

重庆育才信息学奥林匹克测试试题

提高组

(请选手务必仔细阅读本页内容)

一. 题目概况

中文题目名称	表达式求值	对称二叉树	信息传递	选择客栈	排水系统
英文题目名	expr	tree	message	hotel	water
可执行文件名	expr	tree	message	hotel	water
输入文件名	expr.in	tree.in	message.in	hotel.in	Water.in
输出文件名	expr.out	tree.out	message.out	hotel.out	Water.out
时限	0.5 秒	0.5 秒	0.5 秒	0.5 秒	1 秒
测试点数目	10	25	10	10	10
测试点分值	10	4	10	10	10
附加样例文件	有	有	有	有	有
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	传统
内存上限	128MB	128MB	128MB	128MB	512MB

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、特别提醒: 评测在 NOI Linux 下进行。

1. 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】

<code>expr.in</code>	<code>expr.out</code>
1+1*3+4	8

【输入输出样例 2】

<code>expr.in</code>	<code>expr.out</code>
1+1234567890*1	7891

【输入输出样例 3】

<code>expr.in</code>	<code>expr.out</code>
1+10000000003*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， $0 \leq \text{表达式中加法运算符和乘法运算符的总数} \leq 100$ ；

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

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

2. 信息传递

(`message.cpp/c/pas`)

【问题描述】

有 n 个同学（编号为 1 到 n ）正在玩一个信息传递的游戏。在游戏里每人都有一个固定的信息传递对象，其中，编号为 i 的同学的信息传递对象是编号为 T_i 的同学。

游戏开始时，每人都只知道自己的生日。之后每一轮中，所有人会同时将自己当前所知的生日信息告诉各自的信息传递对象（注意：可能有人可以从若干人那里获取信息，但是每人只会把信息告诉一个人，即自己的信息传递对象）。当有人从别人口中得知自己的生日时，游戏结束。请问该游戏一共可以进行几轮？

【输入格式】

输入文件名为 `message.in`。

输入共 2 行。

第 1 行包含 1 个正整数 n ，表示 n 个人。

第 2 行包含 n 个用空格隔开的正整数 T_1, T_2, \dots, T_n ，其中第 i 个整数 T_i 表示编号为 i 的同学的信息传递对象是编号为 T_i 的同学， $T_i \leq n$ 且 $T_i \neq i$ 。

数据保证游戏一定会结束。

【输出格式】

输出文件名为 `message.out`。

输出共1行，包含1个整数，表示游戏一共可以进行多少轮。

【输入输出样例 1】

message in	message out
5 2 4 2 3 1	3

见选手目录下的 `message/message1.in` 与 `message/message1.ans`。

【输入输出样例 1 说明】



游戏的流程如图所示。当进行完第3轮游戏后，4号玩家会听到2号玩家告诉他自己的生日，所以答案为3。当然，第3轮游戏后，2号玩家、3号玩家都能从自己的消息来源得知自己的生日，同样符合游戏结束的条件。

【样例输入输出 2】

见选手目录下的 `message/message2.in` 与 `message/message2.ans`。

【数据规模与约定】

对于 30%的数据， $n \leq 200$ ；

对于 60%的数据， $n \leq 2500$ ；

对于 100%的数据， $n \leq 200000$ 。

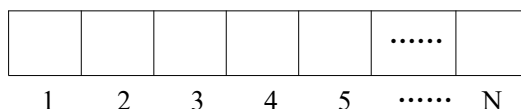
3. 乌龟棋

(`tortoise.pas/c/cpp`)

【问题描述】

小明过生日的时候，爸爸送给他一副乌龟棋当作礼物。

乌龟棋的棋盘是一行 N 个格子，每个格子上一个分数(非负整数)。棋盘第1格是唯一的起点，第 N 格是终点，游戏要求玩家控制一个乌龟棋子从起点出发走到终点。



乌龟棋中 M 张爬行卡片，分成 4 种不同的类型(M 张卡片中不一定包含所有 4 种类型的卡片，见样例)，每种类型的卡片上分别标有 1、2、3、4 四个数字之一，表示使用这种卡片后，乌龟棋子将向前爬行相应的格子数。游戏中，玩家每次需要从所有的爬行卡片中选择一张之前没有使用过的爬行卡片，控制乌龟棋子前进相应的格子数，每张卡片只能使用一次。游戏中，乌龟棋子自动获得起点格子的分数，并且在后续的爬行中每到达一个格子，就得到

该格子相应的分数。玩家最终游戏得分就是乌龟棋子从起点到终点过程中到过的所有格子的分数总和。

很明显，用不同的爬行卡片使用顺序会使得最终游戏的得分不同，小明想要找到一种卡片使用顺序使得最终游戏得分最多。

现在，告诉你棋盘上每个格子的分数和所有的爬行卡片，你能告诉小明，他最多能得到多少分吗？

【输入】

输入文件名 `tortoise.in`。输入文件的每行中两个数之间用一个空格隔开。

第 1 行 2 个正整数 N 和 M ，分别表示棋盘格子数和爬行卡片数。

第 2 行 N 个非负整数， a_1, a_2, \dots, a_N ，其中 a_i 表示棋盘第 i 个格子上的分数。

第 3 行 M 个整数， b_1, b_2, \dots, b_M ，表示 M 张爬行卡片上的数字。

输入数据保证到达终点时刚好用光 M 张爬行卡片，即 $N-1 = \sum_{i=1}^M b_i$ 。

【输出】

输出文件名 `tortoise.out`。

输出只有 1 行，1 个整数，表示小明最多能得到的分数。

【输入输出样例 1】

<code>tortoise.in</code>	<code>tortoise.out</code>
9 5 6 10 14 2 8 8 18 5 17 1 3 1 2 1	73

【输入输出样例 1 说明】

小明使用爬行卡片顺序为 1, 1, 3, 1, 2，得到的分数为 $6+10+14+8+18+17=73$ 。注意，由于起点是 1，所以自动获得第 1 格的分数 6。

【输入输出样例 2】

<code>tortoise.in</code>	<code>tortoise.out</code>
13 8 4 96 10 64 55 13 94 53 5 24 89 8 30 1 1 1 1 1 2 4 1	455

【数据范围】

对于 30% 的数据有 $1 \leq N \leq 30$ ， $1 \leq M \leq 12$ 。

对于 50% 的数据有 $1 \leq N \leq 120$ ， $1 \leq M \leq 50$ ，且 4 种爬行卡片，每种卡片的张数不会超过 20。

对于 100% 的数据有 $1 \leq N \leq 350$ ， $1 \leq M \leq 120$ ，且 4 种爬行卡片，每种卡片的张数不会超过 40； $0 \leq a_i \leq 100$ ， $1 \leq i \leq N$ ； $1 \leq b_i \leq 4$ ， $1 \leq i \leq M$ 。输入数据保证 $N-1 = \sum_{i=1}^M b_i$ 。

4. 选择客栈

(hotel.cpp/c/pas)

【问题描述】

丽江河边有 n 家很有特色的客栈，客栈按照其位置顺序从 1 到 n 编号。每家客栈都按照某一种色调进行装饰(总共 k 种，用整数 $0 \sim k-1$ 表示)，且每家客栈都设有一家咖啡店，每家咖啡店均有各自的最低消费。

两位游客一起去丽江旅游，他们喜欢相同的色调，又想尝试两个不同的客栈，因此决定**分别住在色调相同的两家客栈**中。晚上，他们打算选择一家咖啡店喝咖啡，要求咖啡店位于两人住的两家客栈之间(包括他们住的客栈)，且咖啡店的最低消费不超过 p 。

他们想知道总共有多少种选择住宿的方案，保证晚上可以找到一家最低消费不超过 p 元的咖啡店小聚。

【输入】

输入文件 hotel.in，共 $n+1$ 行。

第一行三个整数 n, k, p ，每两个整数之间用一个空格隔开，分别表示客栈的个数，色调的数目和能接受的最低消费的最高值；

接下来的 n 行，第 $i+1$ 行两个整数，之间用一个空格隔开，分别表示 i 号客栈的装饰色调和 i 号客栈的咖啡店的最低消费。

【输出】

输出文件名为 hotel.out。

输出只有一行，一个整数，表示可选的住宿方案的总数。

【输入输出样例 1】

hotel.in	hotel.out
5 2 3 0 5 1 3 0 2 1 4 1 5	3

【输入输出样例说明】

客栈编号	①	②	③	④	⑤
色调	0	1	0	1	1
最低消费	5	3	2	4	5

2人要住同样色调的客栈，所有可选的住宿方案包括：住客栈①③，②④，②⑤，④⑤，但是若选择住 4、5 号客栈的话，4、5 号客栈之间的咖啡店的最低消费是 4，而两人能承受的最低消费是 3 元，所以不满足要求。因此只有前 3 种方案可选。

【数据范围】

对于 30%的数据，有 $n \leq 100$ ；

对于 50%的数据，有 $n \leq 1,000$ ；

对于 100%的数据，有 $2 \leq n \leq 200,000$ ， $0 < k \leq 50$ ， $0 \leq p \leq 100$ ， $0 \leq \text{最低消费} \leq 100$ 。

5. 排水系统

(water.cpp/c/pas)

【问题描述】

丽江河边有 n 家很有特色的客栈，客栈按照其位置顺序从 1 到 n 编号。每家客栈都按照某一种色调进行装饰(总共 k 种，用整数 $0 \sim k-1$ 表示)，且每家客栈都设有一家咖啡店，每家咖啡店均有各自的最低消费。

两位游客一起去丽江旅游，他们喜欢相同的色调，又想尝试两个不同的客栈，因此决定**分别住在色调相同的两家客栈**中。晚上，他们打算选择一家咖啡店喝咖啡，要求咖啡店位于两人住的两家客栈之间(包括他们住的客栈)，且咖啡店的最低消费不超过 p 。

他们想知道总共有多少种选择住宿的方案，保证晚上可以找到一家最低消费不超过 p 元的咖啡店小聚。

【输入】

输入文件 hotel.in，共 $n+1$ 行。

第一行三个整数 n, k, p ，每两个整数之间用一个空格隔开，分别表示客栈的个数，色调的数目和能接受的最低消费的最高值；

接下来的 n 行，第 $i+1$ 行两个整数，之间用一个空格隔开，分别表示 i 号客栈的装饰色调和 i 号客栈的咖啡店的最低消费。

【输出】

输出文件名为 hotel.out。

输出只有一行，一个整数，表示可选的住宿方案的总数。

【输入输出样例 1】

hotel.in	hotel.out
5 2 3 0 5 1 3 0 2 1 4 1 5	3

【输入输出样例说明】

客栈编号	①	②	③	④	⑤
色调	0	1	0	1	1
最低消费	5	3	2	4	5

2 人要住同样色调的客栈，所有可选的住宿方案包括：住客栈①③，②④，②⑤，④⑤，但是若选择住 4、5 号客栈的话，4、5 号客栈之间的咖啡店的最低消费是 4，而两人能承受的最低消费是 3 元，所以不满足要求。因此只有前 3 种方案可选。

【数据范围】

对于 30% 的数据，有 $n \leq 100$ ；

对于 50% 的数据，有 $n \leq 1,000$ ；

对于 100% 的数据，有 $2 \leq n \leq 200,000$ ， $0 < k \leq 50$ ， $0 \leq p \leq 100$ ， $0 \leq \text{最低消费} \leq 100$ 。