

青少年软件编程（C 语言）等级考试试卷（五级）

分数：100 题数：4

数字变换

给定一个包含 5 个数字（0-9）的字符串，例如“02943”，请将“12345”变换到它。你可以采取 3 种操作进行变换

1. 交换相邻的两个数字
2. 将一个数字加 1。如果加 1 后大于 9，则变为 0
3. 将一个数字加倍。如果加倍后大于 9,则将其变为加倍后的结果除以 10 的余数。

最多只能用第 2 种操作 3 次，第 3 种操作 2 次 求最少经过多少次操作可以完成变换。

时间限制：1000

内存限制：65536

输入

有最多 100,000 组数据 每组数据就是包含 5 个数字的字符串

输出

对每组数据，输出将"12345"变换到给定字符串所需要的最少操作步数。如果无法变换成功，输出-1

样例输入

12435

99999

12374

样例输出

1

-1

3

提示

由于测试数据太多，如果对每组数据都从头进行搜索，就会超时。建议先做预处理，即以“12345”作为初始状态做一遍彻底的广搜，找出“12345”经合法变换能够到达的所有字符串，并记录到达这些字符串各需要多少步操作。然后对读入的每组数据，在上述预处理记录的结果中进行查询即可。

寻找边缘

给定一张 $R \times C$ 的地图，由 "X" 和 "O" 组成。

现在需要重新处理这张地图，找到地图边缘的那些 "O"。你需要将这些地图边缘上的 "O" 保留下来，然后将其他的 "O" 全部替换为 "X"。

地图边缘的 "O" 指的是那些处于第一行/列或最后一行/列上的 "O"，以及从这些 "O" 的相邻位置（上下左右）延伸出去的 "O"。

时间限制：1000

内存限制：65536

输入

第一行是一个正整数 T ，表示一共有 T 组数据。对于每组数据，其第一行是两个正整数 R 和 C ，表示地图的大小，用一个空格分开。接下来的 R 行，每行包含了 C 个字符，分别是 "X" 或 "O"。其中， $0 < T \leq 10$ ， $0 < R, C \leq 500$ 。

输出

对于每组数据，输出 R 行，每行包含了 C 个字符，分别是 "X" 或 "O"。每组数据之间需要额外输出一个空行。

样例输入

```
2
2 3
OXX
XXO
5 5
XXXOX
XXXOX
X00XX
XX0XX
X0XXX
```

样例输出

```
OXX
XXO
```

XXX0X

XXX0X

XXXXX

XXXXX

X0XXX

42 点

42 是：

- 组合数学上的第 5 个卡特兰数
- 字符'+'的 ASCII 码
- 钼的原子序数
- 6 与 9 的乘积结果的 13 进制表示
- 生命、宇宙以及任何事情的终极答案
- 以及……表达式 $(1+5)/2*(6-4)*7$ 的值

因此，小机器人 Marvin 发明了这个叫 42 点的小游戏。在这个游戏中，玩家会获得 n 个数。玩家需要使用 '+'、'-'、'*'、'/'、'('、')' 以及这 n 个数构成一个合法的中缀表达式，并使得该表达式的值为 42。 n 个数之间的顺序可以改变。表达式运算过程中只能出现整数。

由于过于抑郁，Marvin 无力完成这个游戏，于是来找你帮忙。你的任务是对于给定的 n 个数，判断他们是否能根据上述游戏规则算出 42。

时间限制：1000

内存限制：65536

输入

第一行为一个数 n ， $1 \leq n \leq 6$ 。第二行为 n 个数，每个数均为 $[1,13]$ 范围内的整数。

输出

输出一行，若可以算出 42 则输出“YES”，否则输出“NO”（注意大小写）。

样例输入

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

样例输出

```
YES
```

Project Summer 游戏

小 I 和小 B 最近沉迷一款叫做《Project Summer》的游戏，小 I 扮演这个游戏中需要逃生的无辜者（Innocent），小 B 扮演这个游戏中抓住无辜者，阻止其逃生的背叛者（Betraye r）。

这个游戏的地图是一个 N 行 M 列的矩形，每个格点表示一个位置。'#' 表示地图中的障碍物，'.' 表示地图中的空地，此外，地图中还有**只有背叛者才能使用的传送门**，用小写字母 'a' - 'z' 标记，它们在地图上成对出现。

角色可以花费 1 单位的时间从一个格子走到上下左右相邻的 4 个空地中的另一个格子（不可以走出地图边界或者走到障碍物上）。

此外，当小 B 扮演的背叛者走到一个传送门上时，他可以花费 1 单位的时间从当前格子传送到与当前格子相同字母的另一个传送门处（**他也可以选择不传送，此时没有花费任何时间，待在原地不动**）。传送是双向的。比如，现在小 B 走到了标记为 'a' 的格子上，那么他可以选择花费一单位的时间传送到另一个标记为 'a' 的格子上，也可以选择~~不传送~~，那么他就待在原地不动。

现在，小 I 被小 B 的陷阱困住了，无法移动。给出地图上小 B 和小 I 所在的格子（**他们都站在空地上**），求小 B 最少需要花费多少时间才能走到小 I 所在的格子抓住他。**如果小 I 无法抓住小 B，输出 -1**

时间限制：1000

内存限制：65536

输入

第一行一个数字 T, 表示数据组数。接下来描述 T 组数据，每组数据最开始是两个正整数 N, M 表示地图是 N 行 M 列的矩形。接下来 N 行，每行 M 个字符，表示地图。在地图上，用 '.' 表示空地，'#' 表示障碍物，'a'-'z' 表示传送门，'B' 表示小 B 的初始位置，'I' 表示小 I 的初始位置。对于每组数据，保证在地图上标记相同的传送门恰好出现两次。

$T, N, M \leq 100$

输出

T 行，第 i 行输出 'Case #i: t', 表示第 i 组数据的答案是 t. 小 B 最少需要 t 单位时间才能走到小 I 所在的格子。如果小 I 无法抓住小 B，输出 -1

样例输入

```
3
5 5
Bx#..
#a.#.
.....
##...#
.x.aI
5 5
BIa.a
x#.x.
.#.##
.....
#####
2 2
B#
#I
```

样例输出

```
Case #1: 4
Case #2: 1
Case #3: -1
```

提示

对于第一组数据，假设行从上到下标号 1 到 5，列从左到右标号 1 到 5，小 B 初始在 (1, 1)。小 B 的最优路线是：(1, 1) -> (1, 2) -> (2, 2) -> (5, 4) -> (5, 5)。也就是走到标记为 x 的传送门时忽略传送门，走到标记为 a 的传送门时使用传送门。对于第二组数据，小 B 直接花费 1 单位时间向右走一格就可以抓住小 I，故输出 1。对于第三组数据，小 B 无法走到小 I 所在的位置上，故输出 -1。