

重庆育才中学初中信息学测试试题

水波时间： 2021 年 12 月 25 日

友情提示：题目随机排列

一、题目概况

题目名称	负二进制	跳舞	种树	拖拉机	水波
文件名	base	dance	tree	tractor	waves
输入文件名	base.in	dance.in	tree.in	tractor.in	waves.in
输出文件名	base.out	dance.out	tree.out	tractor.out	waves.out
时限	0.1 秒	0.1 秒	0.1 秒	0.1 秒	0.1 秒
内存限制	256M	256M	256M	256M	256M

第一题 负二进制(base)

【问题描述】

负数进制的最大好处是，表示负数不需要用负号。如-2 进使用 -2 作为进制的基数。
-2 进制从最低位开始，每位的权重依次为 1, -2, 4, -8, 16, -32……这种新式的进制可以保证每个整数都有唯一的表示，例如 3 的表示为 $3 = (-2)^2 + (-2)^1 + (-2)^0 = (111)_{-2}$

例如 -9 可以表示成 $-9 = (-2)^3 + (-2)^1 + (-2)^0 = (1011)_{-2}$

来看看更多的例子。1 到 7 的 -2 进制表示依次是

$(1)_{-2}, (110)_{-2}, (111)_{-2}, (100)_{-2}, (101)_{-2}, (11010)_{-2}, (11011)_{-2}$

-1 到 -7 的 -2 进制表示依次是

$(11)_{-2}, (10)_{-2}, (1101)_{-2}, (1100)_{-2}, (1111)_{-2}, (1110)_{-2}, (1001)_{-2}$

请你输出整数 N 的 -2 进制表示。

【输入文件】

单个整数 N, $-2 \cdot 10^9 \leq N \leq 2 \cdot 10^9$

【输出文件】

N 的 -2 进制表示，不要在头部出现多余的 0

样例：

base.in	base.out
-13	110111

第二题 跳舞(dance .cpp)

【问题描述】

哲哲有一个长方形池塘，可以分成 $N \times M$ 个方格，其中一些是坚固得令人惊讶的莲

花,还有一些是岩石,其余的只是纯净湛蓝的水。哲哲正在池塘里练习芭蕾舞,想从起点跳到终点,她的舞步很像象棋中的马:每次跳跃可以横向移动两个 2 格加纵向移 1 格,纵向移动两个 2 格加横向移 1 格,中途必须跳在莲花上,不能呆在水里或岩石上。哲哲发现跳不到终点,因为池塘里缺了些必要的莲花。哲哲计划在有水的格子上加莲花,但石头上不行,他只想添加最少数量的莲花。请帮助哲哲确定至少要添加几朵莲花,顺便帮助他数一下放置这些莲花有多少种不同的方法。

【输入格式】

- 第一行:两个整数 N 和 M , $1 \leq N, M \leq 30$
- 第二行到第 $N+1$ 行:第 $i+1$ 行有 M 个用空格分开的整数,描述了池塘里第 i 行的状态:0 为水,1 为莲花,2 为岩石,3 为起点,4 为终点

【输出格式】

- 第一行:单个整数:表示需要最少增加的莲花数,如果无解则输出 -1
- 第二行:单个整数,表示放置莲花的方案总数,保证这个数字小于 2^{63} ,如果前一行是 -1,这行没有内容。

【输入输出样例】

输入样例(dance.in):	输出样例(dance.out):	样例解释
4 5 1 0 0 0 0 3 0 0 0 0 0 0 2 0 0 0 0 0 4 0	2 3	X 为莲花位置: 1 0 0 0 0 1 0 X 0 0 1 0 X 0 0 3 0 X 0 0 3 0 0 0 0 3 0 0 0 X 0 0 2 0 0 0 X 2 0 0 0 0 2 0 0 0 X 0 4 0 0 0 0 4 0 0 0 0 4 0

第三题 种树(tree.cpp)

【问题描述】

有 N 个城市,编号为 1 到 N 。它们之间有 $N-1$ 条道路,每条道路连接两个城市。通过这些道路,所有城市都是连通的。

刚开始的时候,所有道路都是光秃秃的,没有树。宇宇会在一些道路上批量种树。每次开始种树的时候,宇宇会选择一个城市作为起点,一个城市作为终点,找到从起点到终点的最短路径,在这条路径上所有的道路上分别种下一棵新的树。宇宇在种树时就迫不及待地想知道每条道路上已经有多少树了。

输入数据将会依次出现 M 个事件,以字符 P 开头的事件表示宇宇在一条路径上种植了树,以字符 Q 开头的事件表示在查询一条道路上有多少树。

【输入格式】

- 第一行:两个整数 N 和 M , $2 \leq N \leq 10^5, 1 \leq M \leq 10^5$
- 第二行到第 N 行:第 $i+1$ 行有两个整数 U_i 和 V_i ,表示第 i 条道路连接的两个城市编号, $1 \leq U_i, V_i \leq N$ 。
- 第 $N+1$ 行到第 $N+Q$ 行:每行首先由一个大写字母:
 - 如果是 P,随后会有两个整数 A 和 B ,表示宇宇在从 A 通向 B 的每条道路上都新种

了一棵树， $1 \leq A, B \leq N$ 。

- 如果是 Q，随后会有两个整数 A 和 B，表示查询一条道路上的树数量，A 和 B 是这条道路的两个端点， $1 \leq A, B \leq N$ ，保证输入数据中存在一条 A 到 B 的道路。

【输出格式】

- 对每个查询请求，输出该条道路上树的数量，以换行符分隔。

【输入输出样例】

输入样例(tree.in):	输出样例(tree.out):
4 6	2
1 4	1
2 4	2
3 4	
P 2 3	
P 1 3	
Q 3 4	
P 1 4	
Q 2 4	
Q 1 4	

第四题 拖拉机 (tractor.cpp)

【问题描述】

楠楠有块农田太崎岖了，他要买一辆新拖拉机才能在这里巡视。这块农田由 $N \times N$ 个格子的非负整数表示高度($1 \leq N \leq 500$)。拖拉机从当前格子走到相邻格子（东、南、西、北四个方向）的代价为高度差 D，则楠楠驶过这两个格子的拖拉机最少也要值 D 块钱。

楠楠愿意花足够的钱买一辆新的拖拉机使得他能以最小的高度差走遍所有格子的一半(如果格子总数是奇数，那么一半的值为四舍五入的值)。因为楠楠很懒，所以他找到你帮他编程计算他最小需要花多少钱买到符合这些要求的拖拉机。

【输入格式】

第一行为一个整数 N

第 2 到 N+1 行每行包含 N 个非负整数（不超过 1,000,000），表示当前格子的高度。

【输出格式】

共一行，表示楠楠买拖拉机要花的最小价钱。

【输入输出样例】

输入样例(tractor.in):	输出样例(tractor.out):
5	3
0 0 0 3 3	
0 0 0 0 3	
0 9 9 3 3	
9 9 9 3 3	
9 9 9 9 3	

第五题 水波 (waves.cpp)

【问题描述】

可可喜欢在水塘里打水漂玩。当石子击打水面后，水波就扩散开来了。

湖面用一个矩阵来表示。水面存在一种叫深度的概念，用来描述波的传播。当石子打击水面之前，每个格子（除了那些在水坝上的格子）的深度都为 0。当波传播开来后，每个格子描述如下：o 格子 $\text{depth} < 0$ ，-格子 $\text{depth} = 0$ ，*格子 $\text{depth} > 0$ ，X 格子是河堤的一部分。

当一个石子打击水面后，就会产生一个沿菱形方向每秒向外传播一格的上凸水波，并且使水面的深度提高 1。两秒之后，又产生一个下凹的水波，并且使水面的深度降低 1，同样沿菱形方向每秒向外传播一格。一次击打仅产生两个水波，即一个上凸水波和紧跟的一个下凹水波。

例如，下图描述了一个石子打到湖中心后，0，1，2，3 秒时的情况：

0 seconds	1 second	2 seconds	3 seconds
-----	-----	-----	---*---
-----	-----	---*---	--*-*-
-----	--*---	--*-*-	-*-o*-
---*---	--*-*-	-*-o*-	*-o-o*
-----	--*---	--*-*-	-*-o*-
-----	-----	---*---	--*-*-
-----	-----	-----	---*---

河堤用 x 表示，并且和湖的宽（从顶到底）一样长。当一列波碰到河堤，河堤将反弹波。

下面是一列波碰到河堤的情形。为了简单明了，图中只画了上凸波，而舍去了下凹波：

1 second	2 seconds	3 seconds	4 seconds	5 seconds
X-----	X-----	X--*---	X--*-*	X*---*-
X-----	X--*---	X--*-*	X*---*-	X*---*-
X--*---	X--*-*	X*---*-	X*---*-	X--*---
X--*-*	X*---*-	X*---*-	X--*---	X--*---
X--*---	X--*-*	X*---*-	X*---*-	X--*---
X-----	X--*---	X--*-*	X*---*-	X*---*-
X-----	X-----	X--*---	X--*-*	X*---*-

当一些波相遇后，他们的传播方向不会改变，也就是说，相遇后 1 秒，各波仍按原传播方向移动。记录将表现出他们的复合情形。例如：

1 second	2 seconds	3 seconds	4 seconds
-----	-----	-----	---*---
-----	-----	---*---	--*-*-
-----	--*---	--*-*-	-*-o*-
---*---	--*-*-	-*-o*-	*-o-----
--*-*-	-*-o*-	*-o-----	-o-*o---
---*---	--*-*-	-*-o*-	*-o-----
-----	--*---	--*-*-	-*-o*-
-----	-----	---*---	--*-*-
-----	-----	-----	---*---

写一个程序，告诉可可水波随着时间将如何传播。

【输入格式】

第 1 行：四个用空格隔开的整数：P,B1,B2,R:

P($1 \leq P \leq 5$)表示石子的个数，

B1($-500,000 \leq B1 \leq 500,000$)和 B2($-500,000 \leq B2 \leq 500,000$)表示两个河堤的 x 坐标，

R($1 \leq R \leq 500,000$)表示你要描述湖面多少秒。

没有两个石子会在同一时间砸到同一地点，两个河堤一定有不同的坐标，没有石子会砸到河堤上去。

第 2..P+1 行：每行有三个用空格隔开的整数描述了一颗石子：X($-500,000 \leq X \leq 500,000$), Y($-500,000 \leq Y \leq 500,000$), T($1 \leq T \leq 500,000$)

X,Y 表示石子砸的地点的坐标

T 表示石子是什么时候砸下去的。

【输出格式】

第 1..9 行：输出是一个 9*9 的矩阵，中心在(0,0)点。左下点的坐标为(-4,-4)右上点的坐标为(4,4)。这个矩阵表现的是 R 秒时湖面状态。

【输入输出样例】

输入样例(waves.in):	输出样例(waves.out):
2 4 100 4	-----X
-3 0 1	-*-----X
0 0 2	*-*-*---X
	-o-*-*---X
	o-----*X
	-o-*-*---X
	--*---X
	-*-----X
	-----X

【数据范围】

对于 20%的数据，保证有 $p=1$, $-200 \leq B1, B2 \leq 200$, $1 \leq R \leq 20$ 。

对于 30%的数据，保证有 $2 \leq p \leq 5$, $-10 \leq B1, B2 \leq 10$, $1 \leq R \leq 10$ 。

对于 60%的数据，保证有 $1 \leq p \leq 5$, $-200 \leq B1, B2 \leq 200$, $1 \leq R \leq 200$ 。

对于 100%的数据，保证有 $1 \leq p \leq 5$, $-500,000 \leq B1, B2 \leq 500,000$, $1 \leq R \leq 500,000$ 。