移**。

在逐个字符处理的过程中，我们需要定义以下两个状态：

- **非空白状态 (状态 1)**：当我们遇到非空白符，我们进入这个状态。
- **空白状态 (状态 2)**：当我们遇到空白符（包括空格符、制表符等），我们进入这个状态；在这个状态下，我们连续遇到多个空白符时，依然保持在这个状态。

两个状态之间的转移过程如下：



然后编程实现这张图即可，这种状态转移的思想在文本格式化、输入验证等题目中还是比较常用的。

文件读入与输出可以参考 [选填板子](#) 下的 [文件](#) 部分，里面有一些代码示例，在这里为了方便我就直接使用 `freopen` 了。

6 - 求两组整数的异或集

这题其实有一个偷懒的办法：既然输入保证了 [每组整数中元素不重复](#)，那我们直接把两组整数读进一个数组，最后看看对每个元素数组中有没有相同元素就可以，如果有的话将两个数字都删除掉。

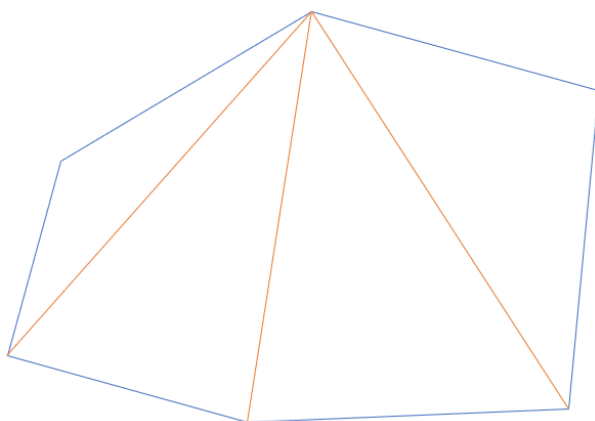
如果想真正区分两组整数的话也可以，可以使用 `scanf("%d%c", &a, &ch)` 进行读入，当发现 `ch` 不是空格时就说明一组数已经读完了。

删除的时候可以使用打标记的方法，而不真正删除。由于还涉及到一个排序操作，所以我们可以最后再去重。

代码实现使用了偷懒的那种办法，别忘了不定组输入如何结束输入（Windows 下是 `ctrl + z`，Linux 和 Mac 是 `ctrl + d`）。

7 - 凸多边形面积

可以将凸多边形拆成若干三角形，求解每个三角形面积后求和，如图所示：



求三角形面积可以用海伦公式或者叉乘，代码实现用的是叉乘的方法。

8 - 整数的N进制字符串表示

输入如果是个负数，先打个标记转换为正数，然后再进行一个简单的进制转换就可以了。