

CCF 全国信息学奥林匹克联赛（NOIP2003）复赛

普及组

（请选手务必仔细阅读本页内容）

一. 题目概况

中文题目名称	乒乓球	数字游戏	栈	麦森数
英文题目与子目录名	table	game	stack	mason
可执行文件名	table	game	stack	mason
输入文件名	table.in	game.in	stack.in	mason.in
输出文件名	table.out	game.out	stack.out	mason.out
每个测试点时限	1 秒	1 秒	1 秒	1 秒
测试点数目	10	10	10	10
每个测试点分值	10	10	10	10
附加样例文件	有	有	有	有
结果比较方式	全文比较（过滤行末空格及文末回车）			
题目类型	传统	传统	传统	传统

二. 提交源程序文件名

对于 C++语言	table.cpp	game.cpp	stack.cpp	mason.cpp
对于 C 语言	table.c	game.c	stack.c	mason.c
对于 pascal 语言	table.pas	game.pas	stack.pas	mason.pas

三. 编译命令（不包含任何优化开关）

对于 C++语言	g++ -o table table.cpp -lm	g++ -o game game.cpp -lm	g++ -o stack stack.cpp -lm	g++ -o mason mason.cpp -lm
对于 C 语言	gcc -o table table.c -lm	gcc -o game game.c -lm	gcc -o stack stack.c -lm	gcc -o mason mason.c -lm
对于 pascal 语言	fpc table.pas	fpc game.pas	fpc stack.pas	fpc mason.pas

四. 运行内存限制

内存上限	128M	128M	128M	128M
------	------	------	------	------

注意事项:

- 1、文件名（程序名和输入输出文件名）必须使用英文小写。
- 2、C/C++中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 0。
- 3、全国统一评测时采用的机器配置为：CPU Intel Core2 Quad Q8200 2.33GHz，内存 2G，上述时限以此配置为准。
- 4、特别提醒：评测在 **NOI Linux** 下进行。

1、乒乓球（`table.cpp/c/pas`）

【问题背景】国际乒联现在主席沙拉拉自从上任以来就立志于推行一系列改革，以推动乒乓球运动在全球的普及。其中 11 分制改革引起了很大的争议，有一部分球员因为无法适应新规则只能选择退役。华华就是其中一位，他退役之后走上了乒乓球研究工作，意图弄明白 11 分制和 21 分制对选手的不同影响。在开展他的研究之前，他首先需要对他多年比赛的统计数据进行一些分析，所以需要你的帮忙。

【问题描述】华华通过以下方式进行分析，首先将比赛每个球的胜负列成一张表，然后分别计算在 11 分制和 21 分制下，双方的比赛结果（截至记录末尾）。

比如现在有这么一份记录，（其中 W 表示华华获得一分，L 表示华华对手获得一分）：

WWWWWWWWWWWWWWWWWWWWLW

在 11 分制下，此时比赛的结果是华华第一局 11 比 0 获胜，第二局 11 比 0 获胜，正在进行第三局，当前比分 1 比 1。而在 21 分制下，此时比赛结果是华华第一局 21 比 0 获胜，正在进行第二局，比分 2 比 1。如果一局比赛刚开始，则此时比分为 0 比 0。

你的程序就是要对于一系列比赛信息的输入（WL 形式），输出正确的结果。

【输入格式】每个输入文件包含若干行字符串（每行至多 20 个字母），字符串有大写的 W、L 和 E 组成。其中 E 表示比赛信息结束，程序应该忽略 E 之后的所有内容。

【输出格式】输出由两部分组成，每部分有若干行，每一行对应一局比赛的比分（按比赛信息输入顺序）。其中第一部分是 11 分制下的结果，第二部分是 21 分制下的结果，两部分之间由一个空行分隔。

【输入样例】

WWWWWWWWWWWWWWWWWWWW
WWLWE

【输出样例】

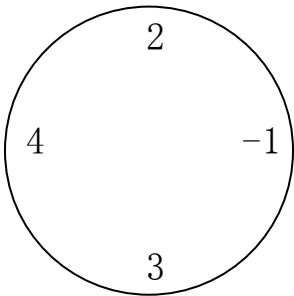
11:0
11:0
1:1

21:0
2:1

2、数字游戏 (game.cpp/c/pas)

【问题描述】丁丁最近沉迷于一个数字游戏之中。这个游戏看似简单，但丁丁在研究了许多天之后却发现原来在简单的规则下想要赢得这个游戏并不那么容易。游戏是这样的，在你面前有一圈整数（一共 n 个），你要按顺序将其分为 m 个部分，各部分内的数字相加，相加所得的 m 个结果对 10 取模后再相乘，最终得到一个数 k 。游戏的要求是使你所得的 k 最大或者最小。

例如，对于下面这圈数字 ($n=4, m=2$):



当要求最小值时， $((2-1) \bmod 10) \times ((4+3) \bmod 10) = 1 \times 7 = 7$ ，要求最大值时，为 $((2+4+3) \bmod 10) \times (-1 \bmod 10) = 9 \times 9 = 81$ 。特别值得注意的是，无论是负数还是正数，对 10 取模的结果均为非负值。

丁丁请你编写程序帮他赢得这个游戏。

【输入格式】输入文件第一行有两个整数， n ($1 \leq n \leq 50$) 和 m ($1 \leq m \leq 9$)。以下 n 行每行有个整数，其绝对值不大于 10^4 ，按顺序给出圈中的数字，首尾相接。

【输出格式】输出文件有两行，各包含一个非负整数。第一行是你程序得到的最小值，第二行是最大值。

【输入样例】

4 2
4
3
-1
2

【输出样例】

7
81

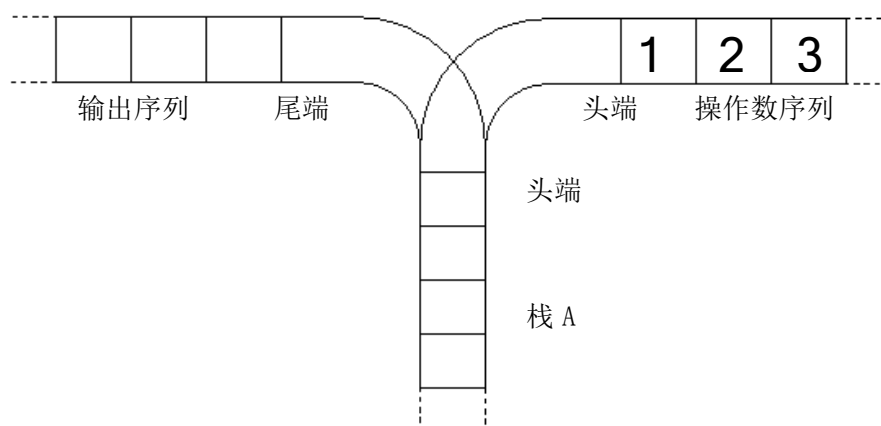
3、栈（stack.cpp/c/pas）

【问题背景】栈是计算机中经典的数据结构，简单的说，栈就是限制在一端进行插入删除操作的线性表。

栈有两种最重要的操作，即 pop（从栈顶弹出一个元素）和 push（将一个元素进栈）。

栈的重要性不言自明，任何一门数据结构的课程都会介绍栈。宁宁同学在复习栈的基本概念时，想到了一个书上没有讲过的问题，而他自己无法给出答案，所以需要你的帮忙。

【问题描述】

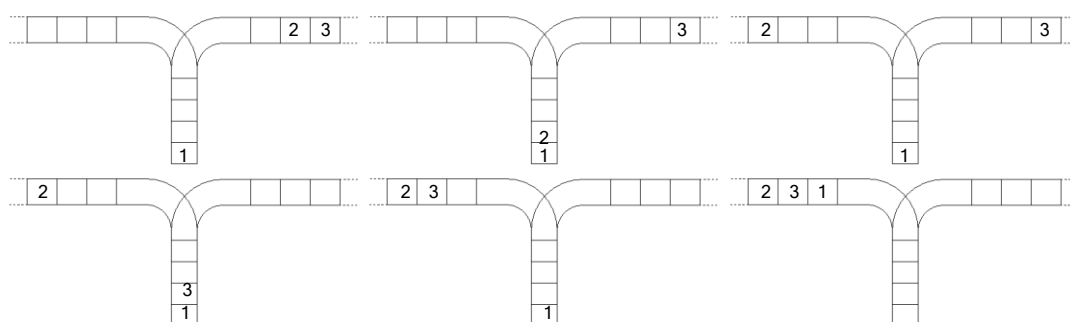


宁宁考虑的是这样一个问题：一个操作数序列，从 1，2，一直到 n（图示为 1 到 3 的情况），栈 A 的深度大于 n。

现在可以进行两种操作，

1. 将一个数，从操作数序列的头端移到栈的头端（对应数据结构栈的 push 操作）
2. 将一个数，从栈的头端移到输出序列的尾端（对应数据结构栈的 pop 操作）

使用这两种操作，由一个操作数序列就可以得到一系列的输出序列，下图所示为由 1 2 3 生成序列 2 3 1 的过程。（原始状态如上图所示）



你的程序将对给定的 n，计算并输出由操作数序列 1，2，…，n 经过操作可能得到的输出序列的总数。

【输入格式】

输入文件只含一个整数 n（ $1 \leq n \leq 18$ ）

【输出格式】

[illegible]