全国信息学奥林匹克联赛（NOIP2011）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	数字反转	统计单词数	瑞士轮	表达式的值
英文题目与子目录名	reverse	stat	swiss	exp
可执行文件名	reverse	stat	swiss	exp
输入文件名	reverse.in	stat.in	swiss.in	exp.in
输出文件名	reverse.out	stat.out	swiss.out	exp.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	reverse.cpp	stat.cpp	swiss.cpp	exp.cpp
对于 C 语言	reverse.c	stat.c	swiss.c	exp.c
对于 pascal 语言	reverse.pas	stat. pas	swiss. pas	exp.pas

三．编译命令（不包含任何优化开关）

对于 C++语言	g++ -o reverse reverse.cpp -lm	g++ -o stat stat.cpp -lm	g++ -o swiss swiss.cpp -lm	g++ -o exp exp.cpp -lm
对于 C 语言	gcc -o reverse reverse.c -lm	gcc -o stat stat.c -lm	gcc -o swiss swiss.c -lm	gcc -o exp exp.c -lm
对于 pascal 语言	fpc reverse.pas	fpc stat.pas	fpc swiss.pas	fpc exp.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 P4 3.0GHz，内存 1G，上述时限以此配置为准。

4、特别提醒：评测在 NOI Linux 下进行。

1．数字反转

(reverse.cpp/c/pas)

【问题描述】

给定一个整数，请将该数各个位上数字反转得到一个新数。新数也应满足整数的常见形式，即除非给定的原数为零，否则反转后得到的新数的最高位数字不应为零（参见样例 2）。

【输入】

输入文件名为 reverse.in。

输入共 1 行，一个整数 N。

【输出】

输出文件名为 reverse.out。

输出共 1 行，一个整数，表示反转后的新数。

【输入输出样例 1】

reverse.in	reverse.out
123	321

【输入输出样例 2】

Reverse.in	reverse.out
-380	-83

【数据范围】

-1,000,000,000≤ N≤1,000,000,000。

2．统计单词数

(stat.cpp/c/pas)

【问题描述】

一般的文本编辑器都有查找单词的功能，该功能可以快速定位特定单词在文章中的位置，有的还能统计出特定单词在文章中出现的次数。

现在，请你编程实现这一功能，具体要求是：给定一个单词，请你输出它在给定的文章中出现的次数和第一次出现的位置。**注意：匹配单词时，不区分大小写，但要求完全匹配，即给定单词必须与文章中的某一独立单词在不区分大小写的情况下完全相同（参见样例 1），如果给定单词仅是文章中某一单词的一部分则不算匹配（参见样例 2）。**

【输入】

输入文件名为 stat.in，2 行。

第 1 行为一个字符串，其中只含字母，表示给定单词；

第 2 行为一个字符串，其中只可能包含字母和空格，表示给定的文章。

【输出】

输出文件名为 stat.out。

只有一行，如果在文章中找到给定单词则输出两个整数，两个整数之间用一个空格隔开，分别是单词在文章中出现的次数和第一次出现的位置（即在文章中第一次出现时，单词首字母在文章中的位置，位置从 0 开始）；如果单词在文章中没有出现，则直接输出一个整数-1。

【输入输出样例 1】

stat.in	stat.out
To to be or not to be is a question	2 0

【输入输出样例 1 说明】

输出结果表示给定的单词 To 在文章中出现两次，第一次出现的位置为 0。

【输入输出样例 2】

stat.in	stat.out
to Did the Ottoman Empire lose its power at that time	-1

【输入输出样例 2 说明】

表示给定的单词 to 在文章中没有出现，输出整数-1。

【数据范围】

1≤单词长度≤10。

1≤文章长度≤1,000,000。

3．瑞士轮

(swiss.cpp/c/pas)

【背景】

在双人对决的竞技性比赛，如乒乓球、羽毛球、国际象棋中，最常见的赛制是淘汰赛和循环赛。前者的特点是比赛场数少，每场都紧张刺激，但偶然性较高。后者的特点是较为公平，偶然性较低，但比赛过程往往十分冗长。

本题中介绍的瑞士轮赛制，因最早使用于 1895 年在瑞士举办的国际象棋比赛而得名。它可以看作是淘汰赛与循环赛的折衷，既保证了比赛的稳定性，又能使赛程不至于过长。

【问题描述】

2*N 名编号为 1~2N 的选手共进行 R 轮比赛。每轮比赛开始前，以及所有比赛结束后，都会按照总分从高到低对选手进行一次排名。选手的总分为第一轮开始前的初始分数加上已参加过的所有比赛的得分和。总分相同的，约定编号较小的选手排名靠前。

每轮比赛的对阵安排与该轮比赛开始前的排名有关：第 1 名和第 2 名、第 3 名和第 4 名、……、第 2K – 1 名和第 2K 名、…… 、第 2N – 1 名和第 2N 名，各进行一场比赛。每场比赛胜者得 1 分，负者得 0 分。也就是说除了首轮以外，其它轮比赛的安排均不能事先确定，而是要取决于选手在之前比赛中的表现。

现给定每个选手的初始分数及其实力值，试计算在 R 轮比赛过后，排名第 Q 的选手编

号是多少。我们假设选手的实力值两两不同，且每场比赛中实力值较高的总能获胜。

【输入】

输入文件名为 swiss.in。

输入的第一行是三个正整数 N、R、Q，每两个数之间用一个空格隔开，表示有 2*N 名选手、R 轮比赛，以及我们关心的名次 Q。

第二行是 2*N 个非负整数 $s_1, s_2, \ldots, s_{2N}$，每两个数之间用一个空格隔开，其中 s_i 表示编号为 i 的选手的初始分数。

第三行是 2*N 个正整数 $w_1, w_2, \ldots, w_{2N}$，每两个数之间用一个空格隔开，其中 w_i 表示编号为 i 的选手的实力值。

【输出】

输出文件名为 swiss.out。

输出只有一行，包含一个整数，即 R 轮比赛结束后，排名第 Q 的选手的编号。

【输入输出样例】

swiss.in	swiss.out
2 4 2 7 6 6 7 10 5 20 15	1

【输入输出样例说明】

	本轮对阵	本轮结束后的得分			
选手编号	/	①	②	③	④
初始	/	7	6	6	7
第 1 轮	①—④ ②—③	7	6	7	8
第 2 轮	④—① ③—②	7	6	8	9
第 3 轮	④—③ ①—②	8	6	9	9
第 4 轮	③—④ ①—②	9	6	10	9

【数据范围】

对于 30%的数据，$1 \le N \le 100$；

对于 50%的数据，$1 \le N \le 10{,}000$；

对于 100%的数据，$1 \le N \le 100{,}000$，$1 \le R \le 50$，$1 \le Q \le 2N$，$0 \le s_1, s_2, \ldots, s_{2N} \le 10^8$，$1 \le w_1, w_2, \ldots, w_{2N} \le 10^8$。

4．表达式的值

(exp.cpp/c/pas)

【问题描述】

对于 1 位二进制变量定义两种运算：

运算符	运算规则
⊕	0 ⊕ 0=0 0 ⊕ 1=1 1 ⊕ 0=1 1 ⊕ 1=1
×	0 × 0=0 0 × 1=0 1 × 0=0 1 × 1=1

运算的优先级是：

1. 先计算括号内的，再计算括号外的。
2. **“×”运算优先于“⊕”运算**，即计算表达式时，**先计算×运算，再计算⊕运算**。
   例如：计算表达式 A ⊕ B × C 时，先计算 B × C，其结果再与 A 做 ⊕ 运算。

现给定一个未完成的表达式，例如_+(_*_)，请你在横线处填入数字 0 或者 1，请问有多少种填法可以使得表达式的值为 0。

【输入】

输入文件名为 exp.in，共 2 行。

第 1 行为一个整数 L，表示给定的表达式中**除去横线外**的运算符和括号的个数。

第 2 行为一个字符串包含 L 个字符，其中只包含’(’、’)’、’+’、’*’这 4 种字符，其中’(’、’)’是左右括号，’+’、’*’分别表示前面定义的运算符“⊕”和“×”。这行字符按顺序给出了给定表达式中除去变量外的运算符和括号。

【输出】

输出文件 exp.out 共 1 行。包含一个整数，即所有的方案数。**注意：这个数可能会很大，请输出方案数对 10007 取模后的结果。**

【输入输出样例 1】

exp.in	**exp.out**
4 +(*)	3

【输入输出样例说明】

给定的表达式包括横线字符之后为：_+(_*_)

在横线位置填入(0、0、0)、(0、1、0)、(0、0、1)时，表达式的值均为 0，所以共有 3 种填法。

【数据范围】

对于 20%的数据有 0≤L≤10。

对于 50%的数据有 0≤L≤1,000。

对于 70%的数据有 0≤L≤10,000。

对于 100%的数据有 0≤L≤100,000。

对于 50%的数据输入表达式中不含括号。