

XXXXXXXXX 赛

命题人: ModCx & LeNotFonud

题面与评测负责人: *ModCx*

时间: 2022 年 11 月 16 日 15:40 ~ 18:00

题目名称	空岛生存	大整数加法	作弊者的归宿
题目类型	传统型	传统型	传统型
目录	survival	addition	pardon
可执行文件名	survival	addition	pardon
输入文件名	survival.in	addition.in	pardon.in
输出文件名	survival.out	addition.out	pardon.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒
内存限制	256 MB	256 MB	256 MB
测试点数目	20	20	10
测试点是否等分	是	是	是

题目名称	陷阱	智慧殿堂	窗外的流星
题目类型	传统型	传统型	传统型
目录	trap	palace	meteor
可执行文件名	trap	palace	meteor
输入文件名	trap.in	palace.in	meteor.in
输出文件名	trap.out	palace.out	meteor.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MB	512 MB	128 MB
测试点数目	60	30	30
测试点是否等分	否	是	是

提交源程序文件名

对于 C++ 语言	survival.cpp	addition.cpp	pardon.cpp
对于 C++ 语言	trap.cpp	palace.cpp	meteor.cpp

编译选项

对于 C++ 语言	-lm -O2 -std=c++17
-----------	--------------------

注意事项与提醒 (请选手务必仔细阅读)

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++ 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. 提交的程序代码文件的放置位置请参照具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 程序可使用的栈内存空间限制与题目的内存限制一致。
7. 若无特殊说明，每道题的 **代码大小限制为 100KB**，输入与输出中同一行的相邻整数、字符串等均使用一个空格分隔，结果比较方式为全文比较（忽略行末空格、文末回车）。
8. 选手不得使用 SSH 等命令。
9. 选手不得使用内嵌汇编，`#pragma` 等指令。
10. 评测时采用的机器配置为：Intel Xeon Silver 4210 CPU 2.20GHz，内存 32G，上述时限以此配置为准。
11. 特别提醒：评测在 Windows 10 下进行。
12. 从 PDF 复制的样例带有行号，建议直接使用样例文件。
13. 提交的代码中请勿包含 `system("pause")` 等语句，否则可能导致评测出错。

空岛生存 (survival)

【题目背景】

LeNotFound 在玩空岛生存,他需要从一个岛搭方块到另一个岛,为了防止刷怪,他选择用半砖搭路。

但是他记错了攻略,搭的全都是上半砖,还是会刷怪。

【题目描述】

他希望快速地把所有的上半砖换成下半砖,但是拆了重新搭太麻烦了,于是他想起了Debug Stick(LeNotFound 太菜了不会 fill 指令),右键方块可以改变物品状态。

一次操作可以使:

上半砖 -> 下半砖

下半砖 -> 上半砖

(箭头表示转换,与游戏内实际情况有出入,请以题面描述为准)

由于路太长,所以他疾跑 + 跳跃疯狂点击右键想把所有半砖都改成下半砖,但遗漏了好多,也有些半砖被点了偶数次。

看着参差不齐的路他没办法了,所以他找到了你。

bottom 表示下半砖, **top** 表示上半砖。

用一整行若干个字符串描述这条路径,你可以进行如下两种操作:

- 操作 1 按住右键沿着这条路走一遍,使每个方块翻转一次
- 操作 2 点击一个上半砖,使它变为下半砖

你能否在只进行 **不超过** K 次操作 2 的情况下把路修整好?如果能输出 **Yes** 否则输出 **No**。

【输入格式】

从文件 *survival.in* 中读入数据。

第 1 行一个整数 K

第 2 行一行若干个字符串,表示路上各个方块的状态。

【输出格式】

输出到文件 *survival.out* 中。

一行一个字符串，表示是否可以完成相应操作，如果能输出 **Yes** 否则输出 **No**。

【样例 1 输入】

```
1 1
2 top bottom top
```

【样例 1 输出】

```
1 Yes
```

【样例 2 输入】

```
1 13
2 top bottom bottom top top top top bottom bottom top top top top
  bottom bottom top bottom bottom top bottom bottom top bottom
  bottom top bottom bottom top bottom bottom top top bottom top
  bottom top bottom bottom top top
```

【样例 2 输出】

```
1 No
```

【样例 3】

见选手目录下的 *survival/survival3.in* 与 *survival/survival3.ans*。

【样例 3】

见选手目录下的 *survival/survival4.in* 与 *survival/survival4.ans*。

【数据范围】

对于 25% 的数据，保证输入字符串长度小于 100

对于 100% 的数据，保证输入字符串长度小于 1000000