2022 信息学国庆集训 B 班

模拟赛

Day1

时间：2022 年 10 月 2 日 08:00 ~ 12:30

题目名称	签到题	机器人	玩具装箱	双色棋
题目类型	传统题	传统题	传统题	传统题
目录	`sign`	`robot`	`toy`	`chess`
可执行文件名	`sign`	`robot`	`toy`	`chess`
输入文件名	`sign.in`	`robot.in`	`toy.in`	`chess.in`
输出文件名	`sign.out`	`robot.out`	`toy.out`	`chess.out`
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	128 MB	128 MB	512 MB	128 MB
测试点数目	10	10	10	10
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	`sign.cpp`	`robot.cpp`	`toy.cpp`	`chess.cpp`

编译选项

对于 C++ 语言	`-std=c++11`

【注意事项（请仔细阅读）】

1. 选手提交的源程序必须存放在**已建立**好的，且**带有样例文件和下发文件**的文件夹。中，文件夹名称与对应试题英文名一致。
2. 文件名（程序名和输入输出文件名）必须使用英文小写。
3. 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。
4. **对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理**。
5. 若无特殊说明，结果比较方式为**忽略行末空格、文末回车后的全文比较**。
6. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 在终端中执行命令 `ulimit -s unlimited` 可将当前终端下的栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
8. 每道题目所提交的**代码文件大小限制为 100KB**。
9. 若无特殊说明，输入文件与输出文件中同一行的相邻整数均使用一个空格分隔。

10. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 `scanf` 函数）避免出错。
11. 直接复制 PDF 题面中的多行样例，数据将带有行号，建议选手直接使用对应目录下的样例文件进行测试。
12. 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。
13. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外**不允许在程序中手动开启其他编译选项**，一经发现，本题成绩以 0 分处理。

签到题（sign）

【题目描述】

给定 N ，求小于 N 的、最大的非负整数 K ，满足：

$N\&(N-1)\&(N-2)\&...\&(K+2)\&(K+1)\&K=0$

其中 & 表示按位与（C++ 中的 & ）。因此上述条件等价于：将区间 $[K,N]$ 里的所有正整数按位与起来等于 0 。

【输入格式】

从文件 ***sign.in*** 中读入数据。

每个测试点有多组测试数据。第一行包含一个整数 T 表示数据组数。对于每组测试数据：

第一行包含一个正整数 N。

【输出格式】

输出到文件 ***sign.out*** 中。

对于每组测试数据，输出一行一个整数 K。

【样例输入 1】

```
3
2
5
17
```

【样例输出 1】

```
1
3
15
```

【测试点约束】

对于所有测试数据，$1\le T\le 10^5, 1\le N\le 10^9$。

对于 20% 的数据，$T = 1, 1 \leq N \leq 10^5$

对于 50% 的数据，$1 \leq N \leq 10^5$

机器人（robot）

【题目描述】

在一个无穷大的二维平面上，你和你的机器人走散了。

通过定位，你知道你自己的坐标和机器人现在所在位置的坐标。并且你知道，机器人存储着一个长度为 S 的字符串 s，只含字符 U 、D 、L 、R，表示向上、下、左、右移动一个单位（即，上表示将自己的 y 坐标 $+1$，左表示将自己的 x 坐标 -1，另外两个相反）。

从你发现机器人走丢的这个时刻（时刻 0 ）开始，机器人每个时刻会读取字符串的下一个字符并进行对应移动，经过 S 个时刻后，机器人会从第一个字符开始再读一次，一直循环；你每个时刻可以向上、下、左、右任意一个方向移动一个单位，**也可以停在原地**。在某个时刻，如果你的坐标和机器人的坐标相同了，你就找到了自己的机器人。

你希望尽早找回自己的机器人，请问这个时刻最早是多少？如果你永远无法找回自己的机器人，输出 -1 并准备好迎接一场即将到来的智械危机。

【输入格式】

从文件 ***robot.in*** 中读入数据。

第一行包含四个整数 x_r, y_r, x_p, y_p 表示机器人的初始坐标 (x_r, y_r) 和你的初始坐标 (x_p, y_p) 。

第二行包含一个正整数 S 表示机器人指令序列的长度。

第三行包含一个长度为 S 的字符串 s 表示指令序列，保证这个字符串只含字符 U 、D 、L 、R 。

【输出格式】

输出到文件 ***robot.out*** 中。

输出一个整数表示答案。

【样例输入 1】

```
0 0 8 8
2
RU
```

【样例输出 1】

```
8
```

【样例解释 1】

机器人初始在 $(0,0)$，你初始在 $(8,8)$。机器人会一直向右、上、右、上……走，因此你只要一直向左、下、左、下……走，8 个时刻后你们刚好坐标同为 $(4,4)$。显然这是可能的最短时间。

【样例输入 2】

```
0 0 0 1
1
L
```

【样例输出 2】

```
-1
```

【样例解释 2】

机器人初始在你的正左方，且它会一直向左走，由于你们的速度是一样的，你永远追不上它。

【测试点约束】

对于所有测试数据，$0 \le x_r, y_r, x_p, y_p \le 10^9, 1 \le S \le 10^5$
对于 20% 的数据，$S = 1$
对于另外 20% 的数据，$x_r, y_r, x_p, y_p, S \le 10$
对于另外 20% 的数据，$x_r = x_p$

玩具装箱（toy）

【题目描述】

~~本题并不是那道经典的斜率优化 DP 题，只是故意起了一样的名字。~~

找回你的机器人后，你们来到了玩具国旅行。

玩具国出售的物品分为玩具和箱子两种。每个玩具具有自己的体积 v_i ，每个箱子具有自己的容积 c_j 。第 i 个玩具可以装进第 j 个箱子，当且仅当 $v_i \le c_j$ 。同时，一个箱子只能装最多一件玩具，显然，一件玩具也最多只能被一个箱子装。我们将一个用箱子装起来的玩具称为一份礼物。

按时间顺序，你一共购买了 N 件物品，它们有些是玩具，有些是箱子。每次买下一件物品（箱子或玩具）后，你都想知道：对于目前你拥有的所有玩具和箱子，最多能组成多少份礼物。

【输入格式】

从文件 ***toy.in*** 中读入数据。

第一行包含一个正整数 N 表示你购买的物品数。

接下来 N 行，每行包含两个整数 a_i, b_i ，保证 $a_i \in \{0, 1\}$ 。

$a_i = 0$ ，表示第 i 个物品是一个箱子，它的容积 $c_i = b_i$ 。

$a_i = 1$ ，表示第 i 个物品是一个玩具，它的体积 $v_i = b_i$ 。

【输出格式】

输出到文件 ***toy.out*** 中。

输出 N 行，第 i 行表示购买第 i 个物品后，最多能组成多少份礼物。

【样例输入 1】

```
6
1 1
1 5
0 6
0 4
1 3
0 2
```

【样例输出 1】

```
0
0
1
2
2
3
```

【样例解释 1】

我们用 (v_i, c_j) 表示一份礼物。
购买第 1 、2 个物品后，不存在箱子，不可能组成礼物。
购买第 3 个物品后，最优方案为 $(1,6)$ ；
购买第 4 个物品后，最优方案为 $(1,4)$ 、$(5,6)$ ；
购买第 5 个物品后，最优方案为 $(1,4)$ 、$(3,6)$ ；
购买第 6 个物品后，最优方案为 $(1,2)$ 、$(3,4)$ 、$(5,6)$ 。

【样例 2】

见选手目录下的 ***toy/toy2.in*** 与 ***toy/toy2.ans***。
这组样例和测试点 5 的生成方式相同。

【测试点约束】

对于所有测试数据，$1 \le N \le 10^5, 1 \le b_i \le 10^9$

测试点编号	$N \le$	特殊性质 A	特殊性质 B
$1,2$	10	有	
$3,4$	1000		
$5,6$	10^5		有
7			
$8,9,10$			

特殊性质 A ：$1 \le b_i \le N$ 且 b_i 两两不同。

特殊性质 B ：存在一个 t 满足 $1 \le t \le i$ 时 $a_i = 1$ ，$t < i \le N$ 时 $a_i = 0$。即，一旦你购买了一个箱子，你就不再购买玩具了。

双色棋（chess）

【题目描述】

咸鱼鱼和蓝猫猫在下双色棋。

双色棋盘是一个 $N(N \geq 2)$ 个格子组成的**环**，从 1 到 N 依次编号。刚开始每个格子都是空的。咸鱼鱼先手，蓝猫猫后手，两个人轮流操作。

轮到某个玩家操作的时候，他可以选择将一个黑色或白色的棋子放在某个空的格子。放置棋子时，必须满足左右相邻的两个格子里没有放和自己正在放置的棋子颜色相同的棋子。**一个玩家在不同的回合可以选择不同颜色的棋子**。

当某个玩家无法操作的时候，对手获胜。

咸鱼鱼和蓝猫猫都极度聪明，他们的每次操作都是**理智的**。也就是说，如果现在的局势下，他有胜利的策略，那么他下过棋后，他依旧有胜利的策略（在此基础上如果有很多可行的走法，他会**随便走一步**）；如果现在的局势下，他已然没有胜利的策略，他就会在规则允许的范围内**随便走一步**。

事实上，这是一个有向图游戏，对于足够聪明的咸鱼鱼和蓝猫猫来说，谁赢谁输一开始就确定了。所以咸鱼鱼希望你告诉他，**有多少种可能的对局**。

两个对局中，只要有至少一个人的至少一步走的不同，包括选择的棋子颜色不同或放置位置不同，两个对局就算不同。

答案可能非常大，蓝猫猫希望你输出的**答案对** 10^9+7 **取模**。

比如，$N=2$ 的时候，显然蓝猫猫只要用另一种颜色填充咸鱼鱼没下的格子就能获胜，咸鱼鱼从一开始就没有胜算。因此咸鱼鱼第一步直接乱下，方案数是 4 （2 种可选颜色乘以 2 个可选位置），蓝猫猫只有 1 种选择，最终可能的对局有 4 种。

比如，$N=3$ 的时候，显然蓝猫猫只要用另一种颜色填充咸鱼鱼其中一个没下的格子就能获胜。因此咸鱼鱼第一步直接乱下，方案数是 6 （2 种颜色，3 个位置），蓝猫猫有 2 种选择（2 个位置），最终可能的对局有 12 种。

比如，$N=42069$ 的时候，显然咸鱼鱼第一步有 84138 种走法，蓝猫猫有……（关于此，我确信已发现了一种美妙的证法，可惜这里空白的地方太小，写不下），最终答案取模后是 675837193 。

【输入格式】

从文件 ***chess.in*** 中读入数据。

第一行包含一个正整数 N 。

【输出格式】

输出到文件 ***chess.out*** 中。

输出一个非负整数表示答案。

【样例输入 1】

```
3
```

【样例输出 1】

```
12
```

【样例输入 2】

```
42069
```

【样例输出 2】

```
675837193
```

【测试点约束】

对于所有测试数据，$2 \leq N \leq 10^6$
对于 40% 的数据，$2 \leq N \leq 8$
对于 70% 的数据，$2 \leq N \leq 1000$