

## Задача A. Admin's Trouble

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 512 mebibytes

Как известно, два наиболее распространённых формата записи даты — это европейский (сначала день, потом месяц, потом год) и американский (сначала месяц, потом день, потом год).

При попытке установить защищённое ПО образца середины 2000-х, разработанное уже давно не существующей компанией и так и не модернизированное с того времени, системный администратор поменял дату на одном из бэкапов и сейчас хочет вернуть дату обратно. Но он не проверил, в каком формате дата используется в системе. Может ли он обойтись без этой информации?

Иначе говоря, вам даётся запись некоторой корректной даты. Требуется выяснить, однозначно ли по этой записи определяется дата даже без дополнительной информации о формате.

### Формат входных данных

Первая строка входных данных содержит три целых числа —  $x$ ,  $y$  и  $z$  ( $1 \leq x \leq 31$ ,  $1 \leq y \leq 31$ ,  $1970 \leq z \leq 2069$ ). Гарантируется, что хотя бы в одном формате запись  $x y z$  задаёт корректную дату.

### Формат выходных данных

Выведите 1, если дата определяется однозначно, и 0 в противном случае.

### Пример

| standard input | standard output |
|----------------|-----------------|
| 1 2 2003       | 0               |
| 2 29 2008      | 1               |

### Note

В первом примере при одной системе записи дата равна 1 февраля, при другой - 2 января 2003 года, то есть однозначно назвать дату не получается.

Во втором примере корректный вариант даты может быть только в американском формате, где она задаёт 29 февраля 2008 года.

## Задача В. Bonus

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

К юбилейному Чемпионату МТС по программированию модный Web-дизайнер Артемий Гусев разработал баннер  $M \times N$ , заполненный заглавными латинскими буквами 'М', 'Т' и 'С'. Дизайнер запросил премию, равную количеству способов прочитать слово "МТС" на баннере.

Считается, что в таблице можно прочитать слово "МТС" если в некоторой клетке записана буква "М", в одной из соседних с ней по стороне — буква "Т", а в одной из соседних уже с ней по стороне — буква "С". Способы считаются различными, если хотя бы одна из клеток, задействованных при прочтении слова этими способами, различается.

Ваша задача — определить сумму премии дизайнеру.

### Формат входных данных

Первая строка входа содержит два целых числа  $M$  и  $N$  — размерность таблицы ( $1 \leq M \leq 1000$ ,  $1 \leq N \leq 1000$ ). Каждая из последующих  $M$  строк содержит по  $N$  заглавных латинских букв 'М', 'Т' или 'С'.

### Формат выходных данных

Выведите одно число — сумму премии дизайнеру.

### Пример

| standard input      | standard output |
|---------------------|-----------------|
| 2 4<br>МТСМ<br>ММТМ | 4               |

### Note

В примере к задаче слово можно прочитать четырьмя способами. Если координаты верхнего левого поля обозначим как  $(1, 1)$ , то возможные способы:

1.  $(1, 1) - (2, 1) - (3, 1)$ ;
2.  $(2, 2) - (2, 1) - (3, 1)$ ;
3.  $(2, 2) - (3, 2) - (3, 1)$ ;
4.  $(4, 2) - (3, 2) - (3, 1)$ .

## Задача C. Cards Sorting

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 512 mebibytes

По заказу ФСБ (Федерации Спортивного Бриджа) Ваш отдел разрабатывает специальную программу, раздающую карты и сортирующую их в руке. Обычно бриджисты раскладывают карты в руке по мастям, а уже внутри масти — по убыванию номинала.

Вам поручили ответственную задачу: написать модуль, сортирующий карты внутри масти.

Обозначения карт (перечислены в порядке убывания номинала):

туз — A, король — K, дама — Q, валет — J, десятка — 10, девятка — 9, восьмёрка — 8, семёрка — 7, шестёрка — 6, пятёрка — 5, четвёрка — 4, тройка — 3, двойка — 2.

Набор карт задаётся строкой без пробелов, составленной из соответствующих обозначений; при этом ни одна карта не повторяется. Например, расклад “K9A10J2” — это король, девятка, туз, десятка, валет и двойка в одной масти. После сортировки должно получиться “AKJ1092”.

По заданному набору карт выведите его отсортированный вариант.

### Формат входных данных

Входные данные состоят из одной строки, представляющей собой перечисление карт в заданном формате (без пробелов между картами). Гарантируется, что строка непуста и что карта каждого номинала встречается в строке не более одного раза.

### Формат выходных данных

Выведите одну строку — выданный Вам набор карт, отсортированный по убыванию номинала.

### Пример

| standard input | standard output |
|----------------|-----------------|
| K9A10J2        | AKJ1092         |
| Q              | Q               |

## Задача D. Distant Control

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 512 mebibytes

Это интерактивная задача.

В комнате отдыха стоят несколько чашек. Каждая чашка вмещает одинаковое количество миллилитров жидкости. Вы хотите узнать объём чашки, но у вас завтра дедлайн по текущему проекту, и идти в комнату отдыха вам некогда.

Вы можете попросить коллег помочь вам. Для этого вы можете выдаёте ему термос, в который налито известное вам количество миллилитров чая, попросить его разлить чайник по чашкам так, чтобы все чашки были налиты полностью, попросить коллег выпить чаю за успех проекта, а остаток принести вам в чайнике (после чего вы можете установить объём оставшегося чая).

Но у вас с собой только два термоса, так что вы должны узнать объём чашки не более, чем за два запроса!

### Протокол взаимодействия

Сначала Вы выводите программе жюри запрос в формате “?  $X$ ”, где  $1 \leq X \leq 10^6$  — количество чая в термосе, который Вы передаёте. На такой запрос программа жюри выдаст оставшееся после разлива количество чая. Если вы хотите ответить объём чашки, сделайте запрос в формате “!  $V$ ”.

Не забудьте вывести символ конца строки после каждого запроса, а также очистить буфер вывода вызовом функции `flush`.

Гарантируется, что объём чашки — целое число, удовлетворяющее неравенству  $2 \leq V \leq 10^6$ .

### Пример

| standard input | standard output |
|----------------|-----------------|
| 3              | ? 8             |
| 2              | ? 12            |
|                | ! 5             |

## Задача Е. Eastern Tale

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 512 mebibytes

Как-то раз пришёл Ходжа Насреддин к визирю эмира договариваться о сумме договора на обучение ишака арабскому языку. Визирь договорился об аудиенции у эмира. Тот написал на бумаге несколько цифр и сказал «О мудрый Ходжа! Ты можешь переставить цифры в этом числе как угодно и вычеркнуть те цифры, которые пожелаешь, и получится число золотых монет, которое ты получишь в казне. Только одно условие: так как ты должен поделиться вознаграждением с ишаком и с визирем, то количество монет должно делиться на три».

Приздумался Ходжа Насреддин: как в записанном числе переставить цифры (и возможно, выкинуть некоторые из цифр) так, чтобы в результате и условие эмира выполнить, и как можно больше денег получить. Помогите ему в этом.

### Формат входных данных

Входные данные содержат одну непустую строку из цифр, начинающуюся ненулевой цифрой — написанное эмиром число. Число состоит не более, чем из  $10^4$  цифр.

### Формат выходных данных

Выведите наибольшее кратное 3 число, которое можно составить, переставляя написанные цифры и выбрасывая некоторые из них, и  $-1$ , если этого сделать не получится.

### Примеры

| standard input | standard output |
|----------------|-----------------|
| 1              | -1              |
| 2020           | 0               |
| 98776          | 9876            |

## Задача F. Find the Base

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 512 mebibytes

Вам задано несколько примеров типа  $A + B = C$  или  $A * B = C$ , в которых  $A$ ,  $B$  и  $C$  — строки, состоящие из цифр или строчных латинских букв (соответствующих цифрам от 10 до 35), причём первый символ каждой строки отличен от 0. Требуется проверить, можно ли однозначно восстановить основание системы счисления, в которой все эти примеры являются тождествами.

### Формат входных данных

В первой строке  $N$  — число примеров ( $1 \leq N \leq 1000$ ), затем в последующих  $N$  строках — тождества в формате

$$X_1 \dots X_l * Y_1 \dots Y_m = Z_1 \dots Z_k$$

или

$$X_1 \dots X_l + Y_1 \dots Y_m = Z_1 \dots Z_k$$

где все  $X_i$ ,  $Y_i$ ,  $Z_i$  — цифры от 0 до 9 или латинские буквы от 'a' до 'z' (цифры от 10 до 35), ( $1 \leq m, l, k \leq 10$ ). Гарантируется, что в случае, если основание восстанавливается однозначно, все числа в примерах не превосходят  $2^{64} - 1$ .

### Формат выходных данных

Вывести основание используемой во всех данных примерах системы счисления, если такая существует и восстанавливается однозначно. Выведите 0, если такой системы не существует или -1, если существует более одной системы счисления, в которой эти примеры верны. Основание необходимо вывести в десятичном виде.

### Примеры

| <i>standard input</i>               | <i>standard output</i> |
|-------------------------------------|------------------------|
| 3<br>2*2=4<br>25*25=625<br>10+10=20 | 10                     |
| 1<br>2*2=5                          | 0                      |
| 1<br>2*2=4                          | -1                     |

## Задача G. Grouping Circles

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 512 mebibytes

На плоскости расположен набор из  $n$  окружностей. Каждая окружность задана тремя различными точками, через которые она проходит. Найдите наибольшее количество окружностей, имеющих один и тот же радиус.

### Формат входных данных

В первой строке входного файла задано целое число  $n$  ( $1 \leq n \leq 1400$ ). В последующих  $n$  строках заданы по шесть целых чисел  $x_1, y_1, x_2, y_2, x_3, y_3$  — координаты трёх точек, не лежащих на одной прямой и задающих соответствующую окружность ( $0 \leq x_i, y_i \leq 1400$ ).

### Формат выходных данных

Одно целое число — количество окружностей, имеющих наиболее часто встречающийся в наборе радиус.

### Примеры

| standard input                                 | standard output |
|--|-----------------|
| 3<br>0 0 0 1 1 0<br>0 0 2 0 0 2<br>1 1 1 2 2 1 | 2               |

c

## Задача Н. Highest Tower

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 512 mebibytes

Заданы  $n$  кирпичей, пронумерованных числами от 1 до  $n$ . Все кирпичи имеют в сечении единичный квадрат и могут отличаться только длиной.

Требуется построить из существующих кирпичей башню максимальной высоты, состоящую из нескольких положенных друг на друга рядов кирпичей (уровней), удовлетворяющую следующим правилам:

- Каждый уровень представляет собой прямоугольный блок  $1 \times 1 \times l_i$ , составленный из некоторого количества кирпичей.
- Суммарная длина  $l_i$  кирпичей на каждом уровне не превосходит суммарной длины кирпичей на уровне, расположенном непосредственно под ним (если таковые имеются).
- Каждый блок, расположенный на некотором уровне, имеет номер больше, чем любой из блоков, расположенных ниже этого уровня (если таковые имеются).

При построении башни должны быть использованы все имеющиеся в наличии кирпичи.

### Формат входных данных

В первой строке входного файла задано одно целое число  $n$  ( $1 \leq n \leq 10^5$ ) — количество кирпичей. В следующей строке заданы длины кирпичей —  $n$  целых положительных чисел, не превышающих  $10^4$ .

### Формат выходных данных

Выведите одно число — максимальную высоту башни, которую можно построить из заданного набора кирпичей.

### Пример

| standard input | standard output |
|----------------|-----------------|
| 3<br>2 3 4     | 2               |



## Задача I. Invasion

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 2 секунда  
Ограничение по памяти: 512 мегабайт

Город Байтсбург подвергся атаке зомби!

Специальному агенту 0x07 удалось пробраться в эпицентр нашествия и выяснить, что зомби появляются из находящегося в заброшенном доме ящика. Чтобы закрыть ящик, нужно набрать код на специальном пульте. На пульте находятся 10 кнопок, соответствующих цифрам от 0 до 9, а также две кнопки «+» и «-». Пульт устроен так, что код можно или сразу набрать на цифровой клавиатуре (потратив по одному нажатию на каждую цифру кода), или набрать какое-то число и после этого использовать стрелки для увеличения или уменьшения набранного числа на 1 (при этом, если текущее число равно 1, кнопка «-» не приводит к изменению числа). К сожалению, пульт уже побывал в руках зомби, так что какие-то цифровые кнопки сломались.

Агент 0x07 исследовал пульт и выяснил следующие детали:

- Начальное состояние пульта (пустая строка) не соответствует никакому коду.
- Работает хотя бы одна цифровая кнопка и обе кнопки «+» и «-».
- «+» и «-» разблокируются **только после того**, как была нажата хотя бы одна цифровая кнопка.
- После первого нажатия «+» и «-» цифровые кнопки **блокируются**.
- При наборе цифр на цифровых кнопках они набираются слева направо, то есть для ввода, например, числа 2020 надо нажать кнопки '2', '0', '2' и '0' (именно в указанной последовательности). При этом после первого нажатия будет набрано число 2, после второго — 20, после третьего — 202.

Агент 0x07 собирается закрыть ящик, сделав как можно меньше нажатий. По заданному состоянию клавиатуры (списку неработающих цифровых кнопок) и заданному коду выясните, сколько нажатий ему потребуется.

### Формат входных данных

Первая строка входного файла содержит одно целое число  $B$  — количество сломанных цифровых кнопок ( $1 \leq B \leq 9$ ). Вторая строка содержит  $B$  попарно различных целых чисел  $b_i$  ( $0 \leq b_i \leq 9$ ) — цифры, написанные на неработающих кнопках. Третья строка содержит одно целое число  $N$  ( $1 \leq N \leq 10^{18}$ ) — код закрытия. Гарантируется, что запись числа  $N$  не содержит ведущих нулей.

### Формат выходных данных

Выведите одно целое число — минимальное количество нажатий на кнопки, которое потребуется агенту 0x07, чтобы закрыть дорогу зомби.

### Примеры

| standard input  | standard output |
|-----------------|-----------------|
| 1<br>0<br>2020  | 25              |
| 2<br>1 9<br>117 | 31              |

## Задача J. Japanese Branch of MTC

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 512 mebibytes

Компания МТС открыла новый филиал. На этот раз — в Японии.

Вы — ведущий системный администратор, командированный в филиал для обустройства сети. Но сеть, как оказалось, надо прокладывать в соответствии с японскими понятиями о субординации.

Вы получили от японских коллег схему разводки сети по офису.. Из этих указаний следует, что в сети должен быть ровно один свитч, в который воткнут ровно один кабель, ровно два свитча, в которые воткнуты ровно два кабеля и так далее. Иначе говоря, для каждого  $1 \leq i \leq N$  в конфигурации сети должны быть ровно  $i$  свитчей, в которые воткнуты ровно  $i$  кабелей.

Никаких других свитчей и кабелей в сети быть не должно. Все свитчи должны быть соединены в одну сеть, то есть данные между любыми двумя свитчами могут быть переданы по системе кабелей и свитчей. Любой кабель соединяет ровно два свитча, никакие два свитча не должны быть соединены напрямую более, чем одним кабелем.

От Вас требуется по заданному  $N$  построить конфигурацию сети, то есть перечислить, какие свитчи будут соединены кабелями, или сообщить, что для заданного  $N$  такую сеть построить невозможно.

### Формат входных данных

Входной файл содержит одно целое число  $N$  ( $1 \leq N \leq 239$ ) — параметр сети.

### Формат выходных данных

Если для заданного  $N$  требуемой конфигурации сети не существует, выведите  $-1$ .

В противном случае выведите конфигурацию, удовлетворяющую условиям задачи: для каждого кабеля в отдельной строке выведите номера свитчей, которые он соединяет. Свитчи должны быть занумерованы подряд, начиная с единицы. Если конфигураций несколько, выведите любую.

### Пример

| standard input | standard output                               |
|----------------|---|
| 3              | 1 2<br>2 3<br>1 3<br>1 4<br>2 5<br>3 6<br>4 5 |

## Задача К. King's Observatory

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 512 mebibytes

При раскопках на месте обсерватории короля Артура археологи нашли две рукописи, предположительно созданные самим Мерлином.

В одной из них был список созвездий, в другой — список всех известных астрономам звёзд (все входящие в созвездия звёзды и возможно, некоторые звёзды, в них не входящие).

Учёные подсчитали количество звёзд в каждом созвездии и количество звёзд во второй рукописи.

По заданным числам требуется проверить, можно ли гарантированно утверждать, что не все звёзды входят в созвездия, а также можно ли гарантированно утверждать, что какие-то два созвездия пересекаются.

### Формат входных данных

Первая строка входных данных содержит два целых числа  $N$  и  $S$  — количество созвездий и количество звёзд во второй рукописи, соответственно ( $2 \leq N \leq 100$ ,  $1 \leq S \leq 10^4$ ). Во второй строке заданы  $N$  чисел  $a_i$  — количество звёзд в  $i$ -м созвездии ( $1 \leq a_i \leq 10^4$ ).

### Формат выходных данных

Если из входных данных следует, что какая-то звезда неизбежно принадлежит более, чем одному созвездию, выведите 1. Если из них следует, что какая-то звезда неизбежно не входит в созвездия, выведите  $-1$ . Иначе выведите 0.

### Примеры

| standard input | standard output |
|----------------|-----------------|
| 3 15<br>4 5 5  | -1              |
| 3 15<br>5 5 5  | 0               |
| 3 15<br>7 8 8  | 1               |