

E7 - Statement

A 不要温和地走进那个良夜

题目背景

《Do Not Go Gentle Into That Good Night》（《不要温和地走进那个良夜》）是著名诗人迪伦·托马斯（Dylan Thomas）创作的一首经典诗歌。诗人通过强烈的语言，表达了对生命不屈的抗争与热爱。每一行诗都充满深意与力量，展现了诗人对亲人、生命与死亡的思考。全诗共有十九行，如下：

```
Do not go gentle into that good night,  
Old age should burn and rave at close of day;  
Rage, rage against the dying of the light.  
Though wise men at their end know dark is right,  
Because their words had forked no lightning they  
Do not go gentle into that good night.  
Good men, the last wave by, crying how bright  
Their frail deeds might have danced in a green bay,  
Rage, rage against the dying of the light.  
Wild men who caught and sang the sun in flight,  
And learn, too late, they grieved it on its way,  
Do not go gentle into that good night.  
Grave men, near death, who see with blinding sight  
Blind eyes could blaze like meteors and be gay,  
Rage, rage against the dying of the light.  
And you, my father, there on the sad height,  
Curse, bless me now with your fierce tears, I pray.  
Do not go gentle into that good night.  
Rage, rage against the dying of the light.
```

现在，输入一个正整数 i ，请你输出这首诗的第 i 行。

输入格式

不定组输入，保证不超过 20 组。

每组输入一行，一个正整数 i ，保证 $1 \leq i \leq 19$ 。

输出格式

对每组输入，输出一行字符串，表示原诗的第 i 行。

输入样例

```
1  
14
```

输出样例

```
Do not go gentle into that good night,  
Blind eyes could blaze like meteors and be gay,
```

Hint

可以考虑调用下面的数组（字符二维数组）：

```
char poem[][105] = {
    "Do not go gentle into that good night,",
    "Old age should burn and rave at close of day;",
    "Rage, rage against the dying of the light.",
    "Though wise men at their end know dark is right,",
    "Because their words had forked no lightning they",
    "Do not go gentle into that good night.",
    "Good men, the last wave by, crying how bright",
    "Their frail deeds might have danced in a green bay,",
    "Rage, rage against the dying of the light.",
    "Wild men who caught and sang the sun in flight,",
    "And learn, too late, they grieved it on its way,",
    "Do not go gentle into that good night.",
    "Grave men, near death, who see with blinding sight",
    "Blind eyes could blaze like meteors and be gay,",
    "Rage, rage against the dying of the light.",
    "And you, my father, there on the sad height,",
    "Curse, bless me now with your fierce tears, I pray.",
    "Do not go gentle into that good night.",
    "Rage, rage against the dying of the light."
};
```

Not Hint

不要温和地走进那个良夜
垂暮之年应在黄昏燃烧与咆哮；
怒斥吧，怒斥光明的消逝。
尽管智者生命尽头明白黑暗无可避免，
因他们的言辞未曾划破雷电，
他们不要温和地走进那个良夜。
善良之人，在最后的浪潮中，哭诉着
他们那脆弱的善行本可在翠绿的海湾中欢舞，
怒斥吧，怒斥光明的消逝。
狂野之人，他们曾捕捉飞驰的阳光并歌唱，
却太迟才明白，他们哀悼的是光辉的逝去，
他们不要温和地走进那个良夜。
严肃之人，临近死亡，用刺目的视线看到
失明的双眼依旧能如流星般闪耀且欢愉，
怒斥吧，怒斥光明的消逝。
而你，我的父亲，在那悲凉的高地上，
用你的激烈泪水诅咒或祝福我，我祈求着。
不要温和地走进那个良夜，
怒斥吧，怒斥光明的消逝。

Problem Author：Moca

B 懒懒排序

题目背景

小懒懒发明了一种高效且幽默的排序算法，她命名为“懒懒排序”。

懒懒排序是一种以 `lazy` 为核心的排序方式，从左到右扫描数组，仅保留符合非递减顺序的元素，其余元素直接舍弃，最终得到一个非递减的数组，时间复杂度为 $O(n)$ 。具体来说，从第一个元素开始，遍历数组，如果当前元素小于前一个被保留的元素，则舍弃它；否则将其保留。

例如，对于数组 `[1, 4, 2, 3, 6, 5, 5, 7, 7]`，经过懒懒排序后得到 `[1, 4, 6, 7, 7]`。

给定一个数组，请你对其进行懒懒排序。

输入格式

第一行包含一个整数 n ($1 \leq n \leq 10^5$)，表示数组的长度。

第二行包含 n 个整数 a_i ($-10^9 \leq a_i \leq 10^9$)，表示数组的元素。

输出格式

输出一行，包含懒懒排序后的数组，相邻两个整数之间用一个空格隔开。

输入样例

```
9
1 4 2 3 6 5 5 7 7
```

输出样例

```
1 4 6 7 7
```

Author: Moca

C 爱吃糖葫芦2

题目描述：

由于上次免单的人太多了，摊主决定在今天难为一下阿呦。于是他制定了新规则：**摊主会按照一个固定的顺序将多种食材喂给阿呦，阿呦需要直接吃掉这些食材。密语就是阿呦按吃掉食材的顺序串联所有食材名称，得到的长字符串的后半部分。**

现在，阿呦又一次来到了小摊前，他已经迫不及待想要品尝今天的糖葫芦了。请你帮助阿呦，根据摊主递给他的食材顺序，大声地说出那句密语，确保他能够准确地触发免单的优惠。

输入格式：

第一行，一个正整数 n ($1 \leq n \leq 9$)，表示可供选择的糖葫芦食材种类数量。

接着 n 行，每行包含一个字符串 s ($2 \leq |s| \leq 1000$)，且 $|s|$ 为偶数，代表一种食材的名称。

最后一行，是一个由数字组成的字符串 S_{order} ($1 \leq |S_{order}| \leq 1000$ ， $|S_{order}|$ 表示字符串长度)，每个数字 S_{order_i} ($1 \leq S_{order_i} \leq n$) 代表第 S_{order_i} 种食材，表示摊主按照这个顺序将食材喂给阿呦。

输出格式：

对于每组数据，第一行输出一行一个字符串，表示密语。

输入样例：

```
3
hawthorn
strawberry
pear
123
```

输出样例：

```
awberrypear
```

样例解释

先按摊主投喂顺序串联起所有的字符串，得到一个字符串 `hawthornstrawberrypear`，再截取字符串的后半段，即可得到密语 `awberrypear`。

D 魔法师水獭与标准咒语指南 2

题目背景

魔法师水獭正在学习标准咒语指南，书中隐藏着许多强大的咒语。为了更深入地掌握魔法，水獭希望找到某些咒语在魔法书中**所有出现的位置**。

请你写一个程序，帮助魔法师水獭在魔法书中查找每个咒语出现的所有起始位置（从 0 开始计数）。如果一个咒语没有出现，则输出 `Spell Not Found!`。

输入格式

第一行是一个字符串，表示魔法书的内容（长度不超过 1000）。

第二行一个正整数 n ，表示要寻找 n 个咒语，保证 n 不超过 100。

接下来 n 行，每行一个字符串，表示要寻找的咒语关键词（长度不超过 1000），保证不以空格开头或结尾，但是中间可能含有空格。

保证字符串中的字符都是可见字符（即 ASCII 在 $[32, 126]$ 范围内）。

输出格式

对于每个咒语关键词，输出一行：

- 如果在魔法书中找到了该关键词，输出所有匹配位置的起始索引，按升序用空格分隔。
- 如果没有找到，输出 `Spell Not Found!`。

输入样例 1

```
Expecto Patronum! Otter whispered into the darkness.
3
Patronum
to
Avada Kedavra
```

输出样例 1

```
8
5 36
Spell Not Found!
```

输入样例 2

```
aaaaaa
1
aaaa
```

输出样例 2

```
0 1 2
```

样例解释

对于第一组样例，`Patronum` 出现在第 8 个位置；`to` 出现在第 5 和 36 个位置；`Avada Kedavra` 没有出现在魔法书中。

对于第二组样例，`aaaa` 在位置 0、1、2 出现了三次。

E 字符轮换

题目描述

输入一个只包含可见字符 ($32 \leq \text{ASCII码} \leq 126$) 的字符串 s ，输出一行新字符串，要求：

- s 中各个全由非数字字符构成的极长子串（对“极长”的含义做解释：假如该子串的起始位置在原串中为 i ，结束位置为 j ，那么在不越界的情况下应满足 $s[i-1]$ 是数字字符 且 $s[j+1]$ 是数字字符，下一条的“极长”的含义同理），对应输出这些字符的ASCII码值求和后的结果。
- s 中各个全由数字字符构成的极长子串，对应输出这个子串数字作为ASCII码值对应的字符。

比如：输入 `ab97c`，那么应该输出 `195a99`，因为其可以分为三段儿极长子串 `ab`，`97`，`c`，第一段是两个非数字字符，其ASCII码相加等于195，第二段儿是两个数字字符97,97作为ASCII码对应的字符是 `a`，第三段儿由一个非数字字符 `c` 构成，其ASCII码为99，由于这个子串只有一个字符 `c` 因此求和的结果就是它本身的ASCII码值。

输入

多组输入（不超过100组）。

一共若干行，每行一行字符串表示一组数据，其仅包含可见字符，保证其中的极长数字字符子串作为ASCII码值对应的字符是可见字符，输入字符串总长度 l 满足 $1 \leq l \leq 100$ 。

输出

每组数据输出一行字符串，表示该组输入字符串按照规则进行数字和非数字字符之间轮换的结果

输入样例

```
ab97c
```

输出样例

```
195a99
```

hint

与 `scanf` 读入结束返回 EOF 不同，`gets` 函数读入结束会返回空指针。

AUTHOR: shtog

F 旋转变化！

题目描述

Xhescia得到了一个由 `*` 和 `.` 组成的方阵 A ，他决定对这个方阵动点手脚。

这个方阵有 N 行 N 列，保证 N 为一个偶数。我们用 (i, j) 表示从上往下数第 i 行的从左往右数的第 j 列的字符，即 $A_{i,j}$ 。

对于 $A_{i,j}$ ，它或者是 `*`，或者是 `.`。

现在，Xhescia将会依次进行第 $i = 1, 2, 3$ 到 $i = \frac{N}{2}$ 次操作，对于第 i 次操作，将会按照如下规则进行：

对于所有满足 $i \leq x \leq N - i + 1$ 且 $i \leq y \leq N - i + 1$ 的 $(y, N + 1 - x)$ 将此位置的字符替换为 (x, y) 的字符。**对于每次操作，所有的替换是同时进行的**

当所有的操作完成之后，请给出 A 最后的形态。

输入格式

第一行一个正整数 $N, 1 \leq N \leq 3000$

接下来 N 行，每行 N 个字符。第 (i, j) 个字符代表 $A_{i,j}$

输出格式

共 N 行，每行 N 个字符，代表最后的方阵的形态。

输入样例

```
8
.....*
.....*
.....*
****.*
****.*
**....*
**....*
*****
*****
```

The diagram illustrates the construction of a 10x10 grid of asterisks. It shows the state of the grid at each step, with the current row's pattern on the left and the current column's pattern on the right. The construction proceeds from the top-left corner, filling the grid in a sequence of steps indicated by arrows. The final state shows a complete 10x10 grid of asterisks.

G IP 地址合法性判断

《互联网用户账号信息管理规定》第十二条：互联网信息服务提供者应当在互联网用户账号信息页面展示合理范围内的互联网用户账号的互联网协议（IP）地址归属地信息，便于公众为公共利益实施监督。

合法的 IPv4 地址 格式为 $x_1.x_2.x_3.x_4$, 其中 $0 \leq x_i \leq 255$ 且 x_i 不能包含前导零.

合法的 IPv6 地址 格式为 $x_1 : x_2 : x_3 : x_4 : x_5 : x_6 : x_7 : x_8$, 其中 x_i 是一个十六进制字符串, 长度为 $[1, 4]$, 可以包含数字、小写英文字母 (a 到 f) 和大写英文字母 (A 到 F) , 在 x_i 中允许前导零。

接下来输入 n 行，每行一个字符串 IP，保证 IP 仅包含英文字母，数字，字符 `.` 和 `:`，长度不超过 100

输出

共 n 行，每行为相应的输出

若为合法 IPv4 地址，则输出 IPv4

若为合法 IPv6 地址，则输出 IPv6

若均不符合，则输出 64vPI

输入样例

```
3
106.39.42.118
2001:da8:203:dffe:a018:a32b:dbda:44f1
256.256.256.256
```

输出样例

```
IPv4
IPv6
64vPI
```

Author : munian

H 水獭哲学家就餐问题

描述

中国哲学家？法国哲学家？水獭哲学家！

n 个水獭哲学家们围坐在圆形餐桌旁就餐。因为相邻的水獭哲学家之间只放着一根筷子而水獭哲学家就餐需要同时拿走两侧筷子组成一双筷子，所以相邻的水獭哲学家不可以同时就餐。于是，聪明的水獭哲学家们决定让更饿的水獭哲学家更先开始就餐。

注意，如果两个水獭哲学家上次结束就餐的时刻不同，那么上次结束就餐时刻更小的水獭哲学家更饿。如果两个水獭哲学家上次结束就餐的时刻相同，那么编号更小的水獭哲学家更饿。

输入

第一行，正整数 n ，保证 $n \leq 10^3$ 。

第二行，正整数 a_1, a_2, \dots, a_n ，其中 a_k 表示第 k 个水獭哲学家每次就餐的耗时，保证 $\max(a_1, a_2, \dots, a_n) \leq 10^9$ 。

第三行，正整数 b_1, b_2, \dots, b_n ，其中 b_k 表示第 k 个水獭哲学家需要就餐的次数，保证 $b_1 + b_2 + \dots + b_n \leq 3 \times 10^3$ 。

输出

参考样例，按照时间顺序输出水獭哲学家的行为。

注意，对于每个时刻：

- 如果多个水獭哲学家会放回筷子，那么编号更小的水獭哲学家会更先放回筷子。
- 如果多个水獭哲学家会拿走筷子，那么更饿的水獭哲学家会更先拿走筷子。
- 如果有水獭哲学家会放回筷子且有水獭哲学家会拿走筷子，那么全部的会放回筷子的水獭哲学家放回筷子后会拿走筷子的水獭哲学家才拿走筷子。

输入样例

```
6
10002 4 1 10000 2 3
1 2 3 1 2 2
```

输出样例

```
time 0 : otter 1 pick-up
time 0 : otter 3 pick-up
time 0 : otter 5 pick-up
time 1 : otter 3 put-down
time 1 : otter 3 pick-up
time 2 : otter 3 put-down
time 2 : otter 5 put-down
time 2 : otter 4 pick-up
time 10002 : otter 1 put-down
time 10002 : otter 4 put-down
time 10002 : otter 2 pick-up
time 10002 : otter 6 pick-up
time 10005 : otter 6 put-down
time 10005 : otter 5 pick-up
time 10006 : otter 2 put-down
time 10006 : otter 3 pick-up
time 10007 : otter 3 put-down
time 10007 : otter 5 put-down
time 10007 : otter 6 pick-up
time 10007 : otter 2 pick-up
time 10010 : otter 6 put-down
time 10011 : otter 2 put-down
```

I ddz 与 vya 分苹果

题目描述

vya 买了 n 个苹果，苹果的重量分别是 w_1, w_2, \dots, w_n ，vya 先将苹果放入三个袋子中，每个袋子的编号为 1, 2, 3，保证每个苹果都会被放入袋子且每个袋子至少装有一个苹果。

ddz 分别从三个袋子中拿出一个苹果，为了让选出的苹果的重量接近，ddz 制定了一个评价指标 $s = |a_1 - a_2| + |a_2 - a_3|$ 其中 a_1, a_2, a_3 分别是袋子 1, 2, 3 拿出来的苹果的重量，ddz 会拿出使得 s 的值最小的三个苹果。但 vya 想让 s 尽可能的大，你能帮 vya 计算出他能通过控制苹果放入袋子让 s 最大达到多少。

也就是说，请你计算：

$$\max \{f(X, Y, Z) \mid X \cup Y \cup Z = \{1, 2, \dots, n\}, X \cap Y = X \cap Z = Y \cap Z = \emptyset\}$$

其中：

$$f(X, Y, Z) = \min \left\{ \sqrt{(w_x - w_y)^2} + \sqrt{(w_y - w_z)^2} \mid x \in X, y \in Y, z \in Z \right\}$$

输入

第一个数为数据组数 t ($1 \leq t \leq 10^4$)

每组数据，第一行一个整数 n ($3 \leq n \leq 10^6$)，第二行 n 个整数 w_1, w_2, \dots, w_n ($1 \leq w_1 \leq w_2 \leq \dots \leq w_n \leq 10^{18}$)

保证所有的 n 的和不大于 10^6

输出

对于每组数据，输出一行一个整数 s 的最大值

输入样例

```
3
5
1 2 3 3 5
4
8 17 19 45
8
265 265 265 265 265 265 265 265
```

输出样例

```
6
63
0
```

Author: ddz

J 摩卡与助教团建

题目背景

不久前，程设 + 航 C 助教团队去中国科技馆举行了一次趣味团建。活动中，助教们遇到了一款有趣的逻辑益智游戏：一个 5×5 的灯光网格游戏。

- 1. 网格是一个 5×5 的灯光阵列，每个灯初始状态为亮（用数字 1 表示）或灭（用数字 0 表示）。
- 2. 玩家每次可以选择点击一个灯的开关，点击后该灯及其上下左右相邻的灯的状态会发生翻转（亮变灭，灭变亮）。
- 3. 你的目标是通过一系列点击操作，使得网格中的所有灯变为亮。

大家被这个游戏难坏了，但是助教们想到他们是程设助教！于是他们拜托你写一个程序来帮他们解决这个难题。

输入格式

第一行输入一个正整数 t ($1 \leq t \leq 10$)，表示有 t 组输入。

接下来输入 t 组数据，每组包含 5 行，每行 5 个整数（0 或 1），表示网格的初始状态；两组数据之间用一个换行符隔开。

输出格式

对于每组数据，输出一个整数：如果存在解，输出一个整数 k ，表示将网格点亮所需的最少点击次数；如果不存在解，输出 -1。

输入样例

```
2
1 1 1 1 1
1 1 0 1 1
1 0 0 0 1
1 1 0 1 1
1 1 1 1 1

0 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
```

输出样例

```
1
-1
```

样例解释

对于第一组样例，不难发现点击 (3, 3) 处的灯就能完成任务，且没有步数更少的方法。

对于第二组样例，无论如何都无法点亮所有灯。

K 超级摩卡与助教团建

题目背景

不久前，程设 + 航 C 助教团队去**超级**中国科技馆举行了一次趣味团建。活动中，助教们遇到了一款有趣的逻辑益智游戏：一个 **19×19** 的灯光网格游戏。

- 网格是一个 **19×19** 的灯光阵列，每个灯初始状态为亮（用数字 1 表示）或灭（用数字 0 表示）。
- 玩家每次可以选择点击一个灯的开关，点击后该灯及其上下左右相邻的灯的状态会发生翻转（亮变灭，灭变亮）。
- 你的目标是通过一系列点击操作，使得网格中的所有灯变为亮。

大家被这个游戏难坏了，但是助教们想到他们是程设助教！于是他们拜托你写一个程序来帮他们解决这个难题。

输入格式

第一行输入一个正整数 t ($t = 1$)，表示有 t 组输入。

接下来输入 t 组数据，每组包含 **19** 行，每行 **19** 个整数（0 或 1），表示网格的初始状态；两组数据之间用一个换行符隔开。

输出格式

对于每组数据，输出一个整数：如果存在解，输出一个整数 k ，表示将网格点亮所需的最少点击次数；如果不存在解，输出 **-1**。

输入样例

```
1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

输出样例

```
141
```