

Public NOIP Round #1

普及组

时间：2022 年 9 月 10 日 14:00 ~ 17:30

题目名称	一维围棋	斜二等轴测图	盒子里的糖果	冲塔
题目类型	传统型	传统型	传统型	传统型
可执行文件名	capture	draw	candy	tower
每个测试点时限	1.0 秒	1.0 秒	2.0 秒	3.0 秒
内存限制	512 MiB	512 MiB	512 MiB	1 GiB
子任务数目	20	10	10	7
测试点是否等分	是	是	是	否

提交源程序文件名

对于 C++ 语言	capture.cpp	draw.cpp	candy.cpp	tower.cpp
-----------	-------------	----------	-----------	-----------

编译选项

对于 C++ 语言	-lm -O2
-----------	---------

注意事项：

1. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
2. 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。
3. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格进行分隔。
4. 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。
5. 程序可使用的栈空间大小与该题内存空间限制一致。
6. 在终端下可使用命令 `ulimit -s unlimited` 将栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
7. 评测时采用的机器配置为 Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz (QOJ 与 Public Judge)。使用的操作系统为 Ubuntu 18.04，编译器版本为 g++ 11.1.0 (Ubuntu 11.1.0-1ubuntu1 18.04.1)，上述时限以此为准。
8. **特别注意，在提交时不需要使用文件输入输出。选手应在标准输入中读入数据，并将答案输出至标准输出。**

一维围棋 (capture)

【题目描述】

小 w 和小 b 在下围棋，与传统围棋不同，他们是在一个由 n 个格子排成一行构成的棋盘上博弈，每个格子至多放一个棋子。

我们设从左到右第 i 个格子上的棋子为 S_i , $S_i \in \{W, B, \cdot\}$, 分别表示这是一个白色的棋子, 或黑色的棋子, 或没有棋子。

假如存在区间 $[l, r]$, 使得 $r - l + 1 \geq 3$, 并且这个区间内的格子全部都放了棋子。此时如果 $l + 1, \dots, r - 1$ 这些格子上的棋子同色, 但 l, r 这两个格子上的棋子与它们异色, 则称 $l + 1, \dots, r - 1$ 这些格子上的棋子被占领了。比如, 如果有 WWBBW.WB, 则第三、四个格子上的棋子被占领了。

现在给定 S_i , 保证现在没有任何棋子被占领。并且下一步是小 w 下, 也就是说, 小 w 要选一个位置放一个白棋子, 要求小 w 放棋子之后没有白棋子被占领 (保证一定存在方案可以做到这一点)。

小 w 希望他放棋子后占领黑棋子数量最多, 但是他棋艺不精, 所以找到了你来帮他算出这个数字。

【输入格式】

从标准输入中读入数据。

一行一个正整数 n 以及一个长度为 n 的只包含 B, W, . 的字符串, 表示 $\{S_i\}$ 。

【输出格式】

输出到标准输出中。

一行一个整数表示占领黑棋子的最大数量。

【样例 1 输入】

```
1 5 .WB..
```

【样例 1 输出】

```
1 1
```

【样例 1 解释】

我们可以把棋子放在第三个格子，这样第二个格子的黑棋子就会被占领。

【样例 2 输入】

```
1 5 WB.B.
```

【样例 2 输出】

```
1 0
```

【样例 2 解释】

假如把棋子放在第三个格子，那么这个棋子就会被占领，不符合题目要求。所以我们只能把棋子放在第五个格子并不占领任何棋子。

【样例 3】

见选手目录下 *capture/capture3.in* 与 *capture/capture3.ans*。

【样例 4】

见选手目录下 *capture/capture4.in* 与 *capture/capture4.ans*。

【子任务】

对于 30% 的数据，保证恰好存在一个 i 使得 $S_i = B$ 。

对于另 5% 的数据，保证 $\forall i, S_i \neq W$ 。

对于 100% 的数据，保证 $1 \leq n \leq 100$ 。

斜二等轴测图 (draw)

【题目描述】

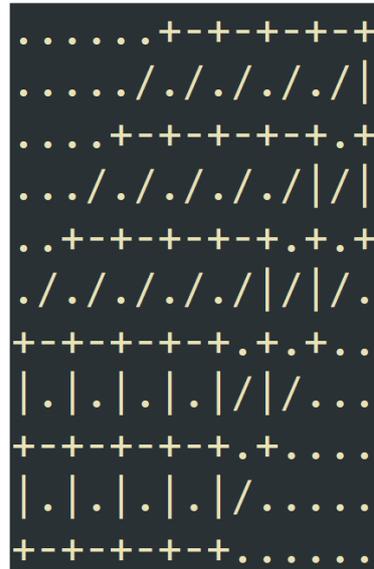
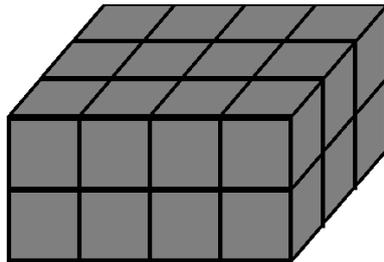
今天是 YQH 的生日，她得到了一个长方体作为生日礼物，这个长方体的长宽高分别为 a, b, c 。这个长方体是由 $a \times b \times c$ 个大小为 $1 \times 1 \times 1$ 的正方体拼接而成。

由于 YQH 十分喜欢立体几何，于是她打算把长方体的斜二等轴测图画出来。但是她发现一个问题：长方体太大了，她找不到一张足够大的纸来画图，于是她退而求次，准备编程用字符画来画图。

可惜，YQH 的编程水平不高，所以她找到你来帮助她。

如果你不知道斜二等轴测图是什么或不知道如何用字符画表示斜二等轴测图，可以参考下面的图片以及样例来理解。

当 $a = 4, b = 3, c = 2$ ，左边是斜二等轴测图，右边是字符画：



【输入格式】

从标准输入中读入数据。

第一行一个正整数 T ，表示数据组数。

对于每组数据一行三个正整数分别表示题面中的 a, b, c 。

【输出格式】

输出到标准输出中。

对于每组数据，输出 $1 + 2(b + c)$ 行，每行一个长度为 $1 + 2(a + b)$ 的字符串表示答案。

【样例 1 输入】

```

1 2
2 1 1 1
3 6 2 4

```

【样例 1 输出】

```

1  ..+--+
2  ././|
3  +-+.+
4  |.|/.
5  +-+..
6  .....+--+--+--+--+
7  .../././././././|
8  ..+--+--+--+--++.+
9  ./././././././|/|
10 +-+--+--+--+--++.+.+
11 |.|.|.|.|.|.|/|/|
12 +-+--+--+--+--++.+.+
13 |.|.|.|.|.|.|/|/|
14 +-+--+--+--+--++.+.+
15 |.|.|.|.|.|.|/|/|.
16 +-+--+--+--+--++.+..
17 |.|.|.|.|.|.|/...
18 +-+--+--+--+--++. ....

```

【样例 2】

见选手目录下 *draw/draw2.in* 与 *draw/draw2.ans*。

【样例 3】

见选手目录下 *draw/draw3.in* 与 *draw/draw3.ans*。

【子任务】

对于 30% 的数据, $b = c = 1$ 。

对于 60% 的数据, $c = 1$ 。

对于 100% 的数据, $1 \leq T \leq 50$, $1 \leq a, b, c \leq 20$ 。

盒子里的糖果 (candy)

【题目描述】

在桌子上摆着 n 个盒子，左数第 i 个盒子里初始装着 a_i 颗糖果。

接下来小 P 进行了 m 次操作，每次操作有参数 l, r ，小 P 可以选择一个操作执行：

1. 在左数第 l 个盒子到第 r 个盒子各放入一颗糖果。
2. 将左数第 l 个盒子到第 r 个盒子任意重排列。

小 P 想知道，所有操作结束之后，每个位置的盒子当中，最多包含了多少糖果。

【输入格式】

从标准输入中读入数据。

第一行两个整数 n, m 表示盒子数和操作数。

第二行包含 n 个整数，第 i 个整数 a_i 表示第 i 个盒子初始糖果数。

接下来 m 行每行两个整数 l, r 描述一次操作。

【输出格式】

输出到标准输出中。

输出一行 n 个整数，第 i 个整数表示左数第 i 个盒子中最多有多少糖果。

【样例 1 输入】

```
1 4 5
2 1 2 2 3
3 2 2
4 2 2
5 1 3
6 1 3
7 3 4
```

【样例 1 输出】

```
1 5 6 6 5
```

【样例 1 解释】

解释一种让第 4 个盒子糖果数为 5 的方法：

$(1, 2, 2, 3) \rightarrow (1, 3, 2, 3) \rightarrow (1, 4, 2, 3) \rightarrow (1, 2, 4, 3) \rightarrow (2, 3, 5, 3) \rightarrow (2, 3, 3, 5)$

【样例 2】

见选手目录下 *candy/candy2.in* 与 *candy/candy2.ans*。

【子任务】

对于 100% 的数据，保证 $1 \leq n, m \leq 5 \times 10^5, 1 \leq a_i \leq 10^9, 1 \leq l \leq r \leq n$ 。

测试点编号	$n \leq$	$m \leq$	$a_i \leq$
1	5	3	10
2			10^6
3			10^4
4	5 000	5 000	10^6
5			10^9
6			100
7	5×10^5	5×10^5	10^6
8			10^6
9			10^9
10			10^9

冲塔 (tower)

【题目描述】

你是一个冲塔大队的首领，要率领部下进行坚定全面地冲塔。

平面上有 n 个塔，第 i 个塔的位置是 (x_i, y_i) ，且位置两两不同。你可以选择任意若干个塔进行冲塔，但是需要满足两个条件：

- 为了保证部下的安全返回，同一个横坐标只能冲至多两个塔，同一个纵坐标也只能冲至多两个塔。
- 为了保证冲塔的全面进行，一个没有被冲的塔必须被夹在两个横坐标相同的塔之间，或是夹在两个纵坐标相同的塔之间。

构造一个合法的冲塔方案，或判断无解。

【输入格式】

从标准输入中读入数据。

第一行一个正整数 n 。

接下来 n 行，第 i 行两个整数 x_i, y_i 。

【输出格式】

输出到标准输出中。

如果无解，输出 -1 。

否则输出一行一个长度为 n 的 01 字符串，第 i 个字符为 1 当且仅当你选择冲第 i 个塔。

【样例 1 输入】

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

【样例 1 输出】

```
1 110
```

【样例 2 输入】

```

1 6
2 1 1
3 1 2
4 2 1
5 2 2
6 3 1
7 3 2

```

【样例 2 输出】

```

1 110011

```

【样例 3】

见选手目录下 *tower/tower3.in* 与 *tower/tower3.ans*。

【样例 4】

见选手目录下 *tower/tower4.in* 与 *tower/tower4.ans*。

【子任务】

子任务编号	$n \leq$	特殊性质	分值
1	3	无	5
2	16		11
3	10^6	存在 a, b 使得 $n = ab$, 且 $1 \leq x_i \leq b, 1 \leq y_i \leq a$ 。	7
4		每个横坐标上至多有两座塔。	6
5	5000	无	31
6	10^5		21
7	10^6		19