

## Problem A. 树

Input file: `tree.in`  
Output file: `tree.out`  
Time limit: 2 seconds  
Memory limit: 1024 megabytes  
Compiling: `-O2 -std=c++14`

从前有一棵  $n$  个节点的树，点从  $1 \sim n$  标号；考虑一个  $1 \sim n$  的排列  $f$ ，若  $i$  与  $j$  有边相连，当且仅当  $f_i$  与  $f_j$  相连，我们就称  $f$  是该树的一个自同构。

现在给定  $K$ ，我们希望你能给出一棵树，点数在  $[1, 200]$  之间，而且其自同构数量恰好为  $K$ 。

### Input

第一行一个整数  $T(1 \leq T \leq 10^4)$ ，表示测试数据的组数。

下面  $T$  行，每行一个整数  $K(1 \leq K \leq 10^{18})$ ； $K \geq 1$  是因为显然有一个自同构： $f_i = i$ 。

保证输入的  $K$  互不相等。

### Output

对于每组数据，如果不存在这样一棵树，你只需要输出一行 `-1`。

否则先输出一行一个整数  $n(1 \leq n \leq 200)$ ，表示树的点数。

下面  $n - 1$  行，每行两个空格隔开的整数  $u, v(1 \leq u, v \leq n)$ ，表示  $(u, v)$  间连有一条树边。

如果有多种解，你只需要任意输出一种。

### Scoring

如果你有解性判断正确，但方案并不正确，你可以获得 40% 的分数。但即使你不会构造方案，也必须按格式输出任意一棵点数在  $[1, 200]$  的树。

#### Subtask 1 (points: 10)

$K \leq 10$ .

#### Subtask 2 (points: 10)

$K \leq 100$ .

#### Subtask 3 (points: 10)

$K \leq 1000$ .

#### Subtask 4 (points: 5)

保证存在整数  $t$  使  $K = t!$ 。

#### Subtask 5 (points: 10)

保证存在整数  $t$  使  $K = 2^t$ 。

#### Subtask 6 (points: 10)

$K \leq 10^5$ .

#### Subtask 7 (points: 10)

$K \leq 10^7$ .

## Subtask 8 (points: 10)

$K \leq 10^9$ .

## Subtask 9 (points: 10)

$K \leq 10^{12}$ .

## Subtask 10 (points: 15)

没有特殊性质。

## Examples

tree.in	tree.out
3	1
1	6
8	4 5
1000000000000000000	1 2
	6 5
	5 2
	2 3
	-1

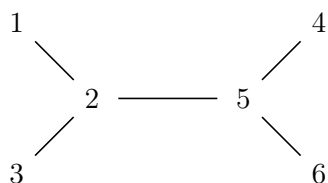
## Notes

树是  $n$  个点  $n - 1$  条边的无向连通图。

一个整数数组  $A_1, A_2, \dots, A_n$  被称为  $1 \sim n$  的排列，当且仅当  $1 \leq A_i \leq n$ ，且  $\forall i \neq j, A_i \neq A_j$ 。

树的同构可以理解成：擦去标号后，两棵树不可区分。

第二组数据的树如下图：



图中可以看到，点对  $(4, 6)$  和  $(1, 3)$  对称，左右两边整体对称，从而可得自同构数为  $2^3 = 8$ 。

## Problem B. 谜题

Input file: puzzle.in  
Output file: puzzle.out  
Time limit: 2 seconds  
Memory limit: 1024 megabytes  
Compiling: -O2 -std=c++14

从前有一个  $1 \sim n$  的排列，从这个排列我们可以得到  $n$  个  $1 \sim (n-1)$  的排列  $P_1, P_2, \dots, P_n$ ；其中  $P_i$  生成方法是：删去原排列中的  $i$ ，剩下的数保持相对位置不变，并将所有大于  $i$  的数减一。

由于一些奇怪的原因，原排列遗失了，这  $n$  个排列的顺序也被打乱了。请你根据这些信息还原出一种可能的原排列。

### Input

第一行一个整数  $n(2 \leq n \leq 1000)$ 。

下面  $n$  行，每行一个  $1 \sim (n-1)$  的排列。

同一行整数间以空格隔开。

输入保证存在至少一种可能的原排列。

### Output

一行一个  $1 \sim n$  的排列，数与数之间以空格隔开，你需要保证这是一种可能的原排列。

### Scoring

#### Subtask 1 (points: 10)

$n \leq 9$ 。

#### Subtask 2 (points: 15)

$n \leq 20$ 。

#### Subtask 3 (points: 20)

$n \leq 50$ 。

#### Subtask 4 (points: 20)

$n \leq 400$ 。

#### Subtask 5 (points: 35)

没有特殊性质。

## Examples

puzzle.in	puzzle.out
6 3 2 1 5 4 3 2 1 5 4 3 4 2 1 5 3 4 2 1 5 2 3 1 5 4 2 3 1 5 4	3 4 2 1 6 5
7 2 6 1 5 4 3 3 2 1 6 5 4 3 6 2 1 5 4 2 6 1 5 4 3 3 6 2 1 5 4 3 6 2 1 5 4 6 2 1 5 4 3	3 7 2 1 6 5 4
2 1 1	1 2

## Notes

样例 3 中，1 2 和 2 1 都正确，输出任意一个即可通过。

## Problem C. 打字机

Input file:            `letter.in`  
Output file:          `letter.out`  
Time limit:           2 seconds  
Memory limit:        1024 megabytes  
Compiling:            `-O2 -std=c++14`

你要借助一个栈打印一个仅由 **ABC** 组成的字符串  $S$ 。初始时栈为空，每次操作可以为以下三种之一：

1. 压入一个字符到栈顶；
2. 弹出当前的栈顶元素（需保证栈**非空**）；
3. 打印当前的栈顶字符（需保证栈**非空**）。

结束操作后你要保证栈**依然为空**；求打印出  $S$  的最少操作次数。

### Input

仅一行一个非空字符串  $S$  ( $1 \leq |S| \leq 5000$ )。

### Output

输出一个整数表示答案。

### Scoring

#### Subtask 1 (points: 20)

$|S| \leq 18$ .

#### Subtask 2 (points: 15)

$S$  只由 **AB** 组成。

#### Subtask 3 (points: 15)

$|S| \leq 36$ .

#### Subtask 4 (points: 20)

$|S| \leq 400$ .

#### Subtask 5 (points: 30)

没有特殊性质。

### Examples

<code>letter.in</code>	<code>letter.out</code>
ABCCBA	12
AAABAAB	13
ABCBCBACBACBCBABCA	38
ABCBCBCABCBABCBCABCBABCBCBABA	74

### Notes

栈是一种后进先出的线性表。

我们用 **A B C** 表示压入一个相应字符，**P** 表示弹栈，**Q** 表示打印。

样例 1 的一个最优的操作序列如下：

**AQBQCQPQPQP**

样例 2 的一个最优的操作序列如下：

**AQQQBQPQQBQPP**