

任务混部

题目描述

公司创新实验室正在研究如何最小化资源成本，最大化资源利用率，请你设计算法帮他们解决一个任务混部问题：有 `taskNum` 项任务，每个任务有开始时间（`startTime`），结束时间（`endTime`），并行度(`parallelism`)三个属性，并行度是指这个任务运行时将会占用的服务器数量，一个服务器在每个时刻可以被任意任务使用但最多被一个任务占用，任务运行完会立即释放（结束时刻不占用）。任务混部问题是指给定一批任务，让这批任务由同一批服务器承载运行，请你计算完成这批任务混部最少需要多少服务器，从而最大化控制资源成本。

输入描述

第一行输入为 `taskNum`，表示有 `taskNum` 项任务
接下来 `taskNum` 行，每行三个整数，表示每个任务的开始时间（`startTime`），结束时间（`endTime`），并行度(`parallelism`)

输出描述

一个整数，表示最少需要的服务器数量

单词倒叙

题目描述

输入单行英文句子，里面包含英文字母，空格以及,.,?
三种标点符号，请将句子内每个单词进行倒序，并输出倒序后的语句

输入描述

输入字符串 `S`，`S` 的长度 $1 \leq N \leq 100$

输出描述

输出逆序后的字符串

获得完美走位

题目描述

在第一人称射击游戏中，玩家通过键盘的 `A`、`S`、`D`、`W` 四个按键控制游戏人物分别向左、向后、向右、向前进行移动，从而完成走位。假设玩家每按动一次键盘，游戏人物会向某个方向移动一步，如果玩家在操作一定次数的键盘并且各个方向的步数相同时，此时游戏人物必定会回到原点，则称此次走位为完美走位。

现给定玩家的走位（例如：`ASDA`），请通过更换其中一段连续走位的方式

使得原走位能够变成一个完美走位。其中待更换的连续走位可以是相同长度的任何走位。

请返回待更换的连续走位的最小可能长度。

若果原走位本身是一个完美走位，则返回 `0`。

输入描述

输入为由键盘字母表示的走位 s ，例如：ASDA

输出描述

输出为待更换的连续走位的最小可能长度

日志采集系统

题目描述

日志采集是运维系统的核心组件。日志是按行生成，每行记做一条，由采集系统分批上报。

如果上报太频繁，会对服务端造成压力；如果上报太晚，会降低用户的体验；如果一次上报的条数太多，会导致超时失败。

为此，项目组设计了如下的上报策略：

- 1、每成功上报一条日志，奖励 1 分
 - 2、每条日志每延迟上报 1 秒，扣 1 分
 - 3、积累日志达到 100 条，必须立即上报
- 给出日志序列，根据该规则，计算首次上报能获得的最多积分

输入描述

按时序产生的日志条数 $T_1, T_2 \dots T_n$ ，其中 $1 \leq n \leq 1000$, $0 \leq T_i \leq 100$

输出描述

首次上报最多能获得的积分

寻找链表的中间结点

题目描述

给定一个单链表 L ，请编写程序输出 L

中间结点保存的数据。如果有两个中间结点，则输出第二个中间结点保存的数据。

例如：给定 L 为 $1 \rightarrow 7 \rightarrow 5$ ，则输出应该为 7；给定 L 为 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ ，则输出应该为 3。

输入描述

每个输入包含 1 个测试用例。每个测试用例第 1

行给出链表首结点的地址、结点总个数正整数 N ($N \leq 105$)。结点的地址是 5

位非负整数，NULL 地址用

-1 表示。

接下来有 N 行，每行格式为：

Address Data Next

其中 Address 是结点地址，Data 是该结点保存的整数数据 ($0 \leq \text{Data} \leq 108$)，Next 是下一结点的地址。

输出描述

对每个测试用例，在一行中输出 L
中间结点保存的数据。如果有两个中间结点，则输出第二个中间结点保存的数据。

学校的位置

题目描述

为了解决新学期学生暴涨的问题，小乐村要建所新学校。考虑到学生上学安全问题，需要所有学生家到学校距离最短。

假设学校和所有的学生家，走在一条直线上。

请问，学校要建在什么位置，能使得学校到各个学生家的距离之和最短？

输入描述

输入的第一行是一个整数 N ($1 \leq N \leq 1000$)，表示有 N 户家庭。

输入的第二行是一个数组 ($0 \leq \leq 10000$)，表示每户家庭的位置，所有家庭的位置都不相同。

输出描述

输出一行，一个整数，表示你确定的学校位置。如有多个位置相同，则输出值最小的位置。

查找单入口空闲区域

题目描述

给定一个 $m \times n$ 的矩阵，由若干字符 'X' 和 'O' 构成，'X' 表示该处已被占据，'O' 表示该处空闲，请找到最大的单入口空闲区域。

解释：

空闲区域是由连通的 'O' 组成的区域，位于边界的 'O' 可以构成入口，单入口空闲区域即有且只有一个位于边界的 'O' 作为入口的由连通的 'O' 组成的区域。

如果两个元素在水平或垂直方向相邻，则称它们是“连通”的。

输入描述

第一行输入为两个数字，第一个数字为行数 m，第二个数字列数 n，两个数字以空格分隔， $1 \leq m, n \leq 200$ ，

剩余各行矩阵各行元素，元素为 'X' 或 'O'，各元素间以空格分隔。

输出描述

若有唯一符合要求的最大单入口空闲区域，输出三个数字，第一个数字为入口行坐标（范围为 $0 \sim \text{行数}-1$ ），第二个数字为入口列坐标（范围为 $0 \sim \text{列数}-1$ ），第三个数字为区域大小，三个数字以空格分隔；

若有多个符合要求的最大单入口空闲区域，输出一个数字，代表区域的大小；

若没有，输出 NUL。

核酸检测人员安排

题目描述

在系统、网络均正常的情况下组织核酸采样员和志愿者对人群进行核酸检测筛查。每名采样员的效率不同，采样效率为 N 人/小时。由于外界变化，采样员的效率会以 M 人/小时为粒度发生变化， M 为采样效率浮动粒度， $M=N10\%$ ，输入保证 $M10\%$ 的结果为整数。采样员

效率浮动规则：采样员需要一名志愿者协助组织才能发挥正常效率，在此基础上，每增加一名志愿者，效率提升 $1M$ ，最多提升 $3M$ ；如果没有志愿者协助组织，效率下降 $2M$ 。

怎么安排速度最快？求总最快检测效率（总检查效率为各采样人员效率值相加）。

输入描述

第一行：第一个值，采样员人数，取值范围 $[1,100]$ ；第二个值，志愿者人数，取值范围 $[1,500]$ ；

第二行：各采样员基准效率值(单位人/小时)，取值范围 $[60,600]$ ，保证序列中每项值计算 10% 为整数

输出描述

第一行：总最快检测效率(单位人/小时)

最小的调整次数

题目描述

有一个特异性的双端队列，该队列可以从头部或尾部添加数据，但是只能从头部移出数据。

小 A 依次执行 $2n$ 个指令往队列中添加数据和移出数据。其中 n 个指令是添加数据（可能从头部添加、也可能从尾部添加

），依次添加 1 到 n ； n 个指令是移出数据。现在要求移除数据的顺序为 1 到 n 。为了满足最后输出的要求，小 A 可以在任何时候调整队列中数据的顺序。

请问 小 A 最少需要调整几次才能够满足移除数据的顺序正好是 1 到 n ；

输入描述

第一行一个整数 n ，表示数据范围。

接下来有 $2n$ 行，其中有 n 行为添加数据：指令“head add x”表示从头部添加数据 x ，“tail add x”表示从尾部添加数据 x ；另外 n

行为移出数据指令，指令为“remove”的形式，表示移出 1 个数据； $1 \leq n \leq 3 * 10^5$ 。

所有的数据均合法。

输出描述

一个整数，表示 小 A 要调整的最小次数。

优秀学员统计

题目描述

公司某部门软件教导团正在组织新员工每日打卡学习活动，他们开展这项学习活动已经一个月了，所以想统计下这个月优秀的打卡员工。每个员工会对应一个 **id**，每天的打卡记录记录当天打卡员工的 **id** 集合，一共 **30** 天。

请你实现代码帮助统计出打卡次数 **top5** 的员工。假如打卡次数相同，将较早参与打卡的员工排在前面，如果开始参与打卡的时间还是一样，将 **id** 较小的员工排在前面。

注：不考虑并列的情况，按规则返回前 **5** 名员工的 **id** 即可，如果当月打卡的员工少于 **5** 个，按规则排序返回所有有打卡记录的员工 **id**。

输入描述

第一行输入为新员工数量 **N**，表示新员工编号 **id** 为 **0** 到 **N-1**，**N** 的范围为[1,100]

第二行输入为 **30** 个整数，表示每天打卡的员工数量，每天至少有 **1** 名员工打卡之后 **30** 行为每天打卡的员工 **id** 集合，**id** 不会重复

输出描述

按顺序输出打卡 **top5** 员工的 **id**，用空格隔开

货币单位换算

题目描述

记账本上记录了若干条多国货币金额，需要转换成人民币分（**fen**），汇总后输出。

每行记录一条金额，金额带有货币单位，格式为数字+单位，可能是单独元，或者单独分，或者元与分的组合。

要求将这些货币全部换算成人民币分（**fen**）后进行汇总，汇总结果仅保留整数，小数部分舍弃。

元和分的换算关系都是 **1:100**，如下：

1CNY=100fen（

1 元=100 分）

1HKD=100cents（

1 港元=100 港分）

1JPY=100sen（

1 日元=100 仙）

1EUR=100eurocents（

1 欧元=100 欧分）

1GBP=100pence（

1 英镑=100 便士）

汇率如下表

CNY

|

JPY

|

HKD
|
EUR
|
GBP
---|---|---|---|---
100
|
1825
|
123|
14
|
12
即 100CNY=1825JPY=123HKD=14EUR=12GBP

输入描述

第一行输入为 N ， N 表示记录数。 $0 < N < 100$
之后 N 行，每行表示一条货币记录，且该行只会是一种货币。

输出描述

将每行货币转换成人民币分（fen）后汇总求和，只保留整数部分。
输出格式只有整数数字，不带小数，不带单位。

简单的自动曝光

题目描述

一个图像有 n 个像素点，存储在一个长度为 n 的数组 `img` 里，每个像素点的取值范围 $[0, 255]$ 的正整数。
请你给图像每个像素点值加上一个整数 k （可以是负数），得到新图 `newImg`，使得新图 `newImg` 的所有像素平均值最接近中位值 128。
请输出这个整数 k 。

输入描述

n 个整数，中间用空格分开
例如：
0 0 0 0
4 个数值，中间用空格分开

输出描述

一个整数 k

查找重复代码

题目描述

小明负责维护项目下的代码，需要查找出重复代码，用以支撑后续的代码优化，请你帮助小明找出重复的代码，。

重复代码查找方法：以字符串形式给定两行代码（字符串长度 $1 < \text{length} \leq 100$ ，由英文字母、数字和空格组成），找出两行代码中的最长公共子串。

注： 如果不存在公共子串，返回空字符串

输入描述

输入的参数 `text1`, `text2` 分别表示两行代码

输出描述

输出任一最长公共子串

工作安排

题目描述

小明每周上班都会拿到自己的工作清单，工作清单内包含 n 项工作，每项工作都有对应的耗时时长（单位 h ）和报酬，工作的总报酬为所有已完成工作的报酬之和。那么请你帮小明安排一下工作，保证小明在指定的工作时间内工作收入最大化。

输入描述

输入的第一行为两个正整数 T, n 。 T 代表工作时长（单位 h ， $0 < T < 100000$ ）， n 代表工作数量($1 < n \leq 3000$)。

接下来是 n 行，每行包含两个整数 t, w 。 t 代表该项工作消耗的时长（单位 h ， $t > 0$ ）， w 代表该项工作的报酬。

输出描述

输出小明指定工作时长内工作可获得的最大报酬。

新员工座位安排系统

题目描述

工位由序列 $F_1, F_2 \dots F_n$ 组成， F_i 值为 0、1 或 2。其中 0 代表空置，1 代表有人，2 代表障碍物。

1、某一空位的友好度为左右连续老员工数之和

2、为方便新员工学习求助，优先安排友好度高的空位

给出工位序列，求所有空位中友好度的最大值。

输入描述

第一行为工位序列： $F_1, F_2 \dots F_n$ 组成， $1 \leq n \leq 100000$ ， F_i 值为 0、1 或 2。其中 0 代表空置，1 代表有人，2 代表障碍物

其中 0 代表空置，1 代表有人，2 代表障碍物。

输出描述

所有空位中友好度的最大值。如果没有空位，返回 0。

找出通过车辆最多颜色

题目描述

在一个狭小的路口，每秒只能通过一辆车，假如车辆的颜色只有 3 种，找出 N 秒内经过的最多颜色的车辆数量

三种颜色编号为 0,1,2

输入描述

第一行输入的是通过的车辆颜色信息

[0,1,1,2] 代表 4 秒钟通过的车辆颜色分别是 0,1,1,2

第二行输入的是统计时间窗，整型，单位为秒

输出描述

输出指定时间窗内经过的最多颜色的车辆数量

找数字

题目描述

给一个二维数组 nums，对于每一个元素 num[i]，找出距离最近的且值相等的元素，输出横纵坐标差值的绝对值之和，如果没有等值元素，则输出-1。

例如：

输入数组 nums 为

0 3 5 4 2

2 5 7 8 3

2 5 4 2 4

对于 num[0][0] = 0，不存在相等的值。

对于 num[0][1] = 3，存在一个相等的值，最近的坐标为 num[1][4]，最小距离为 4。

对于 num[0][2] = 5，存在两个相等的值，最近的坐标为 num[1][1]，故最小距离为 2。

...

对于 num[1][1] = 5，存在两个相等的值，最近的坐标为 num[2][1]，故最小距离为 1。

...

故输出为

-1 4 2 3 3

1 1 -1 -1 4

1 1 2 3 2

输入描述

输入第一行为二维数组的行

输入第二行为二维数组的列

输入的数字以空格隔开。

输出描述

数组形式返回所有坐标值。

字符串重新排序

题目描述

给定一个字符串 s ， s 包含以空格分隔的若干个单词，请对 s 进行如下处理后输出：

1、单词内部调整：对每个单词字母重新按字典序排序；

2、单词间顺序调整：

1) 统计每个单词出现的次数，并按次数降序排列；

2) 次数相同时，按单词长度升序排列；

3) 次数和单词长度均相同时，按字典序升序排列。

请输出处理后的字符串，每个单词以一个空格分隔。

输入描述

一行字符串，每个字符取值范围： $[a-zA-Z0-9]$ 以及空格，字符串长度范围： $[1, 1000]$

输出描述

重新排序后的字符串，每个单词间隔 1 个空格，且首尾无空格

基站维修工程师

题目描述

小王是一名基站维护工程师，负责某区域的基站维护。

某地方有 n 个基站（ $1 < n < 10$ ），已知各基站之间的距离 s （ $0 < s < 500$ ），并且基站 x 到基站 y 的距离，与基站 y 到基站 x 的距离并不一定会相同。

小王从基站 1 出发，途径每个基站 1 次，然后返回基站 1，需要请你为他选择一条距离最短的路线。

输入描述

站点数 n 和各站点之间的距离（均为整数）。如：

3 {站点数}

0 2 1 {站点 1 到各站点的路程}

1 0 2 {站点 2 到各站点的路程}

2 1 0 {站点 3 到各站点的路程}

输出描述

最短路程的数值

AI 处理器组合

题目描述

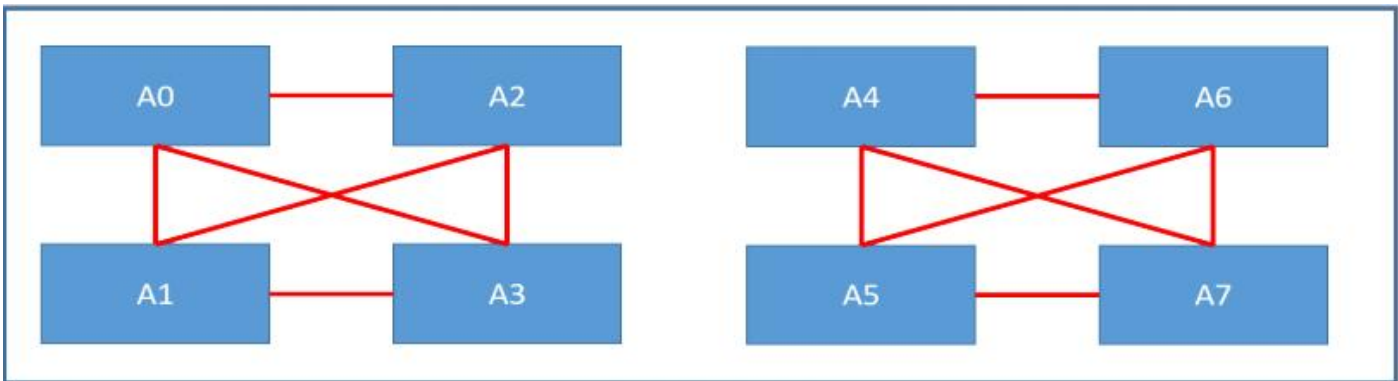
某公司研发了一款高性能 AI 处理器。每台物理设备具备 8 颗 AI 处理器，编号分别为 0、1、2、3、4、5、6、7。编号 0-3 的处理器处于同一个链路中，编号 4-7 的处理器处于另外一个链路中，不通链路中的处理器不能通信，如下图所示。现给定服务器可用的处理器编号数组 `array`，以及任务申请的处理器数量 `num`，找出符合下列亲和性调度原则的芯片组合。如果不存在符合要求的组合，则返回空列表。

亲和性调度原则：

- \- 如果申请处理器个数为 1，则选择同一链路，剩余可用的处理器数量为 1 个的最佳，其次是剩余 3 个的为次佳，然后是剩余 2 个，最后是剩余 4 个。
- \- 如果申请处理器个数为 2，则选择同一链路剩余可用的处理器数量 2 个的为最佳，其次是剩余 4 个，最后是剩余 3 个。
- \- 如果申请处理器个数为 4，则必须选择同一链路剩余可用的处理器数量为 4 个。
- \- 如果申请处理器个数为 8，则申请节点所有 8 个处理器。

提示：

- 1\、任务申请的处理器数量只能是 1、2、4、8
- 2\、编号 0-3 的处理器处于一个链路，编号 4-7 的处理器处于另外一个链路。
- 3\、处理器编号唯一，且不存在相同编号处理器



输入描述

输入包含可用的处理器编号数组 `array`，以及任务申请的处理器数量 `num` 两个部分。
第一行为 `array`，第二行为 `num`。例如：

[0, 1, 4, 5, 6, 7]

1
表示当前编号为 0、1、4、5、6、7 的处理器可用。任务申请 1 个处理器。

0<= array.length <= 8

0<= array[i] <= 7

num in [1, 2, 4, 8]

输出描述

输出为组合列表，当 `array=[0, 1, 4, 5, 6, 7]`，`num=1` 时，输出为[[0], [1]]

整理扑克牌

题目描述

给定一组数字，表示扑克牌的牌面数字，忽略扑克牌的花色，请按如下规则对这一组扑克牌进行整理：

步骤 1、对扑克牌进行分组，形成组合牌，规则如下：

1. 当牌面数字相同张数 **大于等于 4 时**，组合牌为“炸弹”；
2. 3 张相同牌面数字 + 2 张相同牌面数字，且 3 张牌与 2 张牌不相同，组合牌为“葫芦”；
3. 3 张相同牌面数字，组合牌为“三张”；
4. 2 张相同牌面数字，组合牌为“对子”；
5. 剩余没有相同的牌，则为“单张”；

步骤 2、对上述组合牌进行由大到小排列，规则如下：

1. 不同类型组合牌之间由大到小排列规则：“炸弹” > “葫芦” > “三张” > “对子” > “单张”；
2. 相同类型组合牌之间，除“葫芦”外，按组合牌全部牌面数字加总由大到小排列；
3. “葫芦”则先按 3 张相同牌面数字加总由大到小排列，3 张相同牌面数字加总相同时，再按另外 2 张牌面数字加总由大到小排列；
4. 由于“葫芦”

>“三张”，因此如果能形成更大的组合牌，也可以将“三张”拆分为 2 张和 1 张，其中

的 2 张可以和其它“三张”重新组合成“葫芦”，剩下的 1 张为“单张”

步骤 3、当存在多个可能组合方案时，按如下规则排序取最大的一个组合方案：

1. 依次对组合方案中的组合牌进行大小比较，规则同上；
2. 当组合方案 A 中的第 n 个组合牌大于组合方案 B 中的第 n 个组合牌时，组合方案 A 大于组合方案 B；

输入描述

第一行为空格分隔的 N 个正整数，每个整数取值范围[1,13]，N 的取值范围[1,1000]

输出描述

经重新排列后的扑克牌数字列表，每个数字以空格分隔

MVP 争夺战

题目描述

在星球争霸篮球赛对抗赛中，强大的宇宙战队，希望每个人都能拿到 MVP。

MVP 的条件是，单场最高得分获得者，可以并列，所以宇宙战队决定在比赛中，尽可能让更多的队员上场，且让所有有得分的队员得分都相同。

然而比赛过程中的每一分钟的得分都只能由某一个人包揽。

输入描述

输入第一行为一个数字 t ，表示有得分的分钟数（ $1 \leq t \leq 50$ ），第二行为 t 个数字，代表每一分钟的得分 p （ $1 \leq p \leq 50$ ）

输出描述

输出有得分的队员都是 MVP 时最少的 MVP 得分。

对称字符串

题目描述

对称就是最大的美学，现有一道关于对称字符串的美学。已知：

第 1 个字符串：R

第 2 个字符串：BR

第 3 个字符串：RBBR

第 4 个字符串：BRRBRBBR

第 5 个字符串：RBBRBRRBBRRBRBBR

相信你已经发现规律了，没错！就是第 i 个字符串 = 第 $i-1$ 号字符串的取反 + 第 $i-1$ 号字符串；取反(R->B, B->R)；

现在告诉你 n 和 k ，让你求得第 n 个字符串的第 k 个字符是多少。(k 的编号从 0 开始)

输入描述

第一行输入一个 T ，表示有 T 组用例；

接下来输入 T 行，每行输入两个数字，表示 n, k

$1 \leq T \leq 100$;

$1 \leq n \leq 64$;

$0 \leq k < 2^{(n-1)}$;

输出描述

输出 T 行表示答案；

输出 "blue" 表示字符是 B；

输出 "red" 表示字符是 R；

任务总执行时长

题目描述

任务编排服务负责对任务进行组合调度。参与编排的任务有两种类型，其中一种执行时长为 $taskA$ ，另一种执行时长为 $taskB$ 。任务一旦开始执行不能被打断，且任务可连续执行。服务每次可以编排 num 个任务。请编写一个方法，生成每次编排后的任务所有可能的总执行时长。

输入描述

第 1 行输入分别为第 1 种任务执行时长 $taskA$ ，第 2 种任务执行时长 $taskB$ ，这次要编排的任务个数 num ，以逗号分隔。

输出描述

数组形式返回所有总执行时时长，需要按从小到大排列。

最左侧冗余覆盖子串

题目描述

给定 2 个字符串 s_1 和 s_2 和正整数 k ，其中 s_1 长度为 n_1 ， s_2 长度为 n_2 ，在 s_2 中选一个子串，满足：

- * 该子串长度为 n_1+k
- * 该子串包含 s_1 中全部字母
- * 该子串每个字母的出现次数不小于 s_1 中对应的字母

我们称 s_2 以长度 k 冗余覆盖 s_1 。给定 s_1 、 s_2 和 k ，求最左侧的 s_2 以长度 k 冗余覆盖 s_1 的子串的首个元素的下标，如果没有返回-1

举例：

$s_1=ab$

$s_2=aabcd$

$k=1$

则子串 aab 和 abc 均满足此条件，由于 aab 在 abc 的左侧， aab 的第一个元素下标为 0，因此输出 0

输入描述

输入三行，第一行为 s_1 ，第二行为 s_2 ，第三行为 k

s_1 和 s_2 只包含小写字母

输出描述

最左侧的 s_2 以长度 k 冗余覆盖 s_1 的子串首个元素的下标，如果没有返回-1

获取最大软件版本号

题目描述

Maven 版本号定义，<主版本>.<次版本>.<增量版本>-<里程碑版本>，举例 **3.1.4-beta**

其中，主版本和次版本都是必须的，主版本，次版本，增量版本由多位数字组成，可能包含前导零，里程碑版本由字符串组成。

<主版本>.<次版本>.<增量版本>：基于数字比较

里程碑版本：基于字符串比较,采用字典序

比较版本号时，按从左到右的顺序依次比较。基于数字比较，只需比较忽略任何前导零后的整数值。

输入 2 个版本号，输出最大版本号

输入描述

输入 2 个版本号，换行分割，每个版本的最大长度小于 50

输出描述

版本号相同时输出第一个输入版本号

网上商城优惠活动（一）

题目描述

【背景】

某网上商城举办优惠活动，发布了满减、打折、无门槛 3 种优惠券，分别为：

1. 每满 100 元优惠 10 元，无使用数限制，如 100~199 元可以使用 1 张减 10 元，200~299 可使用 2 张减 20 元，以此类推；
2. 92 折券，1 次限使用 1 张，如 100 元，则优惠后为 92 元；
3. 无门槛 5 元优惠券，无使用数限制，直接减 5 元。

【优惠券使用限制】

每次最多使用 2 种优惠券，2 种优惠可以叠加（优惠叠加时以优惠后的价格计算），以购物 200 元为例，可以先用 92 折券优惠到 184 元，再用 1 张满减券优惠 10 元，最终价格是 174 元，也可以用满减券 2 张优惠 20 元为 180 元，再使用 92 折券优惠到 165（165.6 向下取整）元，不同使用顺序的优惠价格不同，以最优惠价格为准。在一次购物中，同一类型优惠券使用多张时必须一次性使用，不能分多次拆开穿插使用（不允许先使用 1 张满减券，再用打折券，再使用一张满减券）。

【问题】

请设计实现一种解决方法，帮助购物者以最少的优惠券获得最优的优惠价格。优惠后价格越低越好，同等优惠价格，使用的优惠券越少越好，可以允许某次购物不使用优惠券。

【约定】

优惠活动每人只能参加一次，每个人的优惠券种类和数量是一样的。

输入描述

第一行：每个人拥有的优惠券数量（数量取值范围为[0, 10]），按满减、打折、无门槛的顺序输入。

第二行：表示购物的人数 n （ $1 \leq n \leq 10000$ ）。

最后 n 行：每一行表示某个人优惠前的购物总价格（价格取值范围(0, 1000]，都为整数）。

约定：输入都是符合题目设定的要求的。

输出描述

每行输出每个人每次购物优惠后的最低价格以及使用的优惠券总数量，每行的输出顺序和输入的顺序保持一致。

异常的打卡记录

题目描述

考勤记录是分析和考核职工工作时间利用情况的原始依据，也是计算职工工资的原始依据，为了正确地计算职工工资和监督工资基金使用情况，公司决定对员工的手机打卡记录进行异常排查。

如果出现以下两种情况，则认为打卡异常：

- 1、实际设备号与注册设备号不一样
- 2、或者，同一个员工的两个打卡记录的时间小于 60 分钟并且打卡距离超过 5km。

给定打卡记录的字符串数组 `clockRecords`（每个打卡记录组成为：工号;时间（分钟）;打卡距离（km）;实际设备号;注册设备号），返回其中异常的打卡记录（按输入顺序输出）。

输入描述

第一行输入为 N ，表示打卡记录数；
之后的 N 行为打卡记录，每一行为一条打卡记录。

例如：

```
2
100000,10,1,ABCD,ABCD
100000,50,10,ABCD,ABCD
```

输出描述

输出为异常的打卡记录，例如：100000,10,1,ABCD,ABCD;100000,50,10,ABCD,ABCD

服务中心的最佳位置

题目描述

一家快递公司希望在一条街道建立新的服务中心。公司统计了该街道中所有区域在地图上的位置，并希望能够以此为依据为新的服务中心选址：使服务中心到所有区域的距离的总和最小。

给你一个数组 `positions`，其中 `positions[i] = [left, right]` 表示第 i 个区域在街道上的位置，其中 `left`

代表区域的左侧的起点，`right` 表示区域的右侧终点，设择服务中心的位置为 `location`。

- * 如果第 i 个区域的右侧起点 `right` 满足 $right < location$ ，则第 i 个区域到服务中心的距离为 $location - right$ ；
- * 如果第 i 个区域的左侧起点 `left` 满足 $left > location$ ，则第 i 个区域到服务中心的距离为 $left - location$ ；
- * 如果第 i 个区域的两侧 `left, right` 满足 $left \leq location \leq right$ ，则第 i 个区域到服务中心的距离为 0 ；

选择最佳的服务中心的位置为 `location`

，请返回最佳的服务中心位置到所有区域的距离总和的最小值。

输入描述

先输入区域数组 `positions` 的长度 n ($1 \leq n \leq 10^5$)
接下来 n 行每行输入成对的 `left` 和 `right` 值，以空格隔开
 $-10^9 < left \leq 10^9$
 $-10^9 < right \leq 10^9$

输出描述

输出为 `location`

不爱施肥的小布

题目描述

某农场主管理了一大片果园，`fields[i]`
表示不同果林的面积，单位: (m^2)，现在要为所有的果林施肥且必须在 n

天之内完成，否则影响收成。小布是果林的工作人员，他每次选择一片果林进行施肥，且一片果林施肥完后当天不再进行施肥作业。假设施肥机的能效为 k ，单位: (m^2/day) ，请问至少租赁能效 k 为多少的施肥机才能确保不影响收成？如果无法完成施肥任务，则返回 -1。

输入描述

第一行输入为 m 和 n ， m 表示 `fields` 中的元素个数， n 表示施肥任务必须在 n 天内（含 n 天）完成；

第二行输入为 `fields`，`fields[i]` 表示果林 i 的面积，单位: (m^2)

输出描述

对于每组数据，输出最小施肥机的能效 k ，无多余空格。

微服务的集成测试

题目描述

现在有 n 个容器服务，服务的启动可能有一定的依赖性（有些服务启动没有依赖），其次**服务自身启动加载会消耗一些时间**。

给你一个 $n \times n$ 的二维矩阵 `useTime`，其中 `useTime[i][i]=10` 表示服务 i

自身启动加载需要消耗 10s，`useTime[i][j]=1` 表示服务 i 启动依赖服务 j

启动完成，`useTime[i][k]=0`，表示服务 i

启动不依赖服务 k

其实 $0 \leq i, j, k <$

n 。服务之间启动没有循环依赖（不会出现环），若想对任意一个服务 i 进行集成测试（服务 i 自身也需要加载），求最少需要等待多少时间。

输入描述

第一行输入服务总量 n ，之后的 n 行表示服务启动的依赖关系以及自身启动加载耗时

最后输入 k 表示计算需要等待多少时间后可以对服务 k 进行集成测试

其中 $1 \leq k \leq n$ ， $1 \leq n \leq 100$

输出描述

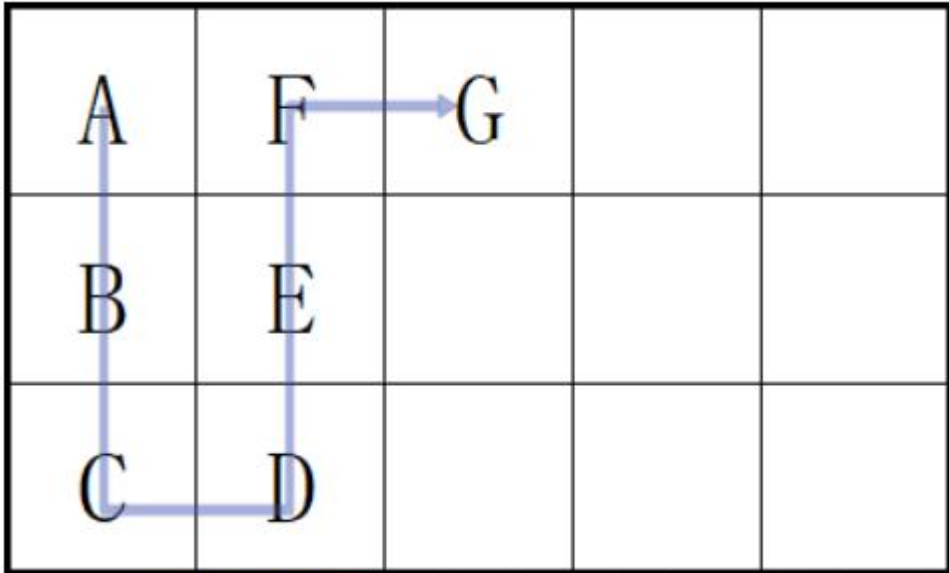
最少需要等待多少时间(s)后可以对服务 k 进行集成测试

箱子之形摆放

题目描述

有一批箱子（形式为字符串，设为 `str`），要求将这批箱子按从上到下以之形的顺序摆放在宽度为 n 的空地，请输出箱子的摆放结果。

例如：箱子为 `ABCDEFGH`，空地宽度为 3，摆放结果如图：



则输出结果为：

AFG
BE
CD

输入描述

输入一行字符串，通过空格分割，前面部分为字母或数字组成的字符串 **str**，表示箱子；后面部分为一个数字 **n**，表示空地的宽度。例如：

ABCDEFG 3

输出描述

箱子摆放结果，如题目示例所示

AFG
BE
CD

字符串解密

题目描述

给定两个字符串 **string1** 和 **string2**。
string1 是一个被加扰的字符串。**string1** 由小写英文字母（'a'~'z'）和数字字符（'0'~'9'）组成，而加扰字符串由'0'~'9'、'a'~'f'组成。**string1** 里面可能包含 0 个或多个加扰子串，剩下可能有 0 个或多个有效子串，这些有效子串被加扰子串隔开。
string2 是一个参考字符串，仅由小写英文字母（'a'~'z'）组成。
你需要在 **string1** 字符串里找到一个有效子串，这个有效子串要同时满足下面两个条件：
(
1) 这个有效子串里不同字母的数量不超过且最接近于 **string2** 里不同字母的数量，即小于或等于 **string2** 里不同字母的数量的同时且最大。
(
2) 这个有效子串是满足条件 (
1) 里的所有子串（如果有多个的话）里字典序最大的一

个。

如果没有找到合适条件的子串的话，请输出"Not Found"

示例：

输入字符串 `string1` 为"thisisanewday111forme"，输入字符串 `string2` 为"good"。`string1` 里有效子串和加扰子串分割后可表示为："thisis"+"a"+"n"+"e"+"w"+"da"+"y"+"111f"+"orm"+"e"，去除加扰子串（"a"、"e"、"da"、"111f"、"e"）后的有效子串候选为("thisis", "n", "w", "y", "orm")。输入字符串 `string2` 里不同字母的数量为 3（'g'、'o'、'd'），从有效子串候选里可以找出"orm"满足要求，其不同字母的数量为 3，最接近于 `string2` 不同字母的数量。

输入描述

`input_string1`

`input_string2`

说明：输入为两个字符串，第 1 行是题目里的 `string1`（被加扰的字符串），第 2 行是题目里的 `string2`（参考字符串）

输出描述

`output_string`

说明：输出为一个字符串（有效字符串）

租车骑绿道

题目描述

部门组织绿道骑行团建活动。租用公共双人自行车骑行，每辆自行车最多坐两人、做大载重 `M`。

给出部门每个人的体重，请问最多需要租用多少双人自行车。

输入描述

第一行两个数字 `m`、`n`，自行车限重 `m`，代表部门总人数 `n`。

第二行，`n` 个数字，代表每个人的体重。体重都小于等于自行车限重 `m`。

$0 < m \leq 200$

$0 < n \leq 1000000$

输出描述

最小需要的双人自行车数量。

打印文件

题目描述

有 5 台打印机打印文件，每台打印机有自己的待打印队列。因为打印的文件内容有轻重缓急之分，所以队列中的文件有 1~10 不同的优先级，其中数字越大优先级越高。打印机会从自己的待打印队列中选择优先级最高的文件来打印。如果存在两个优先级一样的文件，则选择最早进入队列的那个文件。

现在请你来模拟这 5 台打印机的打印过程。

输入描述

每个输入包含 1 个测试用例，每个测试用例第 1 行给出发生事件的数量 N ($0 < N < 1000$)。

接下来有 N 行，分别表示发生的事件。

共有如下两种事件：

1\ "IN P NUM", 表示有一个拥有优先级 NUM 的文件放到了打印机 P 的待打印队列中。 ($0 < P \leq 5, 0 < \text{NUM} \leq 10$);

2\ "OUT P", 表示打印机 P 进行了一次文件打印，同时该文件从待打印队列中取出。 ($0 < P \leq 5$)。

输出描述

对于每个测试用例，每次"OUT

P"事件，请在一行中输出文件的编号。如果此时没有文件可以打印，请输出"NULL"。

文件的编号定义为："IN P NUM"事件发生第 X 次，此处待打印文件的编号为 X 。编号从 1 开始。

开心消消乐

题目描述

给定一个 N 行 M 列的二维矩阵，矩阵中每个位置的数字取值为 0 或 1。矩阵示例如：

1 1 0 0

0 0 0 1

0 0 1 1

1 1 1 1

现需要将矩阵中所有的 1 进行反转为 0，规则如下：

1) 当点击一个 1 时，该 1 变被反转为 0，同时相邻的上、下、左、右，以及左上、左下、右上、右下 8 个方向的 1（如果存在 1）均会自动反转为 0；

2) 进一步地，一个位置上的 1 被反转为 0 时，与其相邻的 8 个方向的 1（如果存在 1）均会自动反转为 0；

按照上述规则示例中的矩阵只最少需要点击 2 次后，所有值均为 0。请问，给定一个矩阵，最少需要点击几次后，所有数字均为 0？

输入描述

第一行为两个整数，分别表示矩阵的行数 N 和列数 M ，取值范围均为 $[1, 100]$

接下来 N 行表示矩阵的初始值，每行均为 M 个数，取值范围 $[0, 1]$

输出描述

输出一个整数，表示最少需要点击的次数

最多提取子串数目

题目描述

给定由 $[a-z]$

26 个英文小写字母组成的字符串 A 和 B，其中 A 中可能存在重复字母，B 中不会存在重复字母。现从字符串 A 中按规则挑选一些字母，可以组成字符串 B。

挑选规则如下：

- 1) 同一个位置的字母只能被挑选一次
- 2) 被挑选字母的相对先后顺序不能改变

求最多可以同时从 A 中挑选多少组能组成 B 的字符串

输入描述

输入为 2 行，第 1 行输入字符串 A，第 2 行输入字符串 B，行首行尾无多余空格

其中 A、B 均由[a-z] 26 个英文小写字母组成

$0 < A.length < 100$ ，A 中可能包含重复字母

$0 < B.length < 10$ ，B 中不会出现重复字母

输出描述

输出 1 行，包含 1 个数字，表示最多可以同时从 A 中挑选多少组能组成 B 的字符串

行末无多余空格

最多等和不相交连续子序列

题目描述

给定一个数组，我们称其中连续的元素为连续子序列，称这些元素的和为连续子序列的和。数组中可能存在几组连续子序列，组内的连续子序列互不相交且有相同的和。求一组连续子序列，组内子序列的数目最多。输出这个数目。

输入描述

第一行输入为数组长度 N， $1 \leq N \leq 10^3$ 。

第二行为 N 个用空格分开的整数 C_i ， $-10^5 \leq C_i \leq 10^5$ 。

输出描述

第一行是一个整数 M，表示满足要求的最多的组内子序列的数目。

相同数字的积木游戏 1

题目描述

小华和小薇一起通过玩积木游戏学习数学。

他们有很多积木，每个积木块上都有一个数字，积木块上的数字可能相同。

小华随机拿一些积木挨着排成一排，请小薇找到这排积木中数字相同且所处位置最远的 2 块积木块，计算他们的距离。

小薇请你帮忙替解决这个问题。

输入描述

第一行输入为 N，表示小华排成一排的积木总数。

接下来 N 行每行一个数字，表示小华排成一排的积木上数字。

输出描述

相同数字的积木的位置最远距离；
如果所有积木数字都不相同，请返回-1.

寻找关键钥匙

题目描述

小强正在参加《密室逃生》游戏，当前关卡要求找到符合给定

密码 K （升序的不重复小写字母组成） 的箱子，并给出箱子编号，箱子编号为 $1 \sim N$ 。

每个箱子中都有一个 字符串 s

，字符串由大写字母，小写字母，数字，标点符号，空格组成，需要在这些字符串中找出所有的字母，忽略大小写后排列出对应的密码串，并返回匹配密码的箱子序号

提示：

满足条件的箱子不超过 1 个

输入描述

第一行为 key 的字符串，第二行为箱子 $boxes$ ，为数组样式，以空格分隔

箱子 N 数量满足 $1 \leq N \leq 10000$,

s 长度满足 $0 \leq s.length \leq 50$,

密码为仅包含小写字母的升序字符串，且不存在重复字母，

密码 K 长度 $K.length$ ， $1 \leq K.length \leq 26$

输出描述

返回对应箱子编号

如不存在符合要求的密码箱，则返回-1

查找充电设备组合

题目描述

某个充电站，可提供 n 个充电设备，每个充电设备均有对应的输出功率。任意个充电设备组合的输出功率总和，均构成功率集合 P 的 1 个元素。功率集合 P 的最优元素，表示最接近充电站最大输出功率 p_{max} 的元素。

输入描述

输入为 3 行：

第 1 行为充电设备个数 n 。

第 2 行为每个充电设备的输出功率。

第 3 行为充电站最大输出功率 p_{max} 。

输出描述

猜字谜

题目描述

小王设计了一个简单的猜字谜游戏，游戏的谜面是一个错误的单词，比如 `nesw`，玩家需要猜出谜底库中正确的单词。猜中的要求如下：

对于某个谜面和谜底单词，满足下面任一条件都表示猜中：

1) 变换顺序以后一样的，比如通过变换 `w` 和 `e` 的顺序，“`nwes`”跟“`news`”是可以完全对应的；

2) 字母去重以后是一样的，比如“`woood`”和“`wood`”是一样的，它们去重后都是“`wod`”

请你写一个程序帮忙在谜底库中找到正确的谜底。谜面是多个单词，都需要找到对应的谜底，如果找不到的话，返回“`not found`”

输入描述

- 1、谜面单词列表，以“,”分隔
- 2、谜底库单词列表，以“,”分隔

输出描述

匹配到的正确单词列表，以“,”分隔
如果找不到，返回“`not found`”

分界线

题目描述

电视剧《分界线》里面有一个片段，男主为了向警察透露案件细节，且不暴露自己，于是将报刊上的字剪切下来，剪拼成匿名信。

现在有一名举报人，希望借鉴这种手段，使用英文报刊完成举报操作。

但为了增加文章的混淆度，只需满足每个单词中字母数量一致即可，不关注每个字母的顺序。

解释：单词“`on`”允许通过单词“`no`”进行替代

报纸代表 `newspaper`，匿名信代表 `anonymousLetter`，求报纸内容是否可以拼成匿名信。

输入描述

第一行输入 `newspaper` 内容，包括 1-N 个字符串，用空格分开

第二行输入 `anonymousLetter` 内容，包括 1-N 个字符串，用空格分开

- 1、`newspaper` 和 `anonymousLetter` 的字符串由小写英文字母组成且每个字母只能使用一次
- 2、`newspaper` 内容中的每个字符串字母顺序可以任意调整,但必须保证字符串的完整性(每个字符串不能有多余字母)
- 3、 $1 < N < 100$, $1 \leq \text{newspaper.length}, \text{anonymousLetter.length} \leq 104$

输出描述

如果报纸可以拼成匿名信返回 `true`，否则返回 `false`

光伏场地建设规划

题目描述

祖国西北部有一片大片荒地，其中零星的分布着一些湖泊，保护区，矿区；整体上常年光照良好，但是也有一些地区光照不太好。某电力公司希望在这里建设多个光伏电站，生产清洁能源。对每平方公里的土地进行了发电评估，其中不能建设的区域发电量为 `0kw`，可以发电的区域根据光照，地形等给出了每平方公里年发电量 `x` 千瓦。我们希望能够找到其中集中的矩形区域建设电站，能够获得良好的收益。

输入描述

第一行输入为调研的地区长，宽，以及准备建设的电站【长宽相等，为正方形】的边长，最低要求的发电量

之后每行为调研区域每平方公里的发电量

例如，输入为：

2 5 2 6

1 3 4 5 8

2 3 6 7 1

表示调研的区域大小为长 2 宽 5 的矩形，我们要建设的电站的边长为 2，建设电站最低发电量为 6

输出描述

输出为这样的区域有多少个

上述输入长宽为 2 的正方形满足发电量大于等于 6 的区域有 4 个。

则输出为：

4

数组的中心位置

题目描述

给你一个整数数组 `nums`，请计算数组的中心位置

。数组中心位置是数组的一个下标，其左侧所有元素相乘的积等于右侧所有元素相乘的积。

数组第一个元素的左侧积为 1，最后一个元素的右侧积为 1

如果数组有多个中心位置，应该返回最靠近左边的那一个。如果数组不存在中心位置，返回 -1。

输入描述

输入只有一行，给出 `N` 个正整数用空格分格：`nums = 2 5 3 6 5 6`

`1 <= nums.length <= 1024`

`1 <= nums[i] <= 10`

输出描述

输出：3

解释：

中心位置是 3 。

左侧数之积 $\text{sum} = \text{nums}[0] * \text{nums}[1] * \text{nums}[2] = 2 * 5 * 3 = 30$ ，

右侧数之积 $\text{sum} = \text{nums}[4] * \text{nums}[5] = 5 * 6 = 30$ ，二者相等。

静态代码扫描服务

题目描述

静态扫描快速快速识别源代码的缺陷，静态扫描的结果以扫描报告作为输出：

1、文件扫描的成本和文件大小相关，如果文件大小为 N ，则扫描成本为 N 个金币

2、扫描报告的缓存成本和文件大小无关，每缓存一个报告需要 M 个金币

3、扫描报告缓存后，后继再碰到该文件则不需要扫描成本，直接获取缓存结果

给出源代码文件标识序列和文件大小序列，求解采用合理的缓存策略，最少需要的金币数

。

输入描述

第一行为缓存一个报告金币数 M ， $1 \leq M \leq 100$

第二行为文件标识序列： $F_1, F_2, F_3 \dots F_n$ ，其中 $1 \leq N \leq 10000$ ， $1 \leq F_i \leq 1000$

第三行为文件大小序列： $S_1, S_2, S_3 \dots S_n$ ，其中 $1 \leq N \leq 10000$ ， $1 \leq S_i \leq 10$

输出描述

采用合理的缓存策略，需要的最少金币数

通信误码

题目描述

信号传播过程中会出现一些误码，不同的数字表示不同的误码 ID，取值范围为 $1 \sim 65535$ ，用一个数组记录误码出现的情况。每个误码出现的次数代表误码频度，请找出记录中包含频度最高误码的最小子数组长度。

输入描述

误码总数目：取值范围为 $0 \sim 255$ ，取值为 0 表示没有误码的情况。

误码出现频率数组：误码 ID 范围为 $1 \sim 65535$ ，数组长度为 $1 \sim 1000$ 。

输出描述

包含频率最高的误码最小子数组长度