NOIP 2020 模拟

一．题目概况

中文题目名称	字符串	选择	键盘
英文题目与子目录名	a	b	c
可执行文件名	a	b	c
输入文件名	a.in	b.in	c.in
输出文件名	a.out	b.out	c.out
每个测试点时限	1s	1s	1s
测试点数目	20	20	20
每个测试点分值	5	5	5
附加样例文件	无	无	无
结果比较方式	SPJ	全文比较（过滤行末空格及文末回车）	
题目类型	传统	传统	传统
运行内存上限	512M	512M	512M

二．提交源程序文件名

对于 C++语言	a.cpp	b.cpp	c.cpp

三．编译命令（不包含任何优化开关）

对于 C++语言	g++ -o a a.cpp -lm	g++ -o b b.cpp -lm	g++ -o c c.cpp -lm

注意事项:

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/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、全国统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz，内存 32GB。上述时限以此配置为准。

4、特别提醒：评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以其为准，不开 O2 优化。

字符串

【问题陈述】
小 A 学习了一种新的语言，其中每个单词由字符集Σ中的字符构成。对于任意两个在Σ中的字符都有一个“编辑距离”。（注意：“编辑距离”是有向的，且两个相同的字符也会有“编辑距离”）

类似地，可以定义两个长度相同的单词的“编辑距离”为对应位置的字符的“编辑距离”之和。（注意：单词的编辑距离同样是有向的）（详细可以看样例解释）

小 A 有两个喜爱的单词s,t。他想把s,t分别扩充成两个长度相同的单词s',t'，使得s'到t'的“编辑距离”最小。（扩充的要求是：s,t分别为s',t'的子序列）

小 A 向你请教最小的“编辑距离”和对应的s',t'。

【输入格式】
第一行给出字符集Σ。
第二行给出单词s。
第三行给出单词t。
假设Σ的大小为n，接下来n行每行n个整数。第i行的第j个数表示Σ中第i个字符到第j个字符的”编辑距离”。

【输出格式】
第一行一个整数表示s'到t'的最小的“编辑距离”。
第二行一个单词s'。
第三行一个单词t'。
选手要保证s'到t'的“编辑距离”最小。如果有多组可行的s',t'，输出任意一种均可。

【数据范围】
对于 30% 的数据，满足Σ的大小不超过50，s,t的长度不超过500。
对于所有数据，满足Σ的大小不超过200，s,t的长度不超过2000，输入的整数$\in [1,10^5]$。
保证可以用$scanf("\%s",\dots)$和$printf("\%s",\dots)$正确读入和输出字符串。
不保证字符的 ASCII 都为正数。

【输入输出样例】

a.in	a.out
ab ab ba 2 1 4 1	4 aba bba

【样例解释】
字符“编辑距离”：a 到 a 为 2，a 到 b 为 1，b 到 a 为 4，b 到 b 为 1。

单词“编辑距离”：aba 到 bba 为 1+1+2=4
可以证明这是最小的“编辑距离”。

选择

【问题陈述】

小 A 正在进行一个游戏。他要在一条街道上选择一些店铺。（店铺一共有n家，从左到右编号依次为$1,2,\ldots,n$）选择的店铺要求是一个连续的区间。相邻两个店铺间的街道有一个价值，初始所有价值都为零。

小 A 认为选择不同的店铺的价值是不一样的。具体地说：对于任意两个的店铺$i,j(i<j)$，产生的价值为它们之间街道的价值和。选择一些店铺的价值为：选中的店铺中，两两店铺价值的和。（详细可以看样例解释）。

小 A 发现有些时候一些街道的价值会改变，具体地说，每次改变可以描述成：在店铺$l,r(l<r)$之间的街道价值$+d$。（d可以为负数）

小 A 会在一些时候询问选择一些店铺的价值，具体地说，每次询问可以描述成：输出选择编号在l,r之间店铺的价值。

【输入格式】

第一行两个整数n,m分别表示店铺数和操作数。

接下来m行，每行有两种形式：$1\ l\ r\ d$ 表示一次改变操作，$2\ l\ r$ 表示一次询问操作。

【输出格式】

对于每次询问操作，输出一行一个整数。

【数据范围】

对于 30% 的数据，满足$n,m\le 10^3$。

对于所有数据，满足$2\le n\le 10^5; 1\le q\le 10^5; 1\le l<r\le n; 0\le |d|\le 10^4;$

【输入输出样例】

b.in	b.out
4 5 1 1 4 2 1 1 2 -1 2 1 2 2 2 4 2 1 4	1 8 17

【样例解释】

前两个操作后每段街道的价值如下：

选择区间[1,2]店铺的价值为：1
选择区间[2,4]店铺的价值为：2+4+2=8
选择区间[1,4]店铺的价值为：1+3+5+2+4+2=17

键盘

【问题陈述】

小 A 获得了一个奇怪的键盘，可以描述成一个$n \times m$的矩阵（保证nm为奇数），每个位置上有一个按键。除了左上角，在键盘上有$\frac{nm-1}{2}$个1×2的骨牌，每个骨牌可以竖着也可以横着。

对于键盘上的每个按键，会有一个对应的字母。当一个按键没有被骨牌覆盖时，可以按下这个按键。

对于每个骨牌，如果它长度为1的边的另一侧没有被其他骨牌覆盖，那么可以将它往对应方向移动一格。（详细可以看样例解释）

小 A 想按下所有对应字母a, e, i, o, u, y的按键，他希望你告诉他最少移动多少次骨牌或者不可行。

【输入格式】

第一行给出两个整数n, m。

接下来n行，每行一个长度为m的字符串，第i行第j个字符表示按键对应的字母。

再接下来n行，每行一个长度为m的字符串。其中有三种字符 $.$, $-$, $|$ 。

其中 . 表示没有被骨牌覆盖，$-$表示被一个横着的骨牌覆盖，$|$表示被一个竖着的骨牌覆盖。

保证只有第1行第1个字符为 . 。可以证明这样的描述方法唯一确定了骨牌的放置。

【输出格式】

如果不能按下要求的按键就输出一行“NO”（不带引号），不然输出一个整数表示最小的移动次数。

【数据范围】

对于 20% 的数据，满足$n, m < 5$。

对于 70% 的数据，满足$n, m < 70$。

对于所有数据，满足$1 \leq n, m < 1000$。

保证按键对应的字母仅有小写字母。

【输入输出样例】

c.in	c.out
3 3 ytr hgf dsa .-- ||| |||	2

【样例解释】

可以发现键盘对应的状态只有如下 4 种:

发现只需移动两次骨牌到达第 4 张图的状态就可以按下所有要求的按键。
可以证明这是最小的答案。