

NOI 模拟赛

一、题目概览

中文题目名称	矩阵	猜数列	围墙
英文题目名称	matrix	hidden	c
可执行文件名	matrix	hidden	c
输入文件名	matrix.in	hidden.in	c.in
输出文件名	matrix.out	hidden.out	c.out
时间限制	1s	1s	2s
空间限制	256MB	256MB	256MB
测试点数目	10	10	捆绑测试
测试点分值	10	10	捆绑测试
题目类型	传统	传统	传统
比较方式	全文比较	全文比较	spj
是否有部分分	否	否	否

二、注意事项：

1. 文件名（程序名和输入输出文件名）必须使用小写。
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. 评测环境为 Windows，使用 lemon 进行评测。
4. 开启 O2 优化，栈空间开大至 256M。

矩阵 (matrix)

【题目描述】

有一个 $n \times n$ 的矩阵，每个位置 (i, j) 如果是 “.” 表示为白色，如果是 “#” 表示为黑色。

初始时，每个位置可以是黑色或白色的， (i, j) 位置的值会作为 a_{ij} 给你。

现在有一种操作，选择两个整数 $i, j \in [1, n]$ ，记 $(i, 1), (i, 2), \dots, (i, n)$ 的颜色为 c_1, c_2, \dots, c_n ，将 $(1, j), (2, j), \dots, (n, j)$ 的颜色赋为 c_1, c_2, \dots, c_n 。

你的任务是将整个矩阵变成全黑，如果能够办到，输出最少步数，否则输出 “-1”。

【输入数据】

第一行一个整数表示 n 。

接下来 n 行，每行 n 个字符表示整个矩阵。

【输出数据】

输出文件只有一行，一个整数表示答案。

【样例输入 1】

```
2
.#
.#
```

【样例输出 1】

```
3
```

【样例输入 2】

```
2
..
..
```

【样例输出 2】

```
-1
```

【数据范围】

对于 30% 的数据，满足 $n \leq 4$ 。

对于另外 20% 的数据，满足每一列都至少有一个黑色的格子。

对于 100% 的数据，满足 $n \leq 1000$ 。

猜数列 (hidden)

【题目描述】

有一个长度为 m 的，由 1 到 9 之间的数构成的未知数列 a 。

你现在有 n 个线索，每个线索都是用如下方式生成的：

(1) 选择序列 a 的某一个位置 p 作为开始；

(2) 选择某个方向（向左或向右）；

(3) 从 p 出发往你选择的方向走，每遇到一个之前未出现的数就将它加到线索中。

现在你需要求出满足所有线索的长度最小的序列的长度。

【输入数据】

输入文件的第一行为一个整数 n ，表示线索的数量。

接下来 n 行，每行有若干个以 0 结尾的整数，表示一条线索。保证一条线索中的数在 $[1, 9]$ 中且不会出现相同的数。

【输出数据】

如果无解请输出 -1，否则输出可能的最小长度。

【样例输入 1】

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

【样例输出 1】

```
7
```

【样例输入 2】

```
3
1 2 0
2 3 0
3 4 0
```

【样例输出 2】

```
-1
```

【数据范围】

对于 20% 的数据，答案不超过 10。

对于另外 40% 的数据，保证存在一个最优解，使得所有线索都可以通过向右遍历得到。

对于 100% 的数据， $1 \leq n \leq 10$ 。

围墙 (c)

【题目描述】

在离跳蚤国很远的地方有一个蚰蚰国，最近蚰蚰国选出了一位新的领导人。

这位领导人上任之后做的第一件事，就是在蚰蚰国和它的一个邻国——蝹蝹国之间修一堵围墙。

围墙可以看成是一个长度为 n 的括号序列，与此同时还有一个长度为 n 的排列 P ，一个围墙被称为稳的，当且仅当。

(1)这个括号序列是合法的。

(2)构造一张 n 个点的图，当且仅当第 i 个位置是左括号时，点 i 向点 P_i 连边，最后形成的图必须满足每个点度数均为一。

保证对于任意 i 有 $P_i \neq i$ 。

一个括号序列合法的定义如下：

(1)空序列是合法的。

(2)如果“A”是合法的，那么“(A)”也是合法的

(3)如果“A”和“B”都是合法的，那么“AB”也是合法的。

例如“()((()))”是合法的，而“()()”不是。

现在蚰蚰国的领导人想知道一种合法的修墙方案。

【输入数据】

第一行一个正整数数 n ，含义如图所示。

接下来一行 n 个正整数表示排列 P ，保证排列合法。

【输出数据】

输出一行一个长度为 n 的括号序列，如果有多种解输出任意一种即可。数据保证有解。

【样例输入】

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

【样例输出】

```
()()
```

【数据范围】

本题有三个 subtask，只有通过一个 subtask 中的全部测试点才能获得该 subtask 中的所有分数。

subtask1: $n = 10$ ，10 分。

subtask2: $n = 40$ ，30 分。

subtask3: $n = 100$ ，60 分。