



哈尔滨工程大学  
HARBIN ENGINEERING UNIVERSITY

计算机科学与技术学院  
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

哈尔滨工程大学

2018·第十三届校程序设计竞赛

# 题册

命题人：2014 级 ACM 校队队员

2018 年 4 月 14 日

## Christian Kramp

*Problem ID: 1001*

阶乘是基斯顿·卡曼 (Christian Kramp, 1760~1826) 于 1808 年发明的运算符号, 是数学术语。

一个正整数的阶乘 (factorial) 是所有小于及等于该数的正整数的积, 并且 0 的阶乘为 1。自然数  $n$  的阶乘写作  $n!$ 。1808 年, 基斯顿·卡曼引进这个表示法。

亦即  $n! = 1 \times 2 \times 3 \times \dots \times n$ 。阶乘亦可以递归方式定义:  $0! = 1$ ,  $n! = (n-1)! \times n$ 。

计算  $n!$  末尾有多少个 0?

### Input

输入的第一行为一个数字  $T$  ( $T \leq 100$ ), 表示数据输入的组数。

之后每行一个数字  $n$  ( $n \leq 1,000,000,000$ )

### Output

输出末尾有多少个 0

#### Sample Input

#### Sample Output

1	2
11	

# GCC

## Problem ID: 1002

GNU 编译器集合（英语：GNU Compiler Collection，缩写为 GCC），指一套编程语言编译器，以 GPL 及 LGPL 许可证所发行的自由软件，也是 GNU 项目的关键部分，也是 GNU 工具链的主要组成部分之一。GCC（特别是其中的 C 语言编译器）也常被认为是跨平台编译器的事实标准。我们在学编译原理的时候都知道，通用编译器分为词法分析、语法分析、语义分析、代码生成等过程。在 GCC 中，所有上述步骤之前还有一个重要过程：预处理，在这一过程中，编译器要将代码中的所有注释都删除掉，并将所有的头文件和宏内联到代码中，并且必须保持源代码的正确性不被破坏。因此，为了让你们知道编译原理有多复杂，HVT 想要你实现自己的 GCC 编译预处理部分中的注释删除组件。

下面是你必须要清楚的 GCC 对于 c 语言注释的处理过程：

1. 对于每一行代码中的单行注释 `//`，要将从 `//` 一直到该行末尾的所有内容删除（但是要保留最后的 `n`），举个例子：

```
int a = 2;// a is a number
```

```
a = a - 5;
```

在处理之后就是：

```
int a = 2;
```

```
a = a - 5;
```

2. 对于多行注释 `/* ... */`，要将从 `/*` 开始一直到 `*/` 之间的所有内容（可能有多行）都删除，并保留一个空格（这是为了解决形如 `int/* Integer Type */a = 6;` 这类的注释，如果不生成空格，那么处理结果就会变成 `inta = 6;` 造成语法错误），举个例子：

```
int add(int a, int b);/* add is a function
```

```
    @param a: int
```

```
    @param b: int
```

```
    return: int
```

```
*/
```

```
c = add(1,2);
```

在处理后就：

```
int add(int a, int b);
```

```
c = add(1,2);
```

注意处理后的第一行代码的末尾有一个**空格**。为了简单起见，所有多行注释的匹配都满足最小贪婪原则，即`/*`永远和最近的一个`*/`匹配，题目中保证不会出现`/* ... */ ... */`形式的多行注释。

3. 对于字符或者字符串中的任何`//`或者`/* ... */`都不应该作为注释处理，而是应该原样输出，比如：

```
char c = '/';
```

```
char str[] = "Comment in string //should not be processed";
```

在处理后的仍然是：

```
char c = '/';
```

```
char str[] = "Comment in string //should not be processed";
```

在这里要特别提示你，字符或者字符串中可能含有一类特殊字符，在 C 语言的语法中要求在其前边加上转义字符反斜杠`\`，比如在字符中不能直接出现`'`而是要用`\'`表示，即形如

```
char special_char = '\';
```

的表达式才是合法表达式

类似的还有，在字符串中出现的`"`要用`\"`表示，在字符串中的换行要用`\n`表示等等，在字符串中的反斜杠要用`\\`表示。

注意在字符中的`'`和在字符串中的`"`是不需要反斜杠`\`做特殊转义处理的，即

```
char normal_char = '';
```

```
char str[] = "Single quote ' is ok";
```

都是合法的表达式。

为了简单起见，本题中所有的转义字符在反斜杠`\`后边保证只有 1 个字符（即都是`\"``\'``\n``\t`等形式，不会出现类似 8 进制数`\054`这种形式）。

## Input

输入有且仅有一段（有多行，请读至 EOF）包含各类注释的 c 语言代码。

代码保证符合 C 语言（C99）的语法标准以及本题描述中的条件。

代码的字符总个数  $N < 100,000$ 。

## Output

输出删除注释之后的代码。

注意对于原代码中与注释无关的换行、空格、制表符等，和正常语句一样，全部保留原样输出不进行处理。

**Sample Input**

**Sample Output**

<pre>int a = 2;// a is a number a = a - 5; int add(int a, int b);/* add is a function     @param a: int     @param b: int     return: int */ char c = '/'; char str[] = "Comment in string //should not be processed"; char normal_char = ""; char str[] = "'Single quote' is ok";</pre>	<pre>int a = 2; a = a - 5; int add(int a, int b); char c = '/'; char str[] = "Comment in string //should not be processed"; char normal_char = ""; char str[] = "'Single quote' is ok";</pre>
--	---

## the Introduction of Our Lab

*Problem ID: 1003*

就像抄课文、抄古诗、写单词等一样，为了加深对本实验室的了解和印象，本题给出了关于本实验室的介绍。念在大家大多是大一的萌新，才出这样一道题，正式比赛是不可能出这么简单的哟

正式比赛时，多为全英文题目，因此，本段介绍用全英文介绍。Although the writer has not passed CET6 yet (he got 407 last time), and he will never pass it probably (he missed the latest test registration, and he is about to graduate soon), he still has a dream to be abroad studying.

Let's go back to the topic.

Our lab is aimed at taking part in the competitions regarding the programming, especially on algorithm.

A series of the competitions requires the participants doing well in using the programming skills, which are attractive to those internet companies.

Thus getting on with the competitions means that you have advantages to compete with other candidates.

So do the interviews for postgraduates.

Now, be a winner"\\(%\_%)//^`And good luck!

### Input

无

### Output

输出本题全部引号(“”)中的内容，真心希望大家仔细审题，应用 C 语言的知识。记得把他们输出在一行（输出时不能有换行，现有换行的前后内容直接接上就好）。

“

Our lab is aimed at taking part in the competitions regarding the programming, especially on algorithm.

A series of the competitions requires the participants doing well in using the programming skills, which are attractive to those internet companies.

Thus getting on with the competitions means that you have advantages to compete with other candidates.

So do the interviews for postgraduates.

Now, be a winner"\\(%\_%)//^`And good luck!”

“

By the way, our lab is located at 21B\573.”

### Sample Input

### Sample Output

Nah!	Nah!
------	------

## 安利一个活动给你和你室友无聊的时候做

*Problem ID: 1004*

室友经常使用电脑做一些事情来发泄和放飞自我，会觉得很累，腰酸背痛脖子抽筋，四肢僵硬不能动，废人……于是他们找到了一种，躺在床上，对着手机屏幕，就可以解决生理需求的事情。本人不是很喜欢做这类事情，但是能让大家一起开心，缺人的时候我也就硬着头皮上了，我一接触，也仿佛着了迷似的，果然这种从古代就有人做的事，是真的又爽，又有趣。

我大概第三轮的时候，就胡了十三幺。

因为大家家在各地，而各地的麻将将有各地的玩法。因此我们找到了一种名为“大众麻将”的玩法，谁都不懂规则，但是听起来就很容易学会。其实比较气的是，每次你想胡大的，总有人胡很小的抢在你前面胡了，就很烦。你不要在心里偷偷念叨什么“怎么不玩重庆‘血流成河’，那样就不会被截胡了”（如果不知道“血流成河”请跳过本段）之类的，在你这个年级你不应该对这些玩法了解这么深刻，快去墙角反思，为什么不务正业，为什么不爱学习，研究麻将的时间敲代码不好吗？也不要吐槽笔者研究麻将谢谢，然后继续专心做题。

有经验的人会知道在比赛时间紧迫的情况下，不要去看没用的东西，希望是有人从这里直接开始读题的，如果你上面全都读完了，希望以后能有经验。本题目要解决的是计算已经胡了的牌，总共能多少分（番数得分+杠分），大众麻将番数计算规则如下（图片为截图，版权归腾讯所有，此处不为商用引用）：

（注：图中文字为“不计 PPH”之类的，说明当前胡法是特殊的“PPH”胡法，计算番数时不要重复计算。）

**32 番：**8 种胡法，本题只记除“天胡”“地胡”之外的 6 种，即：“十八罗汉”“清龙七对”“九莲宝灯”“大三元”“大四喜”“十三幺”。计算总分时，十八罗汉下图样牌为 56 分，大三元样牌、大四喜样牌都为 64 分，之后的番型样牌的总分不再一一给出。

**16 番：**4 种胡法，分别为“字一色”“小三元”“小四喜”“清七对”。

**4 番：**3 种胡法，分别为“七对”“清一色”“混碰”。

**2 番：**7 种胡法，本题不计“杠后炮”“自摸”“抢杠胡”“杠上开花”“门前清”，只计“碰碰胡”“混一色”2 种胡法。

**1 番：**即平胡，平胡的要求为不算杠多出的一张牌。

**其他番型：**杠牌，杠只计分，不算番，本题中，所有杠都为暗杠，且杠分计算在总分中（如：“混碰”为 4 番，若其中有 1 个杠，则总分为 4（混碰番数）\*1（假设底分为 1）+2（分）\*3（其他三位玩家）\*1（杠的个数）=10 分；同理 2 个杠为 16 分，3 个杠为 22 分；但是 4 个杠时升级为“十八罗

汉”，总分为 32（十八罗汉番数）\*1（假设底分为 1）+2（分）\*3（其他三位玩家）\*4（杠的个数）=56 分）。

**【32番】**  
十八罗汉:胡牌牌型中有4种杠牌。(不计碰碰胡)

天胡:庄家发牌后直接胡牌。  
清龙七对:清一色+龙七对。(不计清一色、龙七对)

九莲宝灯:同一花色牌形成111234567899, 再摸到该种牌任何1张即可胡牌。(不计清一色, 门前清)

地胡:闲家摸第1张牌就自摸胡牌。  
大三元:胡牌牌型中有中发白3种刻(杠)牌。

十三幺:由19万, 19条, 19筒及东南西北中发白各1张, 外加上13张牌中任意1张组成的胡牌。(不计门前清)

大四喜:胡牌牌型中有东南西北4种刻(杠)牌。(不计碰碰胡)

**【16番】**  
字一色:由东南西北中发白的4种刻牌(杠)和1对将牌组成的胡牌。(不计碰碰胡)

小三元:胡牌牌型中有中发白组成的2种刻(杠)牌及1对将牌。

小四喜:胡牌牌型中有东南西北组成的3种刻(杠)牌及1对将牌。

清七对:清一色七对。(不计清一色、七对)

**【8番】**  
龙七对:七对牌型中有4张相同牌。

清碰:清一色+碰碰胡。(不计清一色、碰碰胡)

**【4番】**  
七对:由7个对子组成的胡牌。

清一色:由一种花色序数牌组合的胡牌。

混碰:混一色+碰碰胡。(不计混一色、碰碰胡)

**【2番】**  
杠后炮:玩家杠牌后打出的第1张牌是能胡的牌。  
自摸:玩家通过自摸胡牌。  
抢杠胡:其他玩家补杠的牌刚好是能胡的牌。  
门前清:胡牌时, 没有吃、碰、明杠过。

碰碰胡:由4种刻牌(杠)外加1对将牌组成的胡牌。

混一色:由一种花色序数牌及字牌组成的胡牌。

杠上开花:杠牌后补牌正是要胡的牌。

**【1番】**  
平胡:最普通的胡牌牌型。

**【其他番型】**  
明杠:其他玩家直接放杠, 杠牌后放杠玩家扣3分。

暗杠:4张牌都是玩家自己摸到的, 杠牌后其他三家都扣2分。  
明杠:碰牌后, 第4张牌是自己摸到的, 杠牌后其他三家都扣1分。

另外，给一点都不了解麻将的同学普及一下，杠后要再抓一张牌，因此，每有一个杠，最终手中会多一张牌（如：没有杠，胡牌时手中 14 张；有一个杠，15 张；两个杠，16 张……因此，四个杠叫“十八罗汉”）。

## Input

输入数据包含多组测试实例，数据的第一行是一个整数  $T$ ，用来表示实例个数，接下来是  $T$  行测试数据，每行是一串字符串（表示 14~18 张牌）。

字符串中，字符对应的牌如下：

nt: 筒、饼（整数  $n \in [1, 9]$ ），如 6t 代表“6 筒”；

ns: 索、条（整数  $n \in [1, 9]$ ），如 1s 代表“1 条”；

nw: 万（整数  $n \in [1, 9]$ ），如 6w 代表“6 万”；

DXNB: 东西南北，如 X 代表“西风”；

zfb: 中发白，如 z 代表“红中”。

## Output

对于每组实例，输出一个整数用来表示，每组占一行。

### Sample Input

### Sample Output

3	32
1t1s9tDz1w9wNXf1wb9sB	7
4s6t2t2t1s2t1s3s4s4s6t4s1s2s1s	8216
DDDDXXXXNNNNzzzzBB	

## Hint

刚知道我这个应该是一道中等难度的题，因此本人追加了数据和规则要求，具体如下：

为使题目简单一些，所有 **Input** 中的牌均可以胡，因此无需判定是否可胡，只要计算总分即可。为使题目再简单一些，**T** 不会太大，不会卡超时，不是太夸张的搜索和字符串匹配或一些高效的奇怪算法都是可以接受的。为使题目再再简单一些，去掉了太多重叠翻番的可能，如“天胡”“门前清”等（去掉的番型都在每种番型介绍里），并且底分为 1 分，使得胡牌番数即为胡牌分数，再加上杠分即为总分。但仍保留花色类等重叠翻番，如清一色十八罗汉（总分为  $4 \times 32$ （番数）+24（杠分））等。

为使题目再再再简单一些，我连题目里的大坑都告诉你了，如样例 3 所示。

样例 1 整理后是 1w1w9w1s9s1t9tDNXBzfb:

十三幺:由19万, 19条, 19筒及东南西北中发白各1张, 外加以上13张牌中任意1张组成的胡牌。(不计门清)



样例 2 整理后是 1s1s1s1s2s3s4s4s4s2t2t6t6t:



**样例 3** 整理后是 DDDDNNNNXXXXBBzzzz，十八罗汉+小四喜+字一色+4 个杠，因此，总分为  $32*16*16+24=8216$ ，虽然十八罗汉、字一色都不另计碰碰胡，但**字一色不是碰碰胡和其它番型的杂交**，可看作是独立番型，因此无需再处理十八罗汉、字一色都包含的碰碰胡。

**麻将基本术语及规则：**

基本牌型构成：**刻、将、顺、杠**。刻：三张一样的牌（如上图（样例 2）中的 1 索、2 筒）；将：两张一样的牌（如上图中的 6 筒）；顺：三个连续自然数的同花色牌（如上图中的 1 索 2 索 3 索）；杠：四张一样的牌（如上图中的 4 索（1 索被拆开了，所以不是））。特殊地，字牌不能组成顺。

胡牌时，看牌型，杠当作刻来计算（暂时忽略杠的第 4 张），则有 14 张牌。除“对”类胡法（如七对等）外，胡时要求手中有且只有一个对。即： $2（对 2 张）*1（对的个数）+3（刻 3 张）*M（刻的个数）+3（顺 3 张）*N（顺的个数）=14$ ，其中  $M+N=4$ ，且  $0 \leq M \leq 4$ ， $0 \leq N \leq 4$ 。

为防止没见过麻将并且语文不好的不认识，小乌是“1 索”，萬是“万”，東是“东”，發是“发”，两个圆角矩形加 4 个弧线是“白”，美术要是再不好不认识小鸟我也没办法了换题吧这个题你做不了实验室也不太欢迎你（不然你可能比赛的时候好多题看不明白）。

## 蹲着还是站着？

*Problem ID: 1005*

杨主席这段时间由于要找实习，所以非常焦躁。因为公司的面试都非常的难，杨主席不知道从哪入手。于是他就找了她的学长坑坑询问情况，坑坑告诉他要注重算法的学习，于是就给杨主席出了一个题目看看他算法学的怎么样，这道题是这样的：有  $N$  个人排成一排，从 1 到  $N$  按顺序依次编号，现在要执行  $N$  次操作，第一次操作让所有的人都蹲下，之后第二次操作让编号是 2 和 2 的倍数的人全部站起来，然后第三次操作让编号是 3 和 3 的倍数的人全部做相反的动作（站着的人蹲下，蹲下的人站起来），以此类推...，最后第  $N$  此操作让编号为  $N$  的这个人做相反的动作。请问  $N$  次操作后，从第  $A$  个人到第  $B$  个人之间（包括  $A$  和  $B$  这两个数字,且  $A < B$ ）有多少人是站着的？

### Input

输入数据为  $T$  组数据 ( $1 \leq T \leq 10$ )。每组数据输入包含三个数字  $N, A, B$  ( $1 \leq N \leq 1000000$ ,  $1 \leq A < B \leq N$ )。

### Output

对于每组数据，输出一个整数，表示从第  $A$  个人到第  $B$  个人之间有多少人站着。

### Sample Input

### Sample Output

1 5 1 3	2
------------	---

# 坑坑的伴随阵

Problem ID: 1006

坑坑马上就要毕业了，但是在毕业之前需要完成毕业设计才能顺利毕业，可是坑坑的毕设题目很难，是有关图像处理的，涉及到矩阵方面的知识，对数学要求很高。坑坑在研究毕设的时候遇到了一个难题，需要通过今天的比赛发现好的解题方法，希望大家可以帮忙。需要解决的问题是这样的，有一个矩阵  $A$ ，它是  $3 \times 3$  的方阵，现在需要编程实现计算矩阵  $|A^*|$ （矩阵  $A$  对应的伴随阵的行列式的值），矩阵中的每个元素值都在  $0 \sim 255$  之间。这里给出如下定义：

设矩阵  $A = (a_{ij})_{n \times n}$ ，将其元素  $a_{ij}$  所在的第  $i$  行第  $j$  列元素划去后，剩余的各元素按原来的排列顺序组成的  $n - 1$  阶矩阵所确定的行列式称为元素  $a_{ij}$  的余子式，记为  $M_{ij}$ ，我们称  $A_{ij} = (-1)^{i+j} M_{ij}$  为元素  $a_{ij}$  的代数余子式。方阵  $A = (a_{ij})_{n \times n}$  的各元素的代数余子式  $A_{ij}$  所构成的如下矩阵：

$$A^* = \begin{pmatrix} A_{11} & A_{21} & \cdots & A_{n1} \\ A_{12} & A_{22} & \cdots & A_{n2} \\ \vdots & \vdots & & \vdots \\ A_{1n} & A_{2n} & \cdots & A_{nn} \end{pmatrix} \quad \text{该矩阵 } A^* \text{ 称为矩阵 } A \text{ 的伴随矩阵。}$$

$$n \text{ 阶行列式 } \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix} \text{ 等于所有取自不同行不同列的 } n \text{ 个元素的乘积 } a_{1j_1} a_{2j_2} \cdots a_{nj_n}$$

的代数和，这里  $j_1 j_2 \cdots j_n$  是  $1, 2, \dots, n$  的一个排列，每一项都按下列规则带有符号：

当  $j_1 j_2 \cdots j_n$  是偶排列时带有正号，当  $j_1 j_2 \cdots j_n$  是奇排列时带有负号。

$$\text{这一定义可写成 } \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix} = \sum_{j_1 j_2 \cdots j_n} (-1)^{\tau(j_1 j_2 \cdots j_n)} a_{1j_1} a_{2j_2} \cdots a_{nj_n}, \text{ 这里 } \sum_{j_1 j_2 \cdots j_n} \text{ 表}$$

示对所有  $n$  级排列求和， $\tau(j_1 j_2 \cdots j_n)$  表示排列  $j_1 j_2 \cdots j_n$  的逆序数。

在一个排列中，如果一对数的前后位置与大小顺序相反，即前面的数大于后面的数，那么它们就称为一个逆序。一个排列中逆序的总数就称为这个排列的逆序数。

逆序数为偶数的排列称为偶排列；逆序数为奇数的排列称为奇排列。如 2431 中，21，43，41，31 是逆序，逆序数是 4，为偶排列。

### Input

输入数据为 T 组数据 ( $1 \leq T \leq 10$ )。每组数据有三行，每行三个数字，表示矩阵 A 中的元素值 ( $0 \leq \text{元素值} \leq 255$ )。

### Output

对于每组数据，输出一个数字，表示矩阵 A 对应的伴随阵的行列式的值。

#### Sample Input

#### Sample Output

2	1
1 0 0	1
0 1 0	
0 0 1	
0 0 1	
0 1 0	
1 0 0	

## A hard problem

*Problem ID: 1007*

Now we have a function  $f(x)$ :

```
int f ( int x ) {  
    if ( x == 0 ) return 0;  
    return f ( x / 10 ) + x % 10;  
}
```

For a given interval  $[A, B]$  ( $1 \leq A \leq B \leq 10^9$ ), calculate how many integer  $x$  that  $\text{mod } f(x)$  equal to 0.

### Input

The first line has one integer  $T$  ( $1 \leq T \leq 50$ ), indicate the number of test cases. Next  $T$  lines, Each line has one test case that has two integers  $A$  and  $B$ , separated by one blank space.

### Output

For each test case, output only one line containing the case number and an integer indicated the number of  $x$ .

#### Sample Input

```
2  
1 10  
11 20
```

#### Sample Output

```
Case 1: 10  
Case 2: 3
```

## 24 games

*Problem ID: 1008*

You have 4 numbers from 0 to 99. You need to judge whether they could operate through  $*$ ,  $/$ ,  $+$ ,  $-$ ,  $(, )$  to get the value of 24. For example, input [8, 4, 7, 1] could be operated through  $(8-4) * (7-1) = 24$ .

The division operator  $/$  represents real division, not integer division, and you cannot use  $-$  as negative sign.

Be cautious, you don't have to keep the input numbers operated in order. The answer of [1, 4, 7, 8] is the same as [8, 4, 7, 1].

### Input

A T in the first line, which tells the number of test cases ( $T \leq 1000$ )

For every line below T, there are four numbers, all of which are in the range [0, 99]

### Output

For every test case, print "ok" if the four numbers could be operated to 24, or print "no" if they couldn't.

### Sample Input

### Sample Output

3	ok
8 4 7 1	no
1 1 2 2	ok
0 0 24 1	

### Hint

$$(8-4) * (7-1) = 24$$

$$0 + 0 + 24 * 1 = 24$$

## wkroach loves mathematics

*Problem ID: 1009*

wkroach have a dizzying number of math problem need to do, but he wants to dawdle away his time with a young girl. Whatever, helping him to solve these problems, please.

In each problem, you will be given a function whose form is like  $f(x)=ax^2+bx+c$ . Your assignment is to find the maximum value and the minimum value in the integer domain  $[l,r]$ .

### Input

The first line of the input file is an integer  $T$ , indicating the number of test cases. ( $T \leq 10000$ )

In the following  $T$  lines, each line indicates a test case, containing 5 integers,  $a, b, c, l, r$ .

( $|a|, |b|, |c| \leq 1e5, |l| \leq |r| \leq 1e5$ ), whose meanings are given above.

### Output

In each line of the output file, there should be exactly two integers,  $\max$  and  $\min$ , indicating the maximum value and the minimum value of the given function in the integer domain  $[l, r]$ , respectively, of the test case respectively.

### Sample Input

### Sample Output

1 1 0 0 0 100	10000 0
------------------	---------

### Hint

$$f_{\max} = 1*100*100+0*0+0 = 10000;$$

$$f_{\min} = 1*0*0+0*0+0 = 0.$$

## wkroach is dream knight

*Problem ID: 1010*

wkroach 明天终于要去和那个女孩见面了，这天晚上在梦中他变成了一名骑士，然而 wkroach 毕竟是理工男，他变成的是棋盘上骑士，在梦醒之前他有  $N$  步移动的机会，wkroach 想知道他总共可能有多少种走法呢。

这是一个  $8*8$  的棋盘，wkroach 有一个初始位置，每次移动不能超出棋盘并且必须遵循骑士行走的规则也就和中国象棋的“马”类似但是不存在“蹩马腿”（如果你二者的规则都不知道那就看下一题吧）

注意结果要对  $1000000007 (1e9+7)$  取模。

### Input

第一行输入一个正整数  $T$  代表测试样例数目。

每组样例有三个正整数  $N R C(0 < n < 1000000000, 0 < R < 9, 0 < C < 9)$  代表此样例步数  $N$  及 wkroach 的初始点  $(R,C)$ 。

### Output

对于每组测试数据，输出一个整数，表示总的走法数，对  $1000000007 (1e9+7)$  取模。

### Sample Input

### Sample Output

2	12
2 1 1	71386775
1000000000 1 1	

## wkroach's score

*Problem ID: 1011*

wkroach 就要统计他辉煌的本科生涯的最终分数了，他决定采取一种特殊的计分方式，这里有一段长度为  $n$  的序列代表他的  $n$  门科目的分数（ $n$  最大为十万，没办法 wkroach 选修的科目太多了）。但是 wkroach 会从其中挑选出任意数量的一些科目进行分数统计，挑选出的科目组成一个新的序列但是注意这个新序列中每个科目的相对位置必须与原来序列相同（毕竟每门科目的选修时间已经无法改变）。然后 wkroach 会将新序列中每门科目的分数和它的位置标号做异或，最后将新序列中所有科目的异或结果加和就是 wkroach 最终的分数了。当然 wkroach 要这么复杂的统计自己分数的原因就是想要得到尽可能高的最终分数，请帮他算一算吧。（注意 wkroach 是一名程序员，它的序列标号都是从 0 开始的，并且他给出的原始序列中所有分数都是十六进制的）

wkroach 教导我们：

两个数异或就是将两个数换成相同长度的二进制序列之后，每一对对应位置相同则为 0 不同则为 1，得到新的二进制序列换回十进制就是这两个数的异或值。

eg:

$$2 \wedge 27 = 00010 \wedge 11011 = 11001 = 25$$

### Input

第一行输入一个正整数  $T$  代表测试样例数目

每组样例有一个正整数  $n(0 < n < 100000)$ ，代表此样例的原始序列的科目总数

然后是  $n$  个两位字符组成的十六进制数，代表具有先后顺序的每一个科目的分数

### Output

对于每组测试数据，输出一个十进制整数，表示最高分。

#### Sample Input

#### Sample Output

2	29
3	4
03 00 1B	
3	
01 00 02	

### Hint

$$(0 \wedge 03) + (1 \wedge 00) + (2 \wedge 1B) = 3 + 1 + 25 = 29$$

$$(0 \wedge 01) + (1 \wedge 02) = 1 + 3 = 4$$

## wkroach studying abroad

*Problem ID: 1012*

wkroach 就要出国留学了，这段时间他陷入了申请学校的泥潭中无法自拔。最终他决定发挥他一代天才码农的真正本领，首先他使用爬虫获取了往年申请到每个常春藤 offer 的大佬的资料，然后他需要找出这些大佬的共同点，结果 wkroach 发现大佬的共同点也很多。无奈之下 wkroach 决定先找出最重要的那部分 共同点，wkroach 针对每一所学校都找了三个大佬的数据，现在需要你帮助他找出这三位大佬的数据中最明显最长的那个共同点。

### Input

第一行输入一个正整数 T 代表测试样例数目 ( $T \leq 100$ )

每组样例有三段字符串，字符串只包含小写字母，每段字符串长度小于等于 5000。

### Output

对于每组测试数据，输出一个整数，代表这三段字符串的最长公共子串的长度。

### Sample Input

### Sample Output

2	20
mathematicalachievementfirst	4
physicalachievementfirst	
chemicalachievementfirst	
beautifulgirla	
sunshinegirlb	
smartgirlc	

### Hint

case1: "icalachievementfirst"

case2: "girl"

## The water problem

*Problem ID: 1013*

欢迎参加哈尔滨工程大学第十三届程序设计竞赛，请对于每组样例输出  $a*b+c$  的值。

### Input

第一行输入一个正整数  $T$  代表测试样例数目

每组样例有三个整数  $a\ b\ c$  ( $-1000 < a, b, c < 1000$ )

### Output

对于每组测试数据，输出一个整数，表示  $a*b+c$  的值。

#### Sample Input 1

#### Sample Output 1

2	5
1 2 3	10
10 0 10	