

Problem A. As Much As 996

输入文件: standard input

输出文件: standard output

又一次见面了！唐纳德以及所有裁判都准备好迎接这场比赛了，你呢？虽然裁判们对这场比赛的难度的看法并不相同，但是大家还是一致认为这场比赛的绝大部分题目比常规月赛更加有趣。祝你们好运，享受比赛！

作为本次比赛的第一道题目，它将是很容易的。

你可能知道程序员发明了一种表示他们的工作时间的方式：*std*，即在上午 s 时刻开始工作，在下午 t 时刻结束工作，每周工作 d 天。

你可能还知道最近有一个名为 996.ICU 的活动，该活动反对长时间工作并要求休息权利。

由于你还在大学或高中学习，所以这个题目的目标不是要求你支持这个活动，而是要编写一个程序，根据“*std* 表示法”输出程序员每周的工作时间长度。

输入格式

输入只有一行，包含一个 *std*，中间没有空格。保证 $2 \leq s, t \leq 11$ ， $1 \leq d \leq 7$ 。请注意， s 和 t 不会是 12 或 1，因为上午 12 点和下午 12 点经常会引起歧义，而凌晨 1 点或下午 1 点也不太可能是人们开始工作或停止工作的时间。

输出格式

输出每周工作小时数。注意：答案是唯一的。

样例

standard input	standard output
996	72
955	40
8107	98
1127	21
2117	147

Problem B. Black Peter

输入文件: standard input

输出文件: standard output

Black Peter 是一个适合双人或多名玩家的维多利亚牌纸牌游戏, 该游戏起源于一个古老的赌博游戏, 输家被要求买饮料, 在西方国家该游戏也被称为魔法婆婆, 类似中国的“抓乌龟”游戏。

市场上有精心制作的魔法婆婆游戏卡片出售。但我们也可以使用普通的 52 张扑克牌进行游戏。当用普通扑克牌作为游戏卡片时, 我们抽走其中的大王或小王, 或者是再添加一个 A 牌, 这样一个王或一个 A 由于没有牌与它成对而成为单牌。不过在实际游戏过程中, 我们通常是在发牌之前, 从牌中随机取走一张牌, 这样玩家将无法知道哪张牌会成为单牌。这张单牌就被称为“魔法婆婆”, 游戏结束时, 持有“魔法婆婆”牌的玩家则为输家。

游戏过程: 假设有两个玩家 Brett 和 Caoimhe。Caoimhe 先将手上的牌反面扣好, 保证牌不被 Brett 看到。然后, Brett 摸走 Caoimhe 手中的一张牌, 如果 Brett 手上有张牌与摸到的这张牌成对, 那 Brett 把这成对的两张牌面朝上丢弃在桌子上。接着 Caoimhe 按同样的游戏规则摸走 Brett 的一张牌。两人轮流摸牌, 丢弃成对的牌, 直到一个玩家手上的牌全部打完, 另一个玩家手上只拿着一张“魔法婆婆”牌, 则游戏结束。拿“魔法婆婆”牌的玩家则为输家。

作为观察者, 你事先知道 Brett 和 Caoimhe 手上的牌, 因此想要分析 Brett 获胜的概率, 则需要给出一些假设条件:

1. 无论是游戏开始还是游戏过程, 只要玩家看到任何成对牌, 都会立即丢弃它们。
2. 当玩家摸走对手手中的一张牌时, 每张牌被摸走的概率是相同的。

输入格式

第一行输入一个正整数 t , 表示测试数据的组数。每组测试数据包含三行: 第一行为一个整数 n ($2 \leq n \leq 10^6$)。一副扑克牌有两种花色, 每种花色有 n 张, 扑克牌点数从 1 到 n , 总共有 $2n$ 张扑克牌。其中一张扑克牌被抽走了, 因此实际可用的扑克牌只有 $2n - 1$ 张。接下来两行, 分别表示 Brett 和 Caoimhe 手上的牌 ($b_1 b_2 \dots b_n, c_1 c_2 \dots c_n, b_i, c_i \in \{0, 1, 2\}$)。每行为一个长度为 n 的字符串, b_i 为 Brett 拥有的点数为 i 的扑克牌数量, c_i 为 Caoimhe 拥有的点数为 i 的扑克牌数量。数据保证: $1 \leq i \leq n, 1 \leq b_i + c_i \leq 2$, 而且只有唯一一个 k , 使得 $b_k + c_k = 1$ 。游戏开始时, 每个玩家至少有一张扑克牌, 即 $\sum_{i=1}^n b_i > 0$, 且 $\sum_{i=1}^n c_i > 0$ 。

输出格式

每组测试数据, 输出 Brett 获胜的概率。只要绝对误差或相对误差不超过 10^{-9} 则输出结果视为正确。换言之, 假设你的输出结果为 a , 而裁判的答案为 b , a 和 b 视为相等当且仅当 $\frac{|a-b|}{\max(1, |b|)} \leq 10^{-9}$ 。

样例

standard input	standard output
3	0
3	0.75
100	0.3333333333333333
022	
10	
2020202101	
0202020111	
7	
1111112	
1111100	

Problem C. Coprime-Free Permutation

输入文件: standard input

输出文件: standard output

从 $1, 2, 3, \dots, n$ 中选择尽可能多的数, 然后将选出的数重新排列, 使得任意相邻的两个数的最大公约数不小于 2。

输入格式

在一行中输入一个整数 n ($4 \leq n \leq 10^6$)。

输出格式

第一行输出一个正整数 k : 表示选出的整数个数。第二行依次输出 k 个整数: a_1, a_2, \dots, a_k ($1 \leq a_i \leq n$), 整数之间用一个空格分隔, 对于所有 $1 \leq i < k$, 有 $\gcd(a_i, a_{i+1}) \geq 2$ 。如果有多个数量最多的方案, 输出任何一个即可。

样例

standard input	standard output
4	2 2 4
19	14 7 14 2 12 10 5 15 3 6 9 18 4 16 8

Problem D. Distinct Tautonyms

输入文件: standard input

输出文件: standard output

我们称一个序列为“同义序列”当且仅当其可以通过连接某一长度至少为 1 的序列的两份拷贝得到。例如, 序列 1 1 和序列 2 1 1 2 1 1 都是“同义序列”, 而序列 1 和序列 1 3 都不是“同义序列”。

给定一个序列, 请你计算出其有多少个互不相同的子序列 (不一定连续) 是“同义序列”。输出满足要求的子序列数目模上 $10^9 + 7$ 的值。

给定两个子序列 $a_{i_1}, a_{i_2}, \dots, a_{i_p}$ 和 $a_{j_1}, a_{j_2}, \dots, a_{j_m}$, 若满足 $p \neq m$ 或者至少存在一个 $k (1 \leq k \leq p)$ 使得 $a_{i_k} \neq a_{j_k}$, 我们则称这两个子序列是不相同的。

输入格式

输入的第一行为一个整数 n ($2 \leq n \leq 700$)。

输入的第二行为 n 个由空格隔开的整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$)。

输出格式

输出结果模 $10^9 + 7$ 的值。

样例

standard input	standard output
4 1 2 1 2	3
7 7 6 5 4 3 2 1	0
6 1 3 3 3 3 1	3

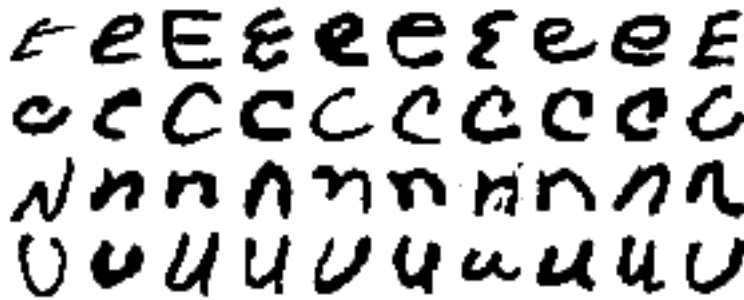
Problem E. ECNU MNIST

输入文件: standard input

输出文件: standard output

kblack: 本题翻译屎烂无药可救, 稍微改了一点改不动了, 完全不建议阅读中文题面。zerol: 尤其是附录部分。

对于那些有过深度学习经验的人来说, 理解这项任务可能会容易一些, 但是出题人 Donald 向你保证, 实现起来并不十分方便, 而且也不是比赛中最容易解决的问题之一。如果你还想看一看, 我们就开始吧。



此任务的主要目标是将四个英文字母“E”、“C”、“N”、“U”分类。你需要接受大约 2000 张图片的输入, 猜测是四个字母中的哪一个, 然后输出你的预测。如果你做得足够好, 你会得到 AC。有关详细信息, 请参阅规格说明部分。

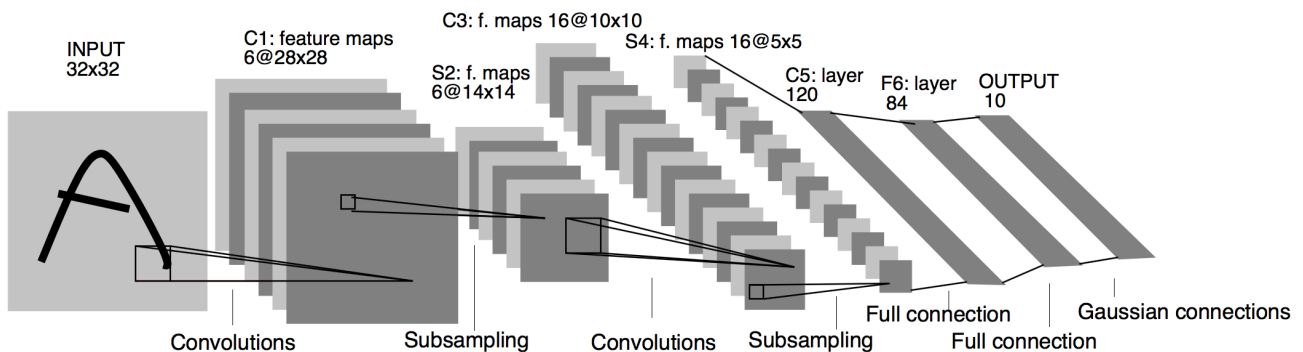


Figure 1: LeNet-5 Architecture, similar to what you are going to see, with subtle differences

题目要求

此任务的测试数据生成与通常的略有不同。Donald 建立了一个序列神经网络, 如下所示:

1. 输入: 大小为 28×28 的二值化¹图像。只有一个通道 (黑白)。
2. 卷积层 1: 卷积核大小为 5×5 , 1 个通道输入, 4 个通道输出, 输出大小为 $4 \times 24 \times 24$ 。
3. 最大池化: 池大小为 2×2 , 输出大小为 $4 \times 12 \times 12$ 。
4. 应用激励函数 ReLU。
5. 卷积层 2: 卷积核大小为 3×3 , 4 个通道输入, 9 个通道输出, 输出大小为 $9 \times 10 \times 10$ 。

¹二值化是指如果像素强度小于某个值, 则将图像中的每个像素替换为黑色像素; 如果像素强度大于或等于某个值, 则将图像中的每个像素替换为白色像素。

6. 最大化池：池大小为 2×2 ，输出大小为 $9 \times 5 \times 5$ 。
7. 应用激励函数 ReLU。
8. 展平：按照下标字典序 ($M_{0,0,0}, M_{0,0,1}, \dots$) 将多维数组调整为 225 维向量。
9. 全连接层 1：从 225 维降至 64 维。
10. 应用激励函数 ReLU。
11. 全连接层 2：从 64 维降至 4。找出最高分：如果为 0，则预测值为“C”；如果为 1，则预测值为“E”；如果为 2，则预测值为“N”；如果为 3，则预测值为“U”。

然后，Donald 应用 SoftMax，计算交叉熵，花几分钟时间进行训练，并导出训练后网络中的所有参数。在附录部分，有一个神经网络教程，将解释所有的细节。

现在，Donald 假设你已经了解了这一工作原理，并将解释所给输入的格式。

第一行包含一个魔法数字；用于测试生成目的；你可以安全地忽略它。

卷积层的参数由两部分组成：权重（一个大小为 $channel_{out} \times channel_{in} \times 5 \times 5$ 的多维矩阵）和偏置（一个维数为输出通道数的向量）。给出了一个 $s_1 \times s_2 \times \dots \times s_k$ 的多维矩阵作为 $(\prod_{i=1}^{k-1} s_i) \times s_k$ ，二维矩阵的输入，其中的行顺序按照下标字典序；一个 n 维向量用一行用空格分割的 n 个数给出。

全连接层的参数包括权重 ($(D_{out} \times D_{in})$) 和偏置 (D_{out} 向量)。

卷积层 1，卷积层 2，全连接层 1 和全连接层 2 的参数依次给出如下：

1. 卷积层 1 weight: $4 \times 1 \times 5 \times 5$.
2. 卷积层 1 bias: 4 维向量.
3. 卷积层 2 weight: $9 \times 4 \times 3 \times 3$.
4. 卷积层 2 bias: 9 维向量.
5. 全连接层 1 weight: 64×225 .
6. 全连接层 1 bias: 64 维向量.
7. 全连接层 2 weight: 4×64 .
8. 全连接层 2 bias: 4 维向量.

然后，你会看到图像。测试数据最初是从 EMNIST 数据集中提取的，每个字母有 5600 个样本。为了进行交叉验证，Donald 将测试数据分成十个子集，每个子集 2240 个样本。除示例外，每个测试都在其他 9 个测试上训练了参数，并呈现了模型以前从未见过的一个子集。这样你在一行中会看到一个 $t = 2240$ ，下面的 $28 \cdot t$ 行包含 t 个 28×28 的 01 图像。并输出你认为的结果。

Donald 向你保证，成功实现此神经网络将使你获得 97% 以上的准确率。你只需要达到 91.02% 就能得到 AC。

你可以使用任何其他方法，而不需要给定的参数，这不违反任何规则。你们都是有才华的程序员，Donald 相信你们至少会找到一些东西。

样例见测试 OJ 系统题目。

样例

standard input	standard output
42	E
<20x5 matrix omitted>	C
<1x4 matrix omitted>	N
<108x3 matrix omitted>	U
<1x9 matrix omitted>	
<64x225 matrix omitted>	
<1x64 matrix omitted>	
<4x64 matrix omitted>	
<1x4 matrix omitted>	
4	
00000000000000000000000000000000	
00000000000000000000000000000000	
00000000000000000000000000000000	
0000000111100000001111111000	
0000001111111111111111111000	
00000001111111111111111000000	
0000001111111111100000000000	
0000001111100000000000000000	
0000001111000000000010000000	
0000001110000000001110000000	
000000110000000011110000000	
0000111100000001111110000000	
0000111100111111110000000000	
0000111111111111000000000000	
0000111111111111000000000000	
0001111100000000000000000000	
0000111000000000000000000000	
0001111000000000111000000000	
0001111010000000111000000000	
0001111111111111100000000000	
0001111111111111000000000000	
0000111111111110000000000000	
0000111111110000000000000000	
0000000000000000000000000000	
0000000000000000000000000000	
0000000000000000000000000000	
0000000000000000000000000000	
0000000000000000000000000000	
< 下面省略了 3 个 28x28 的图像 >	

如果你需要完整的输入文件，你可以在题目页面找到完整样例的下载地址。

附录

在进入卷积层之前，让我们复习一下神经网络的一些基本概念。如果你以前没有学过，这也是一个好的开始。

1. 张量 (Tensor): 在这个任务中，你可以安全地把它作为向量、矩阵或多维矩阵。
2. 全连接层: 假设输入为向量 t ，输出为另一个向量 y ，then $y = a(Wt + b)$ 称为全连接层，其中 W 为矩阵， b 为向量， a 为激励函数。
3. 激励函数: 为网络增加非线性。在我们的任务中，只使用了一个激励函数: ReLU。ReLU 适用于向量或矩阵，对每个元素应用。ReLU(x) = max(0, x)。

卷积层²

密集连接层和卷积运算的专门化层（我们称之为卷积层）之间的根本区别，是密集层在其全局输入空间中学习全局模式，而卷积层在二维的小窗口中学习局部模式。

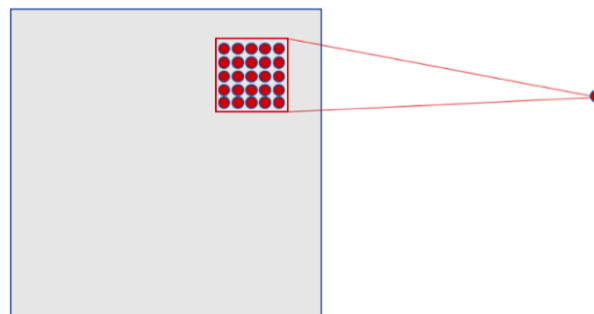
直观地，我们可以说，卷积层的主要目的是检测图像中的特征或视觉特征，如边、线条、色滴等。这是一个非常有趣的特性，因为一旦在图像中某个特定点上学习到一个特征，它就可以稍后在其任何部分中能识别这个特征。相反，在一个密集连接的神经网络中，如果它出现在图像的一个新位置上，它必须重新学习模式。

卷积层另一个重要特性是可以通过保留空间关系来学习模式的空间层次。例如，第一卷积层可以学习边等基本元素，第二卷积层可以学习由上一层中学习的基本元素组合的模式。直到它学会了非常复杂的模式。这使得卷积神经网络能够有效地学习日益复杂和抽象的视觉概念。

通常，卷积层在称为特征映射的三维张量上工作，具有高度和宽度的两个空间轴，以及一个通道轴，也即深度。对于 RGB 彩色图像，深度轴的维度为 3，因为彩色图像有三个通道：红色、绿色和蓝色。对于黑白图像（如 MNIST 数字），深度轴维度为 1（灰度）。

在 MNIST 的案例中，作为神经网络的输入，我们考虑一个二维神经元的空间 28×28 (高度为 = 28, 宽度为 = 28, 深度为 = 1)。第一层隐藏的神经元连接到我们讨论过的输入层的神经元，将执行我们刚才描述的卷积操作。但是，正如我们已经解释过的，并不是所有的输入神经元都与第一级隐藏神经元的所有神经元相连，比如密集连接的神经网络这个例子；它只由存储图像像素的输入神经元空间的小局部区域完成。

这个解释形象地可以表示为：

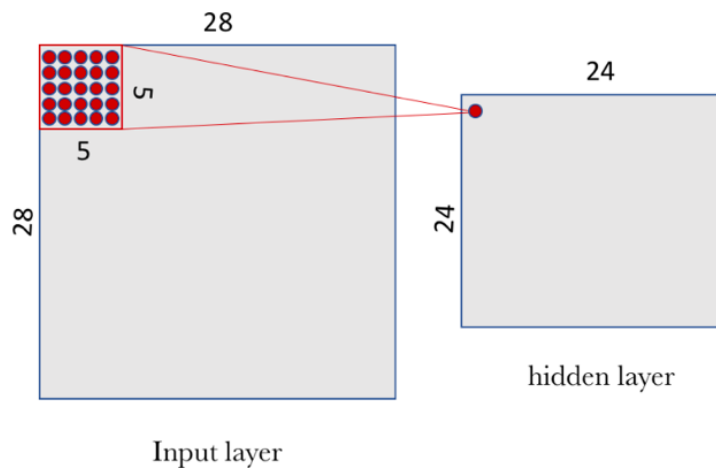


²Convolutional Neural Networks for Beginners, Practical Guide with Python and Keras, Jordi TORRES.AI, Sep 24, 2018, with modifications

在上一个例子中，我们隐藏层的每个神经元将连接到输入层 (28×28) 的 5×5 (即 25 个) 神经元。直观地说，我们考虑一个 5×5 大小的窗口，它滑过整个图像 28×28 神经元输入层。对于窗口的每个位置，都有一个隐藏层中的神经元处理这些信息。

形象地说，我们从图像左上角的窗口开始，而且为隐藏层的第一个神经元提供必要的信息。然后，我们将窗口向右滑动一个位置，将该窗口中包含的输入层的 5×5 个神经元与隐藏层的第二个神经元“连接”。因此，我们依次地，从左到右，从上到下，遍历输入层的整个空间。

分析一下我们提出的具体情况，我们注意到，如果我们有 28×28 像素的输入和 5×5 的窗口，这就定义了在第一隐藏层中的 24×24 个神经元空间，因为我们只有将窗口右移 23 个神经元，下移 23 个神经元，才能到达输入图像的右边或者底部边界。



我们要向读者指出，我们所做的假设是，当开始新的行时，窗口将向前移动 1 个像素，包括水平和垂直移动。因此，在每一步中，新窗口都与前一个窗口重叠，除了我们已经推进过的这一行像素除外。

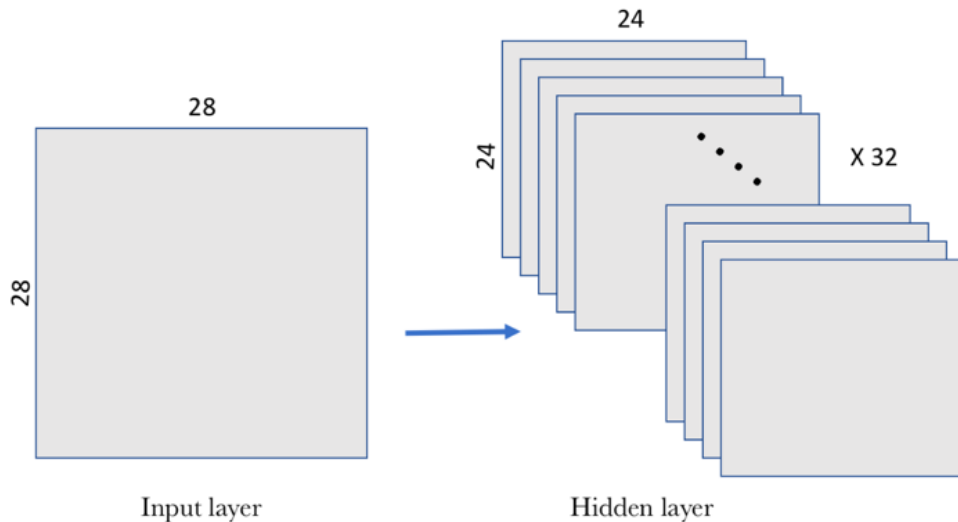
在我们研究的案例中，按照前面提到的形式，为了将隐藏层的每个神经元与输入层的 25 个相应神经元“连接”，我们将使用一个偏差值 b 和一个 5×5 的 W —权重矩阵，我们称它们为过滤器（或核）。隐藏层的每个点的值对应于过滤器和输入层的 25 个神经元 (5×5) 之间的点积。

然而，卷积网络的特殊和非常重要的一点是，我们对隐藏层中的所有神经元使用相同的过滤器（相同的加权 W 矩阵和相同的 b 偏差）：在我们的例子中，对于第一层的 24×24 个神经元（总共 576 个神经元）。读者可以看到，在这种特殊的情况下，这种共享极大地减少了神经网络的参数数量，如果我们不这样做：将不得不把 14400 ($5 \times 5 \times 24 \times 24$) 个参数，调整到 25 (5×5) 个参数加上偏差 b 。

这个共享的 W 矩阵和 b 偏差，我们已经说过，在这个卷积网络的上下文中我们称之为过滤器，类似于我们用来修饰图像的过滤器，在我们的例子中，它用于在小的条目组中寻找局部特征。

但是由矩阵 W 和偏差 b 定义的过滤器只允许检测图像中的一个特定特征；因此，为了执行图像识别，建议同时使用多个过滤器，每个过滤器对应我们想要检测的特征。这就是为什么一个完整的卷积层在一个卷积神经网络包括几个过滤器。

下图显示了这个卷积层常用方法的直观表示，其中隐藏层的层级由几个过滤器组成。在我们的示例中，我们建议使用 32 个过滤器，其中每个过滤器都用 5×5 的 W 矩阵和 b 偏差进行定义。



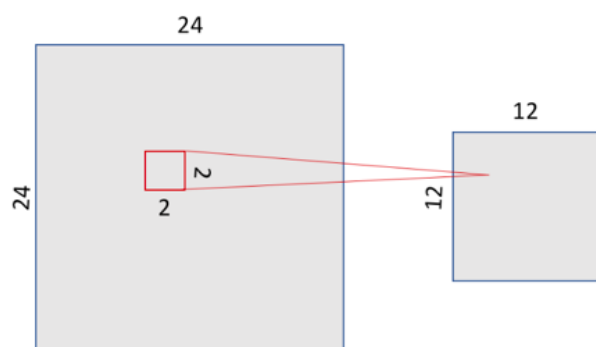
在本例中，第一卷积层接收一个输入张量 $(28, 28, 1)$ ，并生成一个输出 $(24, 24, 32)$ ，该三维张量的输出包含 32 个 24×24 像素的结果，由输入上的 32 个过滤器计算出来的。

另外，请注意，如果你有多个通道/过滤器作为输入，则你输出的每个过滤器都来自于应用核的每个输入的过滤器总和。这使得一共有 输出通道数 \times 输入通道数的过滤器，但仍然存在输出通道数的偏差，需要澄清。

池化层

除了我们刚刚描述的卷积层之外，卷积神经网络还伴随着带池化层的卷积层，池化层通常立即应用在卷积层之后。理解这些层做什么的首选方法是去明白池化层简化卷积层收集的信息和创建其中包含信息的压缩版本的相关内容。

在 MNIST 示例中，我们将选择卷积层的一个 2×2 窗口，并在池层的一个点上合成信息。可以形象地表示如下：



有几种压缩信息的方法，但是在我们的示例中，我们将使用一种称为 max pooling（极大池化）的常见方法，在我们的示例中，这个方法作为一个值保留了在 2×2 输入窗口中那些信息的最大值。在这种情况下，我们将池化层输出的维度除以 4，留下一个 12×12 的图像。

Problem F. Foreigners's Trouble

输入文件: standard input

输出文件: standard output

外国人很喜欢用缩写词, 即使有时他们会忘了缩写词的含义。例如, 今天你了解到 ECNU 可用于指代华东师范大学 (East China Normal University), 但是第二天你可能完全不记得 ECNU 的含义: 你也许会吧 ECNU 解释成国家电路联盟 (Electronic Circuit National Union) 或者欧洲中央诺维奇大学 (European Central Norwich University)。有时你可以根据文章上下文信息猜测缩写词含义, 但是有时你可能完全猜不出。

为了解决这个问题, Donald 发现了一本词典, 里面记录了所有短语及其缩写形式。Donald 想知道该词典里有多少无序的短语对 (即 (短语 1, 短语 2) 和 (短语 2, 短语 1) 被视为同一个短语对) 有着相同的缩写形式。

输入格式

输入的第一行为一个正整数 n ($1 \leq n \leq 5 \cdot 10^6$), 表示词典中短语数量。

接下来输入 n 行数据, 每行数据是由至少 2 个英文单词构成的短语。单词之间用 1 个空格隔开, 每行首尾不包含空格; 每个单词首字母大写, 其余字母小写; 每个单词最短包含 1 个字母, 最长包含 11 个字母。所有输入的短语所包含的单词总数不超过 10^6 。

词典中所有短语都是唯一的, 并且已按字典序排好序。排序过程中, 大写字母会首先被转为小写字母再进行比较。按照遵从 ASCII 码标准, 空格比任何字母都要小。

输出格式

输出为一个整数, 表示词典中有相同缩写形式的无序短语对的数量。

样例

standard input	
5	
East China Normal University	
Electronic Circuit National Union	
European Central Norwich University	
School Community Partnership Council	
Shanghai Collegiate Programming Contest	
standard output	
4	

standard input	standard output
3	0
C S L	
O X X	
Orz Orz	

提示

输入文件大小不超过 10M。请注意优化程序中 I/O 所占的时间, 因为其可能很耗时。

Problem G. Green-Red Tree

输入文件: 标准输入

输出文件: 标准输出

本题为交互题。

绿红树 (Green-Red Tree) 的专家唐纳德最近提出了一个新的博弈游戏, 该游戏得到了包括你在内的很多人的关注。

游戏的规则很简单。现假设你和唐纳德参与游戏, 游戏为回合制。在回合开始时, 唐纳德会给你 n , 作为这一轮中可用的顶点数, 对于任何两个不同的顶点, 都有一个连接它们的无向边可用。游戏从你的回合开始。在你的回合中, 你将选择两条不同的无向边, 你选择的边不能和之前选择的边重复。然后轮到唐纳德, 他将决定将其中一边涂成红色, 另外一边涂为绿色。在 $n-1$ 个回合后, 如果红色边缘形成一棵树, 绿色边缘也形成一棵树, 那么你就赢了; 否则, 你输了。

唐纳德可能对绿红树了解很多, 但对博弈没什么兴趣。所以他决定不在这场比赛中使用的任何最佳策略。换言之, 他会以相同的概率选择他可能的两个动作中的其中一个。即他会以 0.5 的概率绘制第一条边红色; 如果第一个边缘涂成红色, 第二个边缘是绿色。反之, 他会将第一条边绘制成绿色和第二条绘制成红色。

唐纳德每个回合的随机性有时会破坏别人赢得游戏的机会。

在唐纳德决定找到对付你这样做的策略之前, 你要找到每回合的最佳策略来赢得比赛。

交互

在第一行中, 给出一个正整数 t , 决定接下来的总回合数为 t 。

在每轮开始时, 你会被告知一个整数 n ($5 \leq n \leq 10^5$)。它还保证了, 所有测试用例中 n 的总和不超过 10^5 。

之后, 你将开始选边。每次, 你将选择两个不同的边 (u_1, v_1) 和 (u_2, v_2) ($u_1 \neq v_1, u_2 \neq v_2$), 注意它们之前没有被选中过, 然后将它们输出为格式为 $u_1 v_1 u_2 v_2$, 其间有空格。

然后唐纳德将决定。如果他在一行中输出 1, 则意味着他想要 (u_1, v_1) 绿色和 (u_2, v_2) 红色; 反之, 他将输出 0, 即第一个红色第二个绿色。

如果你违反了上述任何可能导致唐纳德不满意的规则, 他可能会提前离开游戏并导致你的程序挂起。因此, 评测机将可能返回 “Memory Limit Exceeded”, “Wrong Answer” 或 “Runtime Error” 中的任何一种。如果你的程序超出 CPU 时间限制, 将会得到 “Time Limit Exceeded” 它的含义和你在其他问题中看到的超时错误一致。

样例

标准输入	标准输出
1	
5	
	3 4 1 5
1	
	2 3 5 3
1	
	4 5 2 5
0	
	4 2 2 1
0	

附录

If you have participated in the warmup round, you might have seen the “Guess the Number” problem; in case you haven’t, we hope the following information may help.

Sometimes you can meet interactive problems on programming contests³.

In problems of this type, the input data given to your program may be not predetermined but is built specifically for your solution. Jury writes a special program – interactor, such that its output is transferred to the input of your solution, and the output of your program is sent to interactor’s input. In other words, your solution and the interactor exchange the data and decide what to print based on the “history of communication”.

When you write the solution for the interactive problem it is important to keep in mind that if you output some data it is possible that this data is first placed to some internal buffer and may be not directly transferred to the interactor. In order to avoid such situation you have to use special flush operation each time you output some data. There are these flush operations in standard libraries of almost all languages. For example, in C++ you may use `fflush(stdout)` or `cout << flush` (it depends on what do you use for output data – `scanf/printf` or `cout`). In Java you can use method `flush` for output stream, for example, `System.out.flush()`. In Python you can use `stdout.flush()`. In Pascal you can use `flush(output)`.

There are some features for interactive problems:

- Input/output in interactive problems works much slower than in usual problems —try to use `scanf/printf` instead of `cin/cout` in C++, `BufferedReader/PrintWriter` in Java and etc.
- Usually, manual testing of the solutions for interactive problems much more difficult, because the participant needed to be in the role of interactor during testing.
- Output `endl` in `cout` in C++ performs flush operation automatically.

³Copied from <https://codeforces.com/blog/entry/45307>, with modifications.

Problem H. Huashui Clock

输入文件: 标准输入

输出文件: 标准输出

对于那些不会说中文的人来说, huashui (划水) 这个词, 可能不是一个正常的单词。如果单是为了解决接下来的问题, 那么你不必知道这个词的含义, 但为了使说明更清楚, 这里的 huashui 指的是不工作的状态, 特别是指在工作时间不工作。

现在有人准备做个计划, 决定什么时候工作, 什么时候划水。因此他们发明了一款带有内置时钟的锁屏应用程序。这个时钟就像任何其他通常的时钟一样, 但是它每天有 h 小时, 每小时 m 分钟。应用程序中不显示秒数指示器, 你可以忽略秒单位。因此, 时间显示范围从 “0 : 0” 到 “ $h - 1 : m - 1$ ”。

现在的计划是: 如果分钟数大于或等于小时数, 这分钟就是 “划水时间”; 否则你会去上班。

你是一个勤奋的人, 你对这个时钟的想法感到不屑; 但唐纳德, 作为一个划水爱好者, 对这个应用程序很着迷, 非常想知道一天中他能划水多少时间。帮助他计算以获得一个气球。

输入格式

输入包括两行 h 和 m , ($2 \leq h, m \leq 10^9$), 用空格隔开。

输出格式

输出划水的时间除以一天的时间长度的分数, 用最简分数表示。

样例

标准输入	标准输出
2 2	3/4
2 7	13/14
7 2	3/14
13 11	6/13
100 33	17/100
100005 100009	50007/100009
1000000000 2	3/2000000000
2 999999999	1999999997/1999999998
914067307 998244353	541210700/998244353

提示

It would be interesting if you have got a “Wrong Answer” on this problem and did not do it on purpose.

Problem I. Induced Metric Space

输入文件: 标准输入

输出文件: 标准输出

在数学中, 度量空间是集合以及集合上的度量。度量是一个函数, 它定义了集合中任意两个成员之间的距离的概念, 通常称为点集。该指标满足一些简单的属性。在这个特定问题的背景下, 我们列举出如下几条:

- 一个点到自身的距离为 0,
- 两个点之间的距离为非负的,
- 从 A 到 B 的距离, 和从 B 到 A 的距离是相同的,
- 从 A 到 B 的 (直接) 距离, 小于或等于从 A , 经由任何第三点 C , 到 B 的距离, 即 $d(A, B) \leq d(A, C) + d(C, B)$ 。

现在有一个集合 S , 包括 n 个点。其中, 某些点之间的距离是已知的, 而其他点之间的距离是未知的。你的任务是填充所有空白, **不对已有的部分进行修改**, 使得任意两点之间建立距离关系, 这样就得到了一个度量空间。

输入格式

输入的第一行是一个正整数 t , 表示后面测试用例的数量。

然后是 t 个测试用例, 每个测试用例的首行是一个 n ($2 \leq n \leq 500$), 表示 S 中有 n 个点。接下来的 n 行的格式为: 第 i 行中的第 j 个整数表示 $d(i, j)$ ($-1 \leq d(i, j) \leq 1000$)。如果 $d(i, j)$ 为 -1 , 表示该值未知状态, 可以修改, 否则不应该改动。

输出格式

对于每个测试用例, 如果无法完成任务, 则在一行中输出 “NO”; 否则输出 “YES”。

如果它是 “YES”, 则应按照和输入相同的格式进行输出—第 i 行中的第 j 个整数表示 $d(i, j)$ ($0 \leq d(i, j) \leq 10^9$)。距离应该都是非负的, 并形成有效的度量空间。

如果有多多种解决方案, 输出任意一种即可。

样例

标准输入	标准输出
4	YES
3	0 3 3
0 3 3	3 0 3
3 0 3	3 3 0
3 3 0	YES
3	0 0 0
0 0 0	0 0 0
0 0 -1	0 0 0
0 -1 0	NO
3	YES
5 6 7	0 3 5
-1 -1 -1	3 0 3
-1 -1 -1	5 3 0
3	
-1 3 5	
-1 -1 3	
-1 -1 -1	

Problem J. Josephus Problem

输入文件: 标准输入

输出文件: 标准输出

在计算机科学和数学中, 约瑟夫斯问题 (或约瑟夫斯置换) 是一个理论问题, 与某个计数游戏有关。

人们站成一个圈, 按照 $1, 2, 3, \dots, n$ 顺时针标号, 从 1 开始, 沿顺时针方向开始。给出一个指定的数, 在跳过指定数量的人之后, 下一个人出局。整个过程由剩余的人重复, 从下一个人开始, 按照相同的方向, 跳过相同的人数, 直到最后只有一个人留下来存活。

典型的问题就是: 给定人数 n 和跳过的数量, 找出哪个人如此幸运可以存活下来。

唐纳德意识到, 对于像你这样才华横溢的程序员, 你可能已经看过这个问题一百万次; 所以他提出了一个更具挑战性的版本。为此, 先把上述问题一般化。

要做到这一点, 每个人都必须计算自然数。例如, 第一个人喊了一个 t 值; 则第二个人计数 $t+1$, 然后是 $t+2$ 等。当他们计算的自然数的是 k 的倍数时, 将出局。也许你已经注意到, 第一个人实际上可以选择一个好的 t 值, 以便某个人可以存活下来被释放。

唐纳德也注意到了这一点。他还发现, 你不能总是释放你想要的任何人。实际上, 在许多情况下, 这是不可能的。因此, 他稍微修改了游戏规则, 人们不仅在计数是 k ($2 \leq k \leq 9$) 的倍数时会出局, 而且在计数的十进制表示中包含 k 也会出局。例如, 19 不是 9 的倍数, 但它包含 9, 同样满足出局条件。

实际上, 可以严格证明上述的做法是可行的, 但是鉴于此题的篇幅限制, 不予给出, 这个问题就留给你了。

输入格式

输入的第一行包含整数 t ($1 \leq t \leq 10^4$), 表示 t 个测试用例。

然后对于每个测试用例, 有一行包含三个空格分隔的整数 n , k 和 x ($2 \leq n \leq 10^6$, $2 \leq k \leq 9$, $1 \leq x \leq n$), n 是一个圆圈中的人数, k 是每次跳过的人数, x 是你想要释放的人的标号。

保证所有测试用例的 n 的和不超过 10^6 。

输出格式

对于每个测试用例, 输出你选择释放 x 的 t ($1 \leq t \leq 10^{18}$)。如果有多个 t 可用, 您可以输出任何一个。

样例

标准输入	标准输出
40	1
3 2 3	2
3 2 2	3
3 2 3	4
3 2 2	5
3 2 3	6
3 2 2	7
3 2 3	8
3 2 2	9
3 2 3	10
3 2 2	1
10 3 5	2
10 3 4	3
10 3 3	4
10 3 5	5
10 3 4	6
10 3 3	7
10 3 10	8
10 3 9	9
10 3 8	10
10 3 1	11
10 3 10	12
10 3 9	13
10 3 5	14
10 3 9	15
10 3 8	1
7 9 7	2
7 9 6	3
7 9 5	4
7 9 4	5
7 9 3	6
7 9 2	7
7 9 1	8
7 9 7	9
7 9 6	10
7 9 7	11
7 9 6	12
7 9 5	13
7 9 4	14
7 9 3	15
7 9 2	

Problem K. Keyboard without Binary Digits

输入文件: 标准输入

输出文件: 标准输出

唐纳德最近从他的朋友那里收到了一台你在任何零售店都找不到的古董打字机。键入字母挺正常, 但键入数值或数字可能会比较痛苦。设计这个键盘的人很可能比较痛恨二进制, 因为“0”按钮被替换为“=”, “1”按钮被替换为“+”, 所以没有办法直接输入一个包含 0 或 1 的数字。除了有一个解决方法: 键入一些数字, 计算他们的和来得到你原本想要输入的数字。

有一个数字 n , 唐纳德真心想要在他的屏幕上 (或纸张, 或随便啥上) 显示, 但遗憾的是它至少有 0 或 1 中的一个。因此, 他要求你写一个程序, 帮助他找到几个和为 n 的正整数, 且数中均不包含数字 0 或 1。

输入格式

输入的第一行包含整数 t ($1 \leq t \leq 1\,000$), 表示数据组数。

接下来 t 行每行包含一组数据, 即一个正整数 n ($10 \leq n \leq 10^{100}$)。 n 没有前导零, 保证 n 中至少一位为 0 或 1。

输出格式

对于每组数据, 输出一个正整数 k 表示你将用 k 个数的加法来表示 n , 在下一行输出这 k 个数, a_1, a_2, \dots, a_k ($2 \leq a_i \leq n$ and $\sum_{i=1}^k a_i = n$), 使用空格分隔, a_i 中不能出现 0 或 1。

注意: k 必须尽可能小。如果仍有多解, 输出任意一个。

样例

标准输入	标准输出
3	2
911	42 869
19	3
300	6 7 6
	2
	258 42

Problem L. Liwa River

输入文件: 标准输入

输出文件: 标准输出



丽娃河是横穿于华东师范大学。杨柳垂堤，花繁叶茂，点缀着这条美丽的河流，俨然是校园最美丽的风景线。宋琳是中国著名的现代诗人，也是 ECNU 的前任教师，他在给朋友的一封信中写道：“如果这世上真有所谓天堂，那就是师大丽娃河边的一草一木，一沙一石。”。

二十世纪初，丽娃河流经的地方是上海的郊区，是江南水网的一部分。据说，该地被一些俄罗斯移民开辟为商业水边休闲度假胜地，主要服务于外国人，中国的中产阶级和资产阶级。

这个休闲度假胜地被称为“Rio Rita”，源自西班牙。这个名字来源于 1992 年的一部 Rio Rita 的美国音乐剧之后，主人以此为命名，这是一个关于墨西哥女孩丽塔的爱情故事，她居住在墨西哥边境的里奥格兰德河附近。由于对西班牙语和电影知之甚少，一些中国人误听为“里约”，并将其音译为“利瓦”，实际上意为“河流”。

休闲度假胜地被关闭后，大夏大学在该区建成，后来并入 ECNU。从此，丽娃河成为 ECNU 的代名词。这可能是你第一次来到这里，或者第二次，甚至是第三次；我非常肯定在这场比赛之后，你会想在丽娃河散步，感受空气中弥漫着的春天气息。同时，坐在河边的板凳上，与来自上海各个大学或高中，以及其他城市和省份的所有有才华的程序员（“大佬”）交朋友，这将是一次难忘的经历。

突然，你意识到为了实现这个美好的愿景，必须事先在这里运送一些长椅，使得朋友们都有个能坐的地方。具体来说，人们会报名参加 1, 2, 3, 4, 5 和 6 人组的比赛。现在举办方只提供 6 人座的长椅，每个队伍里的队员不想被拆散，坐到不同的椅子上，你的任务就是找到至少多少长椅，可以坐下所有人，同时没有一组的队员被拆散。你被要求在 5 个小时内解决这个问题，不然就太迟了，所有人都回家了。

输入格式

第一行包含 t ($1 \leq t \leq 1000$)，表示测试用例的个数。

每组数据包含六个以空格分隔的非负整数，六个数字分别表示 1 人，2 人，3 人，4 人，5 人和 6 人的组的数量。总人数总和大于 0 且不超过 10000，因为我们的竞赛区域不太可能容纳超过 1 万人。

输出格式

对每组数，输出一个正整数，表示所需的最少板凳数。

样例

标准输入	标准输出
6 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1	1 1 1 1 1 1
19 6 9 5 8 3 2 2 2 5 1 3 8 1 2 0 4 0 0 4 1 5 5 2 8 8 6 6 0 0 0 4 8 8 6 0 3 2 2 2 9 5 2 8 1 7 6 3 10 8 9 7 10 6 6 7 3 8 5 2 1 3 5 5 8 6 10 9 2 6 3 9 5 10 8 5 2 0 5 3 7 1 1 4 4 0 9 2 0 5 8 5 5 1 10 6 2 1 5 5 10 3 5 2 5 7 0 1 9 5 3 9 1 4 5	17 15 4 18 7 15 17 23 26 13 27 20 14 12 17 19 21 16 16