

信息学奥林匹克联赛简单模拟

DST 2018.7

（请选手务必仔细阅读本页内容）

一. 题目概况

中文题目名称	最小公倍式	机器人跳跃	植物大战僵尸	营救
英文题目	least	jump	pvz	rescue
可执行文件名	least	jump	pvz	rescue
输入文件名	least.in	jump.in	pvz.in	rescue.in
输出文件名	least.out	jump.out	pvz.out	rescue.out
每个测试点时限	1 秒	1 秒	2 秒	1 秒
测试点数目	10	10	20	10
每个测试点分值	10	10	5	10
附加样例文件	无	无	无	无
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	least.cpp	jump.cpp	pvz.cpp	rescue.cpp
对于 C 语言	least.c	jump.c	pvz.c	rescue.c
对于 pascal 语言	least.pas	jump.pas	pvz.pas	rescue.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. 题目难度不一定随题号递增。

1. 最小公倍式

(least.cpp/c/pas)

【问题描述】

一个整式能够被另一整式整除, 这个整式就是另一整式的倍式。

——《百度汉语》

公倍式(common multiple)是公倍数概念的推广, 若一个多项式是某几个整式共同的倍式, 则这个整式称为这几个整式的一个公倍式。

最小公倍式(least common multiple)亦称最低公倍式, 最小公倍数概念的推广。指几个整式的公倍式中, 能够整除所有公倍式的那一个公倍式。

——《百度百科》

【输入】

输入文件名为 least.in。

共 $n+1$ 行, 第一行包含一个正整数 n , 表示有 n 个单项式。

接下来 n 行, 每行包含一个字符串, 表示一个单项式。注意: 单项式中的字母的指数取消上标形式, 如 $2x^3y$ 表示为 $2x3y$, 并保证单项式的系数在该单项式的最前面。

【输出】

输出文件名为 least.out。

输出共一行, 表示 n 个单项式的最小公倍式。输出必须保证该公倍式是最简的, 且字母按字典序排序。

【输入输出样例】

least.in	least.out
3	12x12y3z12
6x2y3	
4x3yz4	
12x12yz12	

【数据说明】

对于 10% 的数据, 保证字符串中只含小写字母; $n=2$ 。

对于 30% 的数据, 保证字符串以小写字母开头; $n<5$ 。

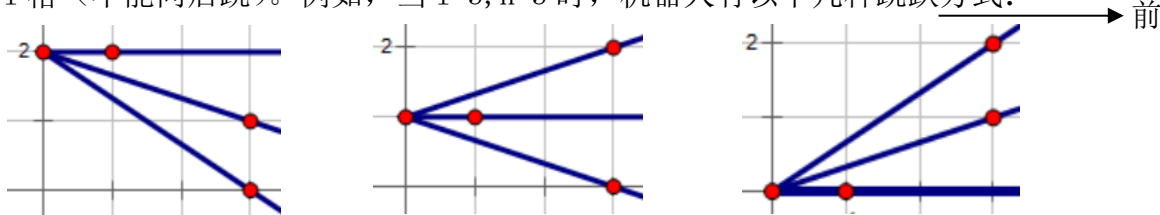
对于 100% 的数据, 保证字符串中只含小写字母、数字; $0<\text{字符串长度} \leq 150$; $1 \leq \text{系数} \leq 80$; $1 \leq \text{指数} \leq 100000$; $2 \leq n < 10$ 。

2. 机器人跳跃

(jump.cpp/c/pas)

【问题描述】

qyy 做了一个机器人。
此时，有一条宽为 n ，长为 m 的路。机器人可以向前跳跃一格或向斜前方跳跃 1 格（不能向后跳）。例如，当 $l=3, n=3$ 时，机器人有以下几种跳跃方式：



路面上有些地方潮湿，有些地方干燥。qyy 希望自己的机器人能够在跳跃过程中跳到的干燥点的个数尽量接近 p 。但由于技术有限（qyy 水平低），机器人只会自动尽量多地跳干燥的点，跳到横坐标大于等于 $m-1$ 的点后称为跳跃过程结束（即整个跳跃过程中跳到干燥点的个数最多）。因此，qyy 只能通过调整起跳的位置来达到目的。那么，请求出满足条件的机器人一开始站的位置的横坐标，若有多个答案，则输出其中横坐标最大的。

【输入】

输入文件名为 jump.in。
共 $n+1$ 行，第一行包含三个正整数 n, m, l, p 。
接下来共 n 行，每行 m 个自然数，中间用一个空格隔开。其中第 i 行的第 j 个数表示点 (i, j) 的潮湿程度 $a_{i,j}$ （0 表示干燥）。

【输出】

输出文件名为 jump.out。
输出共一行，一个自然数，表示机器人一开始站的位置的横坐标。

【输入输出样例 1】

jump.in	jump.out
3 11 2 6	3
1 0 2 1 0 1 1 4 1 2 0	
2 19 1 0 1 0 9 0 0 0 5	
1 3 1 0 0 0 0 7 100 0 1	

【输入输出样例 1 说明】

由于“机器人很智能，会自动尽量多地跳干燥的点”，因此当机器人的起跳横坐标为 3 时，跳跃的路径可以为 (3, 0) (4, 0) (5, 0) (6, 0) (8, 1) (9, 1)，途中共经过 6 个干燥点，与 $p=6$ 最接近且横坐标最大。

【输入输出样例 2】

jump.in	jump.out
3 11 2 4 1 0 1 1 0 1 1 1 1 1 0 1 1 1 0 1 0 1 0 0 0 1 1 1 1 0 0 0 0 1 1 0 1	5

【数据说明】

- 对于 30%的数据， $n \leq 50$ ； $m \leq 50$ ；
- 对于 50%的数据， $n \leq 200$ ； $m \leq 100$ ；
- 对于 70%的数据， $n \leq 1000$ ； $m \leq 100$ ；
- 对于 100%的数据， $n, m \leq 1000$ ； $a_{i,j}, p, l \leq 10000000000$ 。

【提示】

坐标从 0 开始。

3. 植物大战僵尸

(pvz.cpp/c/pas)

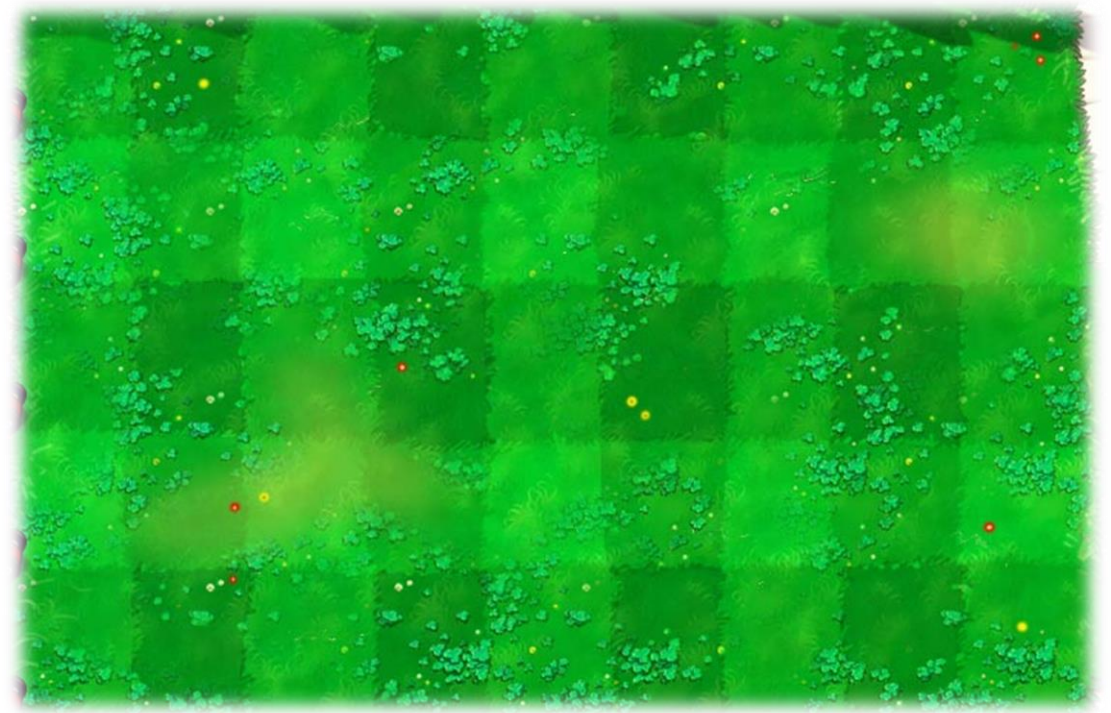
【问题描述】

《植物大战僵尸》(*Plants vs. Zombies*)是由 PopCap Games 开发的一款风靡全球的格斗射击类单机游戏。

——《dst 百科》

然而，y jy 却认为它只是一个益智策略类游戏。这种认知显然是错误的。因此，为了让大家更加全面地了解《植物大战僵尸》（其实是为了让大家帮助他打游戏），dst 决定给大家进行普及。

《植物大战僵尸》的草地地图如图所示：



显而易见的是，草地共有 45 格，即最多能种植 45 株植物。更坑的是，由于 dst 的水平太高，因此他从不使用铲子，即植物种下后不能更改。

下面介绍植物大战僵尸中几种经典的植物：

植物名称	耗费阳光	功能	冷却时间/秒
卷心菜投手	100	每隔 3 秒产生 40 伤害 (在种下后 3 秒第一次产生伤害)	9
樱桃炸弹	150	瞬间产生 1800 伤害 (在种下后 0 秒产生伤害)	27

特殊地，樱桃炸弹在爆炸后，将不再占用原占用的那格（但其依然只能在有格子剩余时被种植）。在游戏开始时，所有植物都处于激活状态（即不处于冷却状态，可直接种植）。

现在，dst 拥有 n 阳光，并且可以且仅可以种植上述植物。那么，他至少需

要多少时间，才能使自己的植物产生至少 p 伤害？

【输入】

输入文件名为 `pvz.in`。

输入共一行，包含两个自然数 n, p ，中间用一个空格隔开。

【输出】

输出文件名为 `pvz.out`。

输出共一行，一个自然数，表示需要的时间。若无法产生至少 p 伤害，输出最大能产生的伤害。

【输入输出样例 1】

<code>pvz.in</code>	<code>pvz.out</code>
250 2000	15

【输入输出样例 1 说明】

在第 0 秒种一株卷心菜投手和一株樱桃炸弹，此时造成樱桃炸弹产生 1800 伤害。15 秒后，卷心菜投手共造成 200 伤害。共计 2000 伤害。

【输入输出样例 2】

<code>pvz.in</code>	<code>pvz.out</code>
32 100	0

【输入输出样例 3】

<code>pvz.in</code>	<code>pvz.out</code>
1754 230768	1092

【数据说明】

对于 10%的数据， $n < 150$ ； $p \leq 500000$ ；

对于 40%的数据， $n \leq 1000$ ； $p \leq 500000$ ；

对于 70%的数据， $n \leq 10000000000$ ； $p \leq 500000$ ；

对于 100%的数据， $n \leq 10000000000$ ； $p \leq 5000000$ 。

4. 营救

(`rescue.cpp/c/pas`)

【问题描述】

dst 被 fjj 绑架了!

fjj 家的俯视图是一个 $n \times m$ 的长方形。我们定义第 i 行第 j 列的坐标为 (i, j) 。dst 目前处于 (sx, sy) ，而 fjj 的家门在 (tx, ty) 。可惜 dst 没有钥匙，因此只能通过到达 dst 的同学 xhr 为他挖的地道 (px, py) 来逃离。同时，由于 fjj 的家被赋予了计算机 buff，任何人只能向上、下、左、右的方向移动。形式化地讲，假设 dst 处于 (x, y) 的位置，则他可以向 $(x+1, y)$ 或 $(x-1, y)$ 或 $(x, y+1)$ 或 $(x, y-1)$ 移动。我们定义每一次移动为走一步路。当然 fjj 早就想到 dst 会逃离，在家中设置了若干障碍。若某点被设置了障碍，则任何人都不能到达该点。

然而，dst 不知道的是，他原本想去的那个地道早已被 fjj 摧毁。不过巧的是，dst 的同学 hc 得到了 fjj 的家门钥匙。在得到消息后，hc 第一时间赶到了 fjj 的家门口。hc 知道 dst 会走最短的路到达地道。因此，hc 需要走最短的路到达 dst 即将行走的路径中的任意一点（包括起点和终点）。那么请问，hc 最少需要走多少步路才能达到目的，成功营救 dst?

最后，dst 为大家补充了最短的路的定义：对于每条路径，dst 都会将他走的每一步的方向连成一个数，其中上代表 1，右代表 2，下代表 3，左代表 4。例如在一条路径中，他的行走方向分别是上(1)→左(4)→下(3)→上(1)→右(2)。那么它连成的数是 14312。在所有路径中，连成的数最小的，即为最短的路。

【输入】

输入文件名为 `rescue.in`。

共 $n+1$ 行，第一行包含八个正整数 $n, m, sx, sy, tx, ty, px, py$ ，中间用一个空格隔开。保证 dst、家门、地道所处位置互不相同，但不保证家门在家的边缘。

接下来共 n 行，每行 m 个整数，中间用一个空格隔开，用来描述 fjj 的家的俯视图。其中第 i 行的第 j 个数表示 (i, j) 是否被设置障碍，有用 1 表示，没有用 0 表示。保证 dst、家门、地道所处位置没有被设置障碍。

【输出】

输出文件名为 `rescue.out`。

输出共一行，表示 hc 最少需要走的步数。若 dst 没有到达地道的可行路径或 hc 没有到达 dst 行走的路径中的任意一点的可行路径，则输出 -1。

【输入输出样例 1】

<code>rescue.in</code>	<code>rescue.out</code>
------------------------	-------------------------

6 6 2 2 5 5 4 4 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0	7
---	---

【输入输出样例 1 说明】

根据题意，dst 即将行走的路径为(1, 2) (1, 3) (1, 4) (2, 4) (3, 4) (4, 4)。此时连成的数为 122333，为最短路径。那么 hc 行走的路径为 (5, 6) (4, 6) (3, 6) (2, 6) (1, 6) (1, 5) (1, 4)，共 7 步，连成的数为 2111114，为最短路径。故答案为 7。

【输入输出样例 2】

rescue.in	rescue.out
5 5 2 2 3 3 4 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0

【输入输出样例 3】

rescue.in	rescue.out
4 4 2 2 3 3 4 3 0 0 0 1	-1

0 0 1 0	
0 1 0 0	
1 0 0 0	

【数据说明】

对于 30%的数据， $n, m \leq 10$ 。
对于 100%的数据， $n, m \leq 1000$ ； $s_x, t_x, p_x \leq n$ ； $s_y, t_y, p_y \leq m$ 。