

A 【美团杯2020】查查查乐乐

“查查查乐乐”是一段古老神秘的咒语，只有被选中的魔法师才有资格使用这一段咒语并享用它所带来的力量；而如果这段咒语出现在了不具资格的魔法师的口中，这个魔法师将会遭到咒语的反噬并付出可怕的代价。

这个学期，镁团在一家魔法早教学校做兼职，他的任务是教小学生们魔法并帮助他们准备一年一度的全国魔法奥林匹克竞赛 (NOMP)。今天，镁团在整理图书的时候，突然发现一本课外教材中包含了 t 段只由查和乐组成的咒语。让小学生们阅读这些咒语是非常危险的：他们可能会在无意识中念出“查查查乐乐”。

因此，作为一名富有责任心的儿童教师，镁团打算修改这些咒语，从而最大程度地杜绝这方面的隐患。镁团认为一段由查和乐组成的咒语是危险的当且仅当在删去咒语中的若干个字（也可以不删）后，剩下的咒语可能变成查查查乐乐。举例来说，“查查查乐乐”，“查查乐查乐乐”就是危险的，而“乐乐查查查”，“乐查乐乐查乐查查”就不是危险的。

对于每一段咒语，镁团都可以选择若干个位置并对这些位置进行修改：他可以把“查”变成“乐”，也可以把“乐”变成“查”。为了最大限度地保留教学效果，镁团希望使用尽可能少的修改来消除所有的危险性：对于每一段咒语，镁团都希望你帮他计算一下最少的修改次数。

输入格式

输入第一行是一个整数 $t(1 \leq t \leq 1000)$ ，表示咒语的数量。

对于每组数据，输入包含一行一个只包含字符 `x` 和 `l` 的字符串 $s(1 \leq |s| \leq 100)$ ，描述了一段咒语。其中 `x` 表示“查”，`l` 表示“乐”。

输出格式

对于每段咒语，输出一行一个整数表示最少的修改次数。

样例一

input

```
3
xxxll
xxlxllllxl
xxxxxlllll
```

output

```
1
1
3
```

explanation

对于三段咒语，我们分别给出一种让修改次数最小的方案：

1. `xx1111`，修改了第 **3** 个字。
2. `xx111111x1`，修改了第 **4** 个字。
3. `xx11111111`，修改了第 **3,4,5** 个字。

限制与约定

Small Task: $n \leq 10$

Large Task: $n \leq 100$ 。

时间限制：2s

空间限制：512MB

B 【美团杯2020】 图片解密

众所周知，有些时候需要进行一些花样才能把一张图片发出去。

今天，蒜斜在网上冲浪的时候，突然发现了两个神秘的文件 `small.csv` 和 `large.csv`。通过浏览相关的信息，他发现这两个文件是由两张图片分别用两种不同的方式加密而来的，但是具体的加密方法已经遗失在了时间的长河之中。

因此，蒜斜希望你能帮他还原这两张图片，并告诉他这两张图片里蕴含的信息。两张图的信息都是只包含数字和大写字母的字符串。

Small Task

提示：《大局观》。

提交文件： `picture1.out`。

Large Task

提示：新加坡樟宜机场大厅的时钟墙。

提交文件： `picture2.out`。

C 【美团杯2020】魔塔

蒜斜是个不折不扣的游戏狂魔。他的电脑上有**1000**款他最喜欢的游戏。每天早上起来，他会随机抽取两个**1**到**1000**之间的整数，第一个数表示他今天要玩的游戏，第二个数表示他要通关的次数。运气最差的一次是他抽到了“点一下玩一年”的传奇霸业，所幸的是第二个数只有**2**，不然他这辈子都通关不了。

今天蒜斜抽到的游戏是魔塔，这是一款很经典的游戏，但是他玩了半天连第**1**层都还没有通过。因此他求助北大算协的朋友把这个游戏放到了UOJ上 —— 不知道此时此刻的你能否帮他一把？

问题描述

这个魔塔游戏有 **100** 层，每一层地图的结构都完全一样，如下图所示：

```
#####  
##### Bb?Gr#####  
#####  
#?RbR GgR?B??####  
#####  
#?GrG BGR?????###  
#####  
#?BgB G?G???Rb###  
#####  
#   R RrG?B???Bg#  
# ? G #####  
#P   B?RRRGGBBB@#  
#####
```

每一个字符都代表了地图中的一个元素，其中：

1. **P** 字符表示玩家的位置，每一层开始的时候，玩家都处在左下角的位置。
2. **#** 表示墙壁，玩家移动的时候无法跨过墙壁。
3. **RGB** 分别表示红色，绿色，蓝色的门，玩家必须要消耗相同颜色的钥匙才能打开对应的门。门被打开之后将会从地图上消失。
4. **rgb** 分别表示红色，绿色，蓝色的钥匙，在到达钥匙所处所在的格子后，玩家会自动捡起这把钥匙。钥匙被捡起后将会从地图上消失。
5. **?** 是一把随机的钥匙，在到达?所处所在的格子之后，玩家将会获得一把随机颜色的钥匙，之后?会从地图上消失。
6. **@** 是终点，当玩家碰到终点的时候就会进入下一层。

在每一层游戏开始的时候，你身上没有任何钥匙。你的目标是到达字符 **@** 的位置从而进入下一层。

Small Task: 你需要达到游戏的第 **2** 层。

Large Task: 你需要达到游戏的第 **100** 层。

游戏文件

在本题最下方的下载链接中，你可以下载到魔塔游戏的可执行文件。请根据你电脑的系统挑选对应的可执行文件运行。

游戏开始时，你需要输入我们下发给你们队伍的随机种子，来表明解题人身份。**如果你使用了别的随机种子，将会导致答案被判错或者比赛被判作弊。**

在输入随机种子之后，游戏正式开始。游戏的界面如下图所示：

```
Level: 1
Answer to the small task: Unlock after reaching level 2.
Answer to the large task: Unlock after reaching level 100.
#####
##### Bb?Gr#####
##### #####
#?RbR GgR?B??####
##### #####
#?GrG BGR?????###
##### #####
#?BgB G?G???Rb###
##### #####
#   R RrG?B???Bg#
# ? G #####
#P  B?RRRGGBBB@#
#####
You keep 0 red keys, 0 blue keys, 0 green keys
Please enter your choice (move: WASD, restart: R):
```

每一步，你可以使用的操作有：

1. 输入 WSAD 来上下左右移动玩家。移动的时候可能会发生的情况有
 1. 移动的目标位置是空地，则字符 `P` 会移动到目标位置。
 2. 移动的目标位置是一把钥匙，则钥匙会被捡起，同时 `P` 会移动到目标位置。
 3. 移动的目标位置是一扇门且你有对应颜色的钥匙，则门会被打开，你会消耗一把对应的颜色的钥匙且 `P` 会移动到目标位置。
 4. 移动的目标位置是墙壁或者一扇你没有对应颜色钥匙的门，则不会发生任何变化，`P` 也不会发生改变。
 5. 移动的目标位置是 `@`，则地图会被重置成初始的样子，且地图上方的 Level 数会增加一。
2. 输入 R 来重置当前层。此时地图会被重置成最开始样子，`?` 钥匙也会被重新随机。按 R 不会重置楼层数，即在 Level 10 按下 R 重置之后玩家仍然处在 Level 10。

在玩家第一次到达 Level 2 的时候，游戏会在相同目录下自动生成文件 `small.ans`；在玩家第一次达到 Level 100 的时候，会自动生成文件 `large.ans`。提交 `small.ans` 可以获得 **35** 分，提交 `large.ans` 可以获得 **100** 分。**注意，在提交的时候需要把对应的文件改名为 `tower1.out`。**

在游戏过程中，你最多只能进行 10^5 步操作：即，向游戏输入的 `WASDR` 字符的总数不能超过 10^5 。操作数大于这个上界的提交将会被视为错误，并无法得分。

其他补充信息

`small.ans` 和 `large.ans` 是在到达对应楼层的时刻生成的，且之后不会进行更新。因此在到达对应楼层之后强制让游戏退出并不会影响答案文件的生成。

游戏流程不会进行保存，因此如果游戏过程中断了，下一次运行的时候将从 Level 1 重新开始。

在某些系统上下发的程序可能不具有可执行权限，你可能需要用 `chmod` 指令来给出对应的权限。

请使用 MacOS 的同学尽量使用命令行来打开可执行文件，而不要双击打开。

如果你的电脑系统比较小众导致下发的所有可执行文件都无法正常运行，我们深表遗憾。在这种情况下，你可以使用阿里云等云服务器来获得可以运行这些可执行文件的环境。

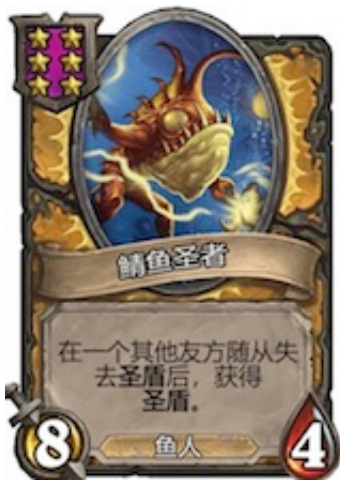
D 【美团杯2020】 版本答案

酒馆战棋也是蒜斜非常喜欢的游戏。他常常一边玩游戏一边思考着哲学问题。众所周知，当前的酒馆战棋的版本答案是多个圣盾剧毒的峰哥。作为一名善于发问的新世纪大学生，蒜斜想要明白当双方都拥有版本答案的时候，游戏会变成什么样呢？究竟是谁杀了峰哥，而峰哥又杀了谁呢？

现在，假设在酒馆战棋的一轮战斗中，战斗的一方有 n 个随从，另一方有 m 个随从，所有的随从都是圣盾剧毒的峰哥，问期望多少次进攻后战斗结束。

题目补充

在这儿，我们补充介绍一下峰哥的属性以及酒馆战棋的战斗流程。峰哥的卡牌如下图所示，在本题中，血量、攻击力、费用都没有意义，你只需要留意峰哥的特殊属性。



酒馆战棋的战斗流程为：

1. 战斗开始前，双方的随从按照从左到右的顺序被标上了序号。如果有 $k \in \{n, m\}$ 个随从，那么第一个随从的编号为 1，最后一个随从的编号为 k 。其中编号为 $i + 1$ 的随从在 i 之后，编号为 1 的随从在 k 之后。
2. 双方轮流行动，随从多的一方先手，如果双方随从数相同，则等概率随机一方开始攻击。
3. 当前攻击的一方的一个随从进行攻击。如果当前回合是这一方的第一个攻击回合，则编号为 1 的随从攻击，否则假设上一次这一方攻击的随从编号是 x ，这一次由 x 后面第一个还活着的随从攻击。注意，当只有一个随从存活的时候，可能出现 x 再次攻击的情况。
4. 攻击的随从等概率随机挑选一个还存活的敌方随从攻击。攻击结果：无论是攻击的随从还是被攻击的随从，如果身上有圣盾，则圣盾失去；如果没有圣盾，则随从死亡。
5. 峰哥的效果触发：对于每一个当前没有圣盾的峰哥，如果有一个和它同一方的其他随从（不包括自己）在刚才的攻击中失去了圣盾，则它获得圣盾。
6. 如果存在一方没有任何随从存活，游戏结束。否则进入对方回合，回到步骤 3。

如果你对战斗流程依然不了解，你可以看[这个视频](#)。该视频从 0:55 开始就对应本题 $n = m = 6$ 的情况。

输入格式

输入三个整数 $n, m, p (1 \leq n, m \leq 2000, n + m < p \leq 10^9 + 7$ 且 p 为质数)

输出格式

输出一行一个整数，表示期望轮数对 p 取模后的结果。即，如果期望轮数的最简分数表示为 $\frac{a}{b}$ ，你需要输出一个整数 c 满足 $c \times b = a \bmod p$ 。

样例一

input

```
1 1 3
```

output

```
2
```

样例二

input

```
1 2 5
```

output

```
2
```

限制与约定

Small Task: $n + m \leq 5$ 。

Large Task: $n, m \leq 2000$ 。

时间限制：2s

空间限制：512MB

E【美团杯2020】半前缀计数

蒜斜刚来PKU的时候还不知道有“北大算协”这个社团，因此他总是觉得周围的人在偷偷议论着他，比如说：

“算协（注：非蒜斜）举办的活动好有趣啊！”

“算协（注：非蒜斜）好帅啊！”

蒜斜每次听到这些话就会想入非非，但仔细想想，自己好像也没有那么帅吧？最离谱的一次还是：

“算协（注：非蒜斜）有好多小哥哥”（雾）

自从蒜斜学习了半前缀之后，他把这些话都看开了——对他来说，只要这些话里有“蒜斜好”的半前缀就足够了。

题目描述

设小写字母字符串 s ，长度为 n ， $s[l:r]$ 表示第 l 个到第 r 个字符构成的子串， $l > r$ 时对应空串。

定义半前缀是 $s[1:i] + s[j:k]$ ，其中 $0 \leq i < \text{len}(s)$ ， $i < j \leq \text{len}(s)$ ， $j-1 \leq k \leq \text{len}(s)$ 。直观上来说，你可以把半前缀理解成某一个前缀 $s[1:k]$ 删除掉某一个子串后形成的结果（当然也允许不删）。

给出字符串 s ，你需要求出 s 的所有半前缀中，有多少个不同的字符串。

输入格式

输入一行包含一个小写字母串 s ($1 \leq |s| \leq 10^6$)。

输出格式

输出一行一个整数，表示答案。

样例一

input

aab

output

6

explanation

字符串 `aab` 的半前缀有：空串，`a`，`b`，`aa`，`ab`，`aab`。

样例二

input

```
pkusaamtcup
```

output

```
217
```

限制与约定

Small Task: $|s| \leq 3000$ 。

Large Task: $|s| \leq 10^6$ 。

时间限制: 2s

空间限制: 512MB

F【美团杯2020】程序解密

今天晚上有程序设计课的DDL，但是蒜斜的程设作业还一个空格都没有打过——因为他的魔塔还没有通关……于是蒜斜打算向他的好朋友镁团求助。

镁团很愿意帮忙，所以他将自己的两个程序 `small.cpp` 和 `large.cpp` 用以下方式加密后发给了蒜斜：

1. 读入原程序中的所有字符，包括空格与换行。这样就得到了一个字符串 s 。
2. 把 s 中的每一个字符都替换成它的 ASCII 码，这样就得到了一个数字序列 A 。
3. 随机生成一个 $0 \sim 255$ 的排列 p ，并把所有的 A_i 替换成 $p[A_i]$ ，这样就得到了数字数列 B 。
4. 把 B 输出到密文文件中。

现在给出这两个程序的加密结果 `small.encode`，`large.encode`。你需要帮助蒜斜还原出这两个程序原本的功能，以帮助他完成作业。

为了帮助你解密，蒜斜还额外提供了作业里的样例输入输出 `small.in`，`small.out`，`large.in`，`large.out`：

1. 已知 `small.cpp` 在输入 `small.in` 的时候会输出 `small.out`。
2. 已知 `large.cpp` 在输入 `large.in` 的时候会输出 `large.out`。

提交方式

你提交的程序需要把 `small.cpp` 与 `large.cpp` 合并到一起：输入的第一行包含一个整数 $t \in \{1, 2\}$ ，当 $t = 1$ 的时候，你的程序需要执行 `small.cpp` 的功能；当 $t = 2$ 的时候，你的程序需要执行 `large.cpp` 的功能。

样例输入一

input

```
1
10
1 10 7 6 2 8 9 3 5 4
```

output

```
221
```

样例输入二

见样例数据下载。

限制与约定

Small Task: $t = 1$ 。

Large Task: $t = 2$ 。

时间限制: 1s

空间限制: 512MB

G 【美团杯2020】 平行四边形

蒜斜非常喜欢下围棋。自从AlphaOg面世以来，他就立志一定要研究出AlphaOg的破绽。终于，他发现当AlphaOg遇到一种特殊局面后，它的神经网络会自动输出“投降”！

随着进一步的研究，蒜斜发现这种局面有着更一般的特性，不仅仅局限于固定大小棋盘。具体来说，当棋盘大小是 n ($n + 1$ 是一个质数) 且棋盘上恰好有 n 个棋子的时候，如果这些棋子的位置满足下列条件，那么 AlphaOg 就会直接投降。假设第 i 个棋子的位置是点 P_i ，处在第 x_i 行第 y_i 列，那么这些坐标需要满足：

1. x_1 至 x_n 是 $1 \sim n$ 的排列。
2. y_1 至 y_n 是 $1 \sim n$ 的排列。
3. 这些点之间不构成平行四边形（包括退化）。即对于任何两个不完全相同的棋子对 $(P_a, P_b), (P_c, P_d)$ （允许它们之间共用至多一个棋子），线段 $P_a P_b$ 与 $P_c P_d$ 要么长度不同，要么所在的直线不平行且不重合。

凭借这项发现，蒜斜荣获了“北大算协吉祥物”的称号。如果你也能找出一种合法方案，蒜斜的称号就是你的了！

输入格式

输入第一行包含一个整数 t ($1 \leq t \leq 10$)，表示数据组数。

对于每组数据，输入第一行包含一个整数 n ($4 \leq n \leq 1000$)，保证 $n + 1$ 是一个质数。

输出格式

对于每组数据，如果无解输出一行一个整数 -1。否则输出 n 行，每行两个整数 (x_i, y_i) ($1 \leq x_i \leq n, 1 \leq y_i \leq n$)，表示第 i 个棋子的坐标。如果坐标方案不唯一，你只需要输出任意一种。

样例一

input

```
1
4
```

output

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

限制与约定

Small Task: $n \leq 12$ 。

Large Task: $n \leq 1000$ 。

时间限制: 1s

空间限制: 512MB

H 【美团杯2020】 运气游戏

可怜、蒜斜、镁团经常在一起打三人麻将。可是今天蒜斜和镁团去玩“你问你猜”了，因此可怜只能自己一个人打。可怜找了一套只有一种花色的麻将，即这套麻将中只有 9 种不同的牌，大小分别为 1 到 9，每种牌都有 4 张。为了方便，在这儿我们只考虑一个简化后的麻将规则。

定义面子为三张大小相同或者大小相邻的麻将牌，即大小形如 $i, i, i (1 \leq i \leq 9)$ 或者 $i, i+1, i+2 (1 \leq i \leq 7)$ 。定义对子为两张大小相同的麻将牌，即大小形如 $i, i (1 \leq i \leq 9)$ 。

定义一个麻将牌集合 S 是胡的当且仅当它的大小为 14 且满足下面两个条件中的至少一个：

1. S 可以被划分成五个集合 S_1 至 S_5 。其中 S_1 为对子， S_2 至 S_5 为面子。
2. S 可以被划分成七个集合 S_1 至 S_7 ，它们都是对子，且对应的大小两两不同。

举例来说，下列集合都是胡的：

1. $\{1, 1, 1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 9, 9\}$
2. $\{1, 1, 2, 2, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8\}$
3. $\{1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7\}$

而下列集合都不是胡的：

1. $\{1, 1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 9, 9\}$
2. $\{1, 1, 1, 1, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8\}$
3. $\{1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 7\}$

在麻将游戏游戏开始的时候，这 36 张牌被随机打乱并从左到右摆成一排，形成了牌山。两个牌山是不同的当且仅当存在一个位置 i ，这两个牌山第 i 个位置的牌的大小不同。经过简单的排列组合，我们可以发现一共有 $36!/(4!)^9 \approx 1.408 \times 10^{29}$ 种不同的牌山。

游戏开始时，玩家会依次从牌山的最左边摸取 13 张牌。接着游戏会进行至多 23 轮，在每一轮中：

1. 玩家先从当前牌山的最左边摸取一张牌。
2. 如果当前玩家手上的 14 张牌可以胡，玩家可以选择胡，也可以选择不胡。如果选择胡，则游戏结束。
3. 玩家从手上的 14 张牌中选一张打出去。

众所周知，麻将是一个运气游戏：当运气来了的时候，牌桌前栓一条狗都能赢。可怜随手写了三个简单的麻将 AI，在给定参数 k 的情况下，这三个 AI 会分别按照以下策略打牌：

1. 每一次要打牌的时候打出手上数值最小的牌，如果有多张则打出最早摸上来的那一张。并只有在游戏的第 k 轮的时候才会选择胡牌，其他轮数即使当前手牌是胡的也选择不胡牌。
2. 每一次要打牌的时候打出手上数值最大的牌，如果有多张则打出最早摸上来的那一张。并只有在游戏的第 k 轮的时候才会选择胡牌，其他轮数即使当前手牌是胡的也选择不胡牌。
3. 每一次要打牌的时候打出手上最早摸上来的牌。并只有在游戏的第 k 轮的时候才会选择胡牌，其他轮数即使当前手牌是胡的也选择不胡牌。

可怜发现，即使这些 AI 的策略非常的简单，但是它们也是能胡牌的。定义一个牌山为 k -天选的，当且仅当在固定参数为 k 的时候，不管选用这三个 AI 中的哪一个来打牌都能胡牌。举例来说，下面这个牌山就是一个 2-天选的牌山：

1 2 2 3 3 4 4 5 6 6 7 7 7 1 5 1 1 2 2 3 3 4 4 5 5 6 6 7 8 8 8 8 9 9 9 9

现在，可怜想要你帮她计算一些有关 k -天选牌山的数据。

Small Task

在这个部分中，你需要对于每一个 $k \in [1, 23]$ ，构造一个 k -天选的牌山。

输出格式：输出 **23** 行，每行 **36** 个空格隔开的 **1 — 9** 的数字，第 i 行描述一个 i -天选的牌山。

提交文件：mahjong1.out。

Large Task

在这个部分中，你需要对于每一个 $k \in [1, 23]$ ，计算有多少个不同的 k -天选的牌山。

输出格式：输出 **23** 行，每行一个整数。第 i 行描述 i -天选的牌山数量对 **998244353** 取模后的值。

提交文件：mahjong2.out。

I 【美团杯2020】 未来程序·改二

加入算协以后，蒜斜发现周围的小伙伴都有一项特殊的技能，那就是：算的比谁都快！有时候一个程序用电脑跑需要好几天，但是算协的小伙伴几秒钟就能手算出来。

下面是两个算协同学平时拿来练手的程序，你的任务是上传它们的运行结果——当然你可以用电脑试一试。

Small Task

你需要提交下发程序中的 `small.cpp` 的输出。

输出格式：一行一个整数，表示 `small.cpp` 的输出。

提交文件： `future1.out`

Large Task

你需要提交下发程序中的 `large.cpp` 的输出。

输出格式：一行一个整数，表示 `large.cpp` 的输出。

提交文件： `future2.out`

J【美团杯2020】分形之美

蒜斜一直怀疑自己和北大算协是高维空间中某个分形的两个分支。支撑这个观点的证据有很多。其一，他们都叫蒜斜（算协）；其二，他们都低调、奢华、有内涵，帅气、阳光、有魅力；其三，他们的好朋友都叫镁团（美团）。

蒜斜也常常思考，在高维空间里，他和北大算协之间有什么关系呢？这个问题过于高端，让我们先来看一个二维的版本吧。

定义如下的分形：

- 0 级的 ○ 是一个 1×1 的字符矩阵，里面包含了一个字符 ○ 。
- 0 级的 × 是一个 1×1 的字符矩阵，里面包含了一个字符 × 。
- i 级的 × 和 ○ 均是 $3^i \times 3^i$ 的字符矩阵，它们通过将 $i - 1$ 级别的 × , ○ 通过如下方式排列得到（左侧表示 × ，右侧表示 ○ ）：

```
xox  ooo
oxo  oxo
xox  ooo
```

举例来说，二级的 × 和 ○ 分别为：

```
xoxoooxox  oooooooooo
oxooxooxo  oxooxooxo
xoxoooxox  oooooooooo
oooxooxoo  oooxooxoo
oxooxooxo  oxooxooxo
oooxooxoo  oooxooxoo
xoxoooxox  oooooooooo
oxooxooxo  oxooxooxo
xoxoooxox  oooooooooo
```

一个人穷极一生，接触到的也只不过是世界的一部分。在这个二维的版本中，这个范围可以表示成 n 级别的 × 中的一个子矩形：它包含第 xl 行至第 xr 行的第 yl 列至 yr 列。

人一生中接触到的一些事物是连续的，就像这个分形中的 ○ 一样；而又有一些事物是突然发生的，就像这个分形中的 × 一样。而这些突然发生的事物又把连续的一生给划分成了一块一块丰富多彩的篇章。在这题中，蒜斜希望你计算 n 级 × 的某一个子矩形中， ○ 被分割成了多少个四连通块。

给定一个矩阵，两个字符 ○ 在同一个四连通块里，当且仅当从一个 ○ 出发，不停的向上下左右移动，可以在只经过 ○ 的情况下到达另一个 ○ 。

输入格式

输入第一行是一个整数 $t(1 \leq t \leq 50)$ ，表示数据组数。

对于每组数据，输入包含一行五个整数

$n, xl, xr, yl, yr (1 \leq n \leq 35, 1 \leq xl \leq xr \leq 3^n, 1 \leq yl \leq yr \leq 3^n)$ 。

输出格式

对于每组数据，输出一行一个整数表示 的四连通块个数。答案可能很大，你只需要输出对 **998244353** 取模后的值。

样例一

input

```
5
1 1 3 1 3
2 1 9 1 9
2 3 5 4 8
2 1 6 2 7
35 1 50031545098999707 1 50031545098999707
```

output

```
4
12
3
5
679477107
```

explanation

这儿以第三组数据为例，在第三组数据中，涉及到的子矩形为：

```
OOOXO
XOXOO
OXOOX
```

显然，在这个子矩形中， 被 划分成了 **3** 个四连通块。

限制与约定

Small Task: $n \leq 35, xr - xl \leq 2$ 。

Large Task: $n \leq 35$ 。

时间限制：2s

空间限制：512MB

K【美团杯2020】114514

众所周知，114514 是一个带有味道的数字。

蒜斜不太喜欢 114514，就像米斯达不喜欢 4 一样。但是总有一些讨厌的群友给蒜斜发这个数字，最开始，在蒜斜的禁言+拉黑手段下，这样的风气稍微缓解了一点。但是好景不长，群友们渐渐发明出了新的手段：有一些群友会把若干个 114514 穿插在一起，从而变成一个新的字符串。例如两个 114514 就能够穿插形成 111145144514，114151145144 等字符串。

对于一些短的字符串，蒜斜还能勉强分辨出这些字符串是怎么由 114514 得到的。但是有些时候群友会发一些特别复杂的字符串，这就让蒜斜犯了难。于是他找到了你，希望你能帮助他把一些字符串拆分成 114514，这样他就有证据可以去禁言那些讨厌的群友了。

具体来说，蒜斜会给你一些长度为 6 的倍数的只包含字符 1, 4, 5 的字符串 S 。而你要找到 $n = |S|/6$ 个序列， $x_{1,1}, x_{1,2}, \dots, x_{1,6}, \dots, x_{n,1}, \dots, x_{n,6}$ 。要求：

1. 对于每个 $i (1 \leq i \leq n)$ ，有 $1 \leq x_{i,1} < x_{i,2} < \dots < x_{i,6} \leq |S|$ 。
2. 所有的 $x_{i,j} (1 \leq i \leq n, 1 \leq j \leq 6)$ 两两不同。
3. 对于每个 $i (1 \leq i \leq n)$ ，有 $S_{x_{i,1}} = S_{x_{i,2}} = S_{x_{i,5}} = 1, S_{x_{i,3}} = S_{x_{i,6}} = 4, S_{x_{i,4}} = 5$ 。这里对于 S 中的字符，下标从 1 开始。

输入格式

第一行一个整数 $T (1 \leq T \leq 10^5)$ 表示数据组数。

接下来 T 行，每行一个字符串 $S (1 \leq |S| \leq 6 \times 10^5)$ 。保证所有的字符串长度总和不超过 6×10^5 ，且保证存在一种拆分方案。

输出格式

对于每组数据，输出 $|S|/6$ 行，每行 6 个整数，表示一种拆分方案。如果存在多种合法的拆分方案，你只需要输出任意一种即可。

样例输入

```
3
114514
111144551144
111141154115114454514414
```

样例输出

```
1 2 3 4 5 6
1 2 5 7 9 11
3 4 6 8 10 12
1 2 5 8 13 18
3 4 9 12 14 21
6 7 15 17 20 22
10 11 16 19 23 24
```

限制与约定

Small Task: 保证 S 的最后 $|S|/6$ 个字符都为 `4`。

Large Task: 无特殊限制。

时间限制: 1s

空间限制: 1024MB

L【美团杯2020】汉明距离

蒜斜一旦有作业做不出的时候，就会刷一刷北大算协的公众号。他最喜欢的一篇推文是[震惊 | 50% 程序猿都不会的算法题！](#)。每做出里面的一道题，他就感觉自己的智商提高了一些，作业里的困难也立马迎刃而解。

这一回算协的朋友送给了蒜斜一道非常很有趣的题目。所谓“独IQ++ 不如众IQ++”，蒜斜也把这道题送给了你～

题目描述

对于两个长度为 m 的01串 α, β ，定义它们的汉明距离 $d(\alpha, \beta) = \sum_{i=1}^m |\alpha_i - \beta_i|$ 。例如，当 $\alpha = 001, \beta = 100$ 时， $d(\alpha, \beta) = |0 - 1| + |0 - 0| + |1 - 0| = 2$ 。

现在给出一个神秘01串 $\pi = 1100100100001111\dots$ ，它的长度是 $n = 2^{19}$ 。我把完整的串放在了下发文件 `pi.txt` 里。

你需要写一个程序，程序输入一个长度为 n 的01串 σ 后，需要输出 $d(\pi, \sigma)$ 的估计值：设你输出的整数为 D ，则 D 需要满足不等式 $D \leq 2d(\pi, \sigma) \leq 4D$ 。

特别地，你的程序代码长度不能超过**2048B**（超出长度限制的代码无法提交）。

输入格式

输入一个长度为 2^{19} 的01串 σ 。

输出格式

输出一个非负整数，表示你给出的 $d(\pi, \sigma)$ 的估计值。

样例一

见下发文件下载。在这个样例中，汉明距离的准确值是 **3**。因此所有被视为正确的输出为 **2, 3, 4, 5, 6**。

限制与约定

Small Task: 对于所有 $11000 < i \leq 2^{19}$ 都满足 $\sigma_i = \pi_i$ 。包含 **20** 个数据，只有同时通过这些数据才能得到这一部分的 **35** 分。

Large Task: 无特殊限制，同样包含 **20** 个数据。只有同时通过 Small 和 Large 的共计 **40** 个数据，才能获得这一部分的 **65** 分。

时间限制：1s

空间限制：512MB

M【美团杯2020】最长公共子序列

蒜斜和镁团在玩一个叫做“你问你猜”的游戏（可怜去哪了？）。规则如下：

镁团手中有 n 个数，且恰好是 $1 - n$ 的排列。每次询问，蒜斜需要给出一个长度不超过 100，且每个元素都在 $1 - n$ 之间的数列（不需要是排列）；之后镁团会告诉蒜斜这两个数列的最长公共子序列长度。蒜斜需要在不超过 650 次询问内猜出镁团手中的排列。

这对于蒜斜来说实在太困难了，因此他找到了玉树临风文质彬彬英俊潇洒神采奕奕温文尔雅风度翩翩的你，你能帮助他吗？

任务

你需要编写一个函数 `find_permutation`，以确定镁团手中的排列是什么。

- `find_permutation(n, res)`
 - `n`：镁团手中排列的长度。
 - `res`：返回数组，你需要把你确定的排列存储到 `res` 中。

你可以调用函数 `get_lcs` 以帮助你确定镁团手中的排列。我们会根据你调用这个函数的次数评分。

- `get_lcs(len, A)` 接受整数 `len` 和一个长度为 `len` 的数组 `A`，并会返回数组 `A` 与目标排列的最长公共子序列长度。

在一组测试数据中，`find_permutation` 只会被调用一次。

实现细节

本题只支持 C++。

你只能提交一个源文件实现如上所述的 `find_permutation` 函数，并且遵循下面的命名和接口。

C++

你需要包含头文件 `lcs.h`。

```
void find_permutation(int n, int res[]);
```

你需要把答案排列存储在 `res[0]` 至 `res[n-1]` 中。

函数 `get_lcs` 的接口信息如下。

```
int get_lcs(int len, int A[]);
```

你需要把询问的数组存储在 `A[0]` 至 `A[len-1]` 中，数组 `A` 中的元素必须是区间 $[1, n]$ 中的整数。请保证你的所有询问都满足这个要求，不然的话可能会出现包括但不限于 `Wrong Answer`，`Dangerous Syscalls` 的评测错误。

如果有不清楚的地方，见样例及测评库下载，内附了样例程序。

评测方式

评测系统将读入如下格式的输入数据：

1. 第 1 行: n ，表示镁团手中的排列长度。
2. 第 2 行: n 个空格隔开的整数，表示镁团手中的排列。

在 `find_permutation` 返回后，评测系统将输出你的答案以及 `get_lcs` 的调用次数。

样例输入

样例一

input

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

output

```
1 5 2 4 3
10
```

explanation

样例输出的含义为 `find_permutation` 在 10 次询问之后，确定了镁团手中的排列为 **1 5 2 4 3**。

限制与约定

`find_permutation` 只能进行不超过 **650** 次询问。如果超过了这个询问数量，你的程序将无法得分。

本题严禁任何形式的攻击交互库的行为，一旦发现，将取消相关选手的参赛资格。

Small Task: $n \leq 30$

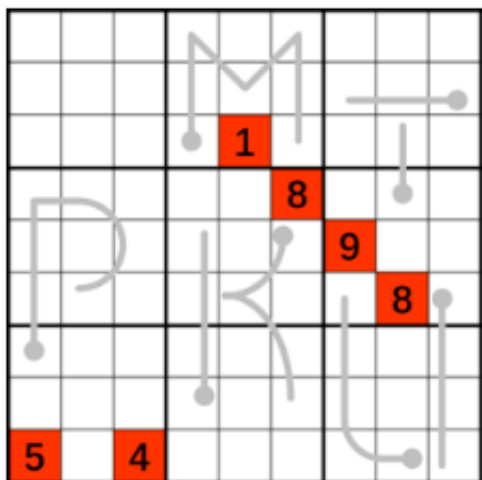
Large Task: $n \leq 100$

时间限制: 1s

空间限制: 256MB

N 【美团杯2020】热身题

本次比赛的热身题是一道数独。通过这一道热身题将能够减免 **20** 分钟的罚时。



规则：

1. 数字 1-9 在每行每列以及每个九宫格内部都必须出现恰好一次。
2. 对于每一条曲线，从带有圆球的一端开始，这条曲线严格经过（不包括角）的数字必须严格递增。

输出格式

输出 **9** 行，每行九个空格隔开的整数，表示数独的一个填法。如果有多解，你只需要提交任何一个解即可。

下面是一个样例的提交格式，直接提交它并没有办法通过本题。

[illegible]