Day 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 题目名称 | 消消乐 | 猜猜乐 | 跑步 | 树 |
| 可执行文件名 | eliminate | guess | run | tree |
| 输入文件名 | eliminate.in | guess.in | run.in | tree.in |
| 输出文件名 | eliminate.out | guess.out | run.out | tree.out |
| 每个测试点时限 | 1秒 | 1秒 | 1秒 | 1秒 |
| 内存限制 | 256MB | 256MB | 256MB | 256MB |
| 测试点数目 | 20 | 20 | 20 | 25 |
| 每个测试点分值 | 5 | 5 | 5 | 4 |
| 是否有部分分 | 否 | 否 | 否 | 否 |
| 题目类型 | 传统型 | 传统型 | 传统型 | 传统型 |
| 是否有样例文件 | 否 | 否 | 否 | 否 |
| 是否有附加文件 | 否 | 否 | 否 | 否 |

提交源程序须加后缀

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 对于C++ 语言 | eliminate.cpp | guess.cpp | run.cpp | tree.cpp |

编译开关

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

nodesnodesnodes1.in~nodes10.innodes1.out~nodes10.outnodes1.ans~nodes10.ans

消消乐(Eliminate)

【问题描述】

你在玩一种消消乐。

在你面前有两个白盘子，染料有红绿蓝三种。在游戏中，你会按顺序获得个染料。

每一次你获得染料，你可以选择染或者不染，若选择染，则遵守以下规则，

* 若第一个盘子是白色的，则染第一个盘子
* 否则，若第二个盘子是白色的，则染第二个盘子
* 否则，若两个盘子里的颜色和你手上的染料相同，则你用手上的染料染一个奖励的白盘子，并且将三个相同颜色的盘子卖出，**你获得一分**。在此之后，你重新获得两个白盘子，并且将第一个盘子染成你指定的颜色。
* 否则，若两个盘子里的颜色和你手上的染料两两不同，则你用手上的染料染一个奖励的白盘子，并且将三个相同颜色的盘子销毁。在此之后，你重新获得两个白盘子，并且将两个盘子分别染成你指定的颜色（可以不同）。
* 否则，将第一个盘子里丢弃，将第二个盘子放到第一个盘子的位置，你用手上的染料染一个奖励的白盘子放到第二个盘子的位置

你已经知道了将要获得的染料序列，你想要知道，你最多能获得多少分

【输入格式】

输入文件为***eliminate.in***。

输入文件的第一行包含一个字符串， 由字母’R’, ‘G’, ‘B’ 组成，表示染料序列。

【输出格式与部分分】

输出文件为***eliminate .out***。

输出仅一行一个整数，表示最多能获得的分数。

【样例1输入】

RRRRR

【样例1输出】

2

【样例2输入】

RBGG

【样例2输出】

1

【样例3输入】

RBRBR

【样例3输出】

1

【子任务】

所有测试数据的范围和特点如下:

对于的数据有

对于的数据有 。

对于的数据有

猜猜乐(Guess)

【问题描述】

现在有一个数据结构，但是你不知道这到底是什么数据结构，我们只知道这是一个数据结构。为了让你知道这是什么数据结构，你制造了很多次的加入和取出操作。每次加入操作，你向这个数据结构加入了一个数；每次取出操作，你从数据结构之中取出一个数。

现在你想知道，这个数据结构可不可能是堆、栈、队列。

【输入格式】

输入文件为***guess.in***。

第一行一个整数𝑁代表操作的数目。

接下来𝑁行，每行两个整数𝑜𝑝𝑡,𝑣。如果𝑜𝑝𝑡=1，代表把𝑣加入了数据结构；如果𝑜𝑝𝑡=2，代表了从数据结构中取出了一个数，值是𝑣。

【输出格式与部分分】

输出文件为 ***guess .out***。

输出总共三行，第一行代表数据结构是否可能是栈，第二行代表数据结构是否可能是队列，第三行代表数据结构是否可能大根堆。每一行的结果都只可能是YES或者 NO。

【样例1输入】

2

1 1

2 1

【样例1输出】

YES

YES

YES

【子任务】

所有测试数据的范围和特点如下:

对于的数据有

跑步(Run)

【问题背景】

老虎是一名旅游爱好者。

时至8 月，正是前往海边度假的好时节。在海边享受阳光，沙滩和海风呼啸，真是虎生的一大享受。海边的居民有着自己的风俗习惯，具体来说，海边可以看做是一个 的网格，左上角为，右下角为 。在每个网格中，都生活着渔民一家。对渔民来说，售卖新鲜的贝类显然是谋生的一大工具，而在小镇中，只有网格中有海鲜市场。每天，渔民们会空手从自己的家离开，并通过向左或向上走的方式到达海鲜市场。渔民会在沿途中拾取贝类，并在海鲜市场出售。具体来说，如果一个渔民经过了点，那么他能够拾取到价值为 的贝壳。

显然，每个渔民会选择一条收益最大的路径前进，现在老虎想要知道，所有渔民的最大收益和是多少。

由于自然环境的变化，渔民经过某个位置的代价可能会改变。但自然环境是动态平衡而稳定的，因此渔民经过某个位置时的收益变化不会超过1。同时，渔民显然不会做不利于自己的事，因此我们保证条件 始终满足。

【输入格式】

输入文件为run.in。

第一行包含一个整数，表示网格的行数，同时也表示修改的次数。

接下来行每行个数，第行第列表示。

接下来行每行的一开始是一个字符，紧跟的两个数 表示修改的位置。当输入的字符是U 时代表代价增加1，否则代表代价减少1

【输出格式】

输出文件为run***.out***。

输出行，表示每次修改后的答案。

【样例1输入】

3

3 2 7

4 2 6

5 3 8

U 1 2

D 3 2

U 1 2

【样例1输出】

107

111

110

114

【子任务】

所有测试数据的范围和特点如下:

对于的数据有

对于额外的数据有 数据随机

对于的数据有

树(Tree)

【问题背景】

现在有有一颗有个节点的树，每个节点有一个颜色a或b或c。请问树上有多少条简单路径，路径上三种颜色的节点数量相同。

【输入格式】

输入文件为tree.in。

输入文件的第一行包含个正整数 ，表示节点数量。

第二行包含长度为的由a,b,c组成的字符串，表示各个节点的颜色。

接下来 行，每行个整数，表示树上的一条边。

【输出格式】

输出文件为tree***.out***。

输出文件共一个整数，表示树上三种颜色节点数量相同的简单路径数量。

【样例1输入】

6

abbccb

1 2

1 3

1 4

1 5

4 6

【样例1输出】

5

【子任务】

所有测试数据的范围和特点如下:

对于的数据有

对于的数据有

对于的数据有

对于的数据有