

CSP2023-J模拟赛（九）

入门组

时间：8:00–11:30

（请选手务必仔细阅读本页内容）

一. 题目概况

中文题目名称	计数问题	表达式求值	小朋友的数字	车站分级
英文题目与子目录名	count	expr	number	level
可执行文件名	count	expr	number	level
输入文件名	count.in	expr.in	number.in	level.in
输出文件名	count.out	expr.out	number.out	level.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
运行内存上限	128M	128M	128M	128M

二. 提交源程序文件名

对于 C++ 语言	count.cpp	expr.cpp	number.cpp	level.cpp
对于 C 语言	count.c	expr.c	number.c	level.c
对于 pascal 语言	count.pas	expr.pas	number.pas	level.pas

三. 编译命令（不包含任何优化开关）

对于 C++ 语言	g++ -o count count.cpp -lm	g++ -o expr expr.cpp -lm	g++ -o number number.cpp -lm	g++ -o level level.cpp -lm
对于 C 语言	gcc -o count count.c -lm	gcc -o expr expr.c -lm	gcc -o number number.c -lm	gcc -o level level.c -lm
对于 pascal 语言	fpc count.pas	fpc expr.pas	fpc number.pas	fpc level.pas

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 AMD Athlon(tm) 64x2 Dual Core CPU 5200+，2.71GHz，内存 2G，上述时限以此配置为准。
- 4、只提供 Linux 格式附加样例文件。
- 5、特别提醒：评测在 NOI Linux 下进行。

1. 记数问题

(count.cpp/c/pas)

【问题描述】

试计算在区间 1 到 n 的所有整数中，数字 x ($0 \leq x \leq 9$) 共出现了多少次？例如，在 1 到 11 中，即在 1、2、3、4、5、6、7、8、9、10、11 中，数字 1 出现了 4 次。

【输入】

输入文件名为 count.in。

输入共 1 行，包含 2 个整数 n 、 x ，之间用一个空格隔开。

【输出】

输出文件名为 count.out。

输出共 1 行，包含一个整数，表示 x 出现的次数。

【输入输出样例】

count.in	count.out
11 1	4

【数据说明】

对于 100% 的数据， $1 \leq n \leq 1,000,000$ ， $0 \leq x \leq 9$ 。

2. 表达式求值

(expr.cpp/c/pas)

【问题描述】

给定一个只包含加法和乘法的算术表达式，请你编程计算表达式的值。

【输入】

输入文件为 expr.in。

输入仅有一行，为需要你计算的表达式，表达式中只包含数字、加法运算符“+”和乘法运算符“*”，且没有括号，所有参与运算的数字均为 0 到 $2^{31}-1$ 之间的整数。输入数据保证这一行只有 0~9、+、* 这 12 种字符。

【输出】

输出文件名为 expr.out。

输出只有一行，包含一个整数，表示这个表达式的值。**注意：当答案长度多于 4 位时，请只输出最后 4 位，前导 0 不输出。**

【输入输出样例 1】

expr.in	expr.out
1+1*3+4	8

【输入输出样例 2】

expr.in	expr.out
1+1234567890*1	7891

【输入输出样例 3】

expr.in	expr.out
1+1000000003*1	4

【输入输出样例说明】

样例 1 计算的结果为 8，直接输出 8。

样例 2 计算的结果为 1234567891，输出后 4 位，即 7891。

样例 3 计算的结果为 1000000004，输出后 4 位，即 4。

【数据范围】

对于 30% 的数据， $0 \leq$ 表达式中加法运算符和乘法运算符的总数 ≤ 100 ；

对于 80% 的数据， $0 \leq$ 表达式中加法运算符和乘法运算符的总数 ≤ 1000 ；

对于 100% 的数据， $0 \leq$ 表达式中加法运算符和乘法运算符的总数 ≤ 100000 。

3. 小朋友的数字

(number.cpp/c/pas)

【问题描述】

有 n 个小朋友排成一列。每个小朋友手上都有一个数字，这个数字可正可负。规定每个小朋友的特征值等于排在他前面（包括他本人）的小朋友中连续若干个（最少有一个）小朋友手上的数字之和的最大值。

作为这些小朋友的老师，你需要给每个小朋友一个分数，分数是这样规定的：第一个小朋友的分数是他的特征值，其它小朋友的分数为排在他前面的所有小朋友中（不包括他本人），小朋友分数加上其特征值的最大值。

请计算所有小朋友分数的最大值，输出时保持最大值的符号，将其绝对值对 p 取模后输出。

【输入】

输入文件为 number.in。

第一行包含两个正整数 n 、 p ，之间用一个空格隔开。

第二行包含 n 个数，每两个整数之间用一个空格隔开，表示每个小朋友手上的数字。

【输出】

输出文件名为 `number.out`。

输出只有一行，包含一个整数，表示最大分数对 p 取模的结果。

【输入输出样例 1】

<code>number.in</code>	<code>number.out</code>
5 997 1 2 3 4 5	21

【输入输出样例说明】

小朋友的特征值分别为 1、3、6、10、15，分数分别为 1、2、5、11、21，最大值 21 对 997 的模是 21。

【输入输出样例 2】

<code>number.in</code>	<code>number.out</code>
5 7 -1 -1 -1 -1 -1	-1

【输入输出样例说明】

小朋友的特征值分别为-1、-1、-1、-1、-1，分数分别为-1、-2、-2、-2、-2，最大值 -1 对 7 的模为-1，输出-1。

【数据范围】

对于 50% 的数据， $1 \leq n \leq 1,000$ ， $1 \leq p \leq 1,000$ 所有数字的绝对值不超过 1000；

对于 100% 的数据， $1 \leq n \leq 1,000,000$ ， $1 \leq p \leq 10^9$ ，其他数字的绝对值均不超过 10^9 。

4. 车站分级

(`level.cpp/c/pas`)

【问题描述】

一条单向的铁路线上，依次有编号为 1, 2, ..., n 的 n 个火车站。每个火车站都有一个级别，最低为 1 级。现有若干趟车次在这条线路上行驶，每一趟都满足如下要求：如果这趟车次停靠了火车站 x ，则始发站、终点站之间所有级别大于等于火车站 x 的都必须停靠。（注意：起始站和终点站自然也算作事先已知需要停靠的站点）

例如，下表是 5 趟车次的运行情况。其中，前 4 趟车次均满足要求，而第 5 趟车次由于停靠了 3 号火车站（2 级）却未停靠途经的 6 号火车站（亦为 2 级）而不满足要求。

车站编号	1		2		3		4		5		6		7		8		9
车站级别 车次	3		1		2		1		3		2		1		1		3
1	始	→	→	→	停	→	→	→	停	→	终						
2					始	→	→	→	停	→	终						
3	始	→	→	→	→	→	→	→	停	→	→	→	→	→	→	→	终
4							始	→	停	→	停	→	停	→	停	→	终
5					始	→	→	→	停	→	→	→	→	→	→	→	终

现有 m 趟车次的运行情况（全部满足要求），试推算这 n 个火车站至少分为几个不同的级别。

【输入】

输入文件为 level.in。

第一行包含 2 个正整数 n, m ，用一个空格隔开。

第 $i+1$ 行 ($1 \leq i \leq m$) 中，首先是一个正整数 s_i ($2 \leq s_i \leq n$)，表示第 i 趟车次有 s_i 个停靠站；接下来有 s_i 个正整数，表示所有停靠站的编号，从小到大排列。每两个数之间用一个空格隔开。输入保证所有的车次都满足要求。

【输出】

输出文件为 level.out。

输出只有一行，包含一个正整数，即 n 个火车站最少划分的级别数。

【输入输出样例】

level.in	level.out
9 2 4 1 3 5 6 3 3 5 6	2
9 3 4 1 3 5 6 3 3 5 6 3 1 5 9	3

【数据范围】

对于 20% 的数据， $1 \leq n, m \leq 10$ ；

对于 50% 的数据， $1 \leq n, m \leq 100$ ；

对于 100% 的数据， $1 \leq n, m \leq 1000$ 。