

## Задача А. Игра в кости

Имя входного файла: `dice.in`  
 Имя выходного файла: `dice.out`  
 Ограничение по времени: 2 seconds  
 Ограничение по памяти: 64 Mebibytes

Представитель Космического Союза, заинтересованный в непрерывных поставках гоплана, необходимого навигаторам для того, чтобы как-то справляться с управлением неотлаженными системами на борту кораблей, нанёс визит на Ерракис.

Старый барон Харконнен принял гостя в своей резиденции. Когда представитель поднял вопрос о возможности оптовых закупок и соответствующем снижении цены, барон предложил представителю сыграть в кости. Выиграет барон — цена остаётся прежней, а объёмы увеличиваются. Выиграет представитель — требуемая скидка будет предоставлена.

Согласно традиций, берущих начало со Второго Кодекса РППГ (о том, что это был за Кодекс, не знает толком даже Устная История), для игры могут использоваться кости в виде многогранников с 4, 6, 8, 10, 12 и 20 гранями. Игральная кость с  $n$  гранями, на которые нанесены различные числа от 1 до  $n$ , обозначается как  $dn$ , при этом вероятность выпадения каждого из этих чисел равна  $1/n$ .

Таким образом, существуют 6 типов костей —  $d4$ ,  $d6$ ,  $d8$ ,  $d10$ ,  $d12$ ,  $d20$ . Барон вытащил несколько костей и предложил сделать ставки. Представитель Союза, опасаясь нечестной игры со стороны Харконнена, хочет знать, какова вероятность выпадения заданного числа на заданном наборе костей.

### Формат входного файла

Входной файл состоит из нескольких тестовых примеров, каждый из которых расположен на отдельной строке. Сначала задано целое число  $d$  ( $1 \leq d \leq 13$ ) — количество костей, участвующих в игре. Далее идёт описание костей (для каждой из  $d$  костей — одна из строк  $d4$ ,  $d6$ ,  $d8$ ,  $d10$ ,  $d12$  или  $d20$ ). Далее идёт целое число  $x$  ( $0 \leq x \leq 1000$ ), вероятность выпадения которого при броске заданного набора и необходимо определить. Ввод заканчивается строкой, начинающейся с 0, которую обрабатывать не надо.

### Формат выходного файла

В выходной файл для каждого примера в своей строке выведите требуемую вероятность с 5 точными знаками после десятичной точки.

### Пример

<code>dice.in</code>	<code>dice.out</code>
1 d10 5	0.10000
2 d10 d6 9	0.10000
2 d8 d8 9	0.12500
0	

## Задача В. Оптимизация

Имя входного файла: `opt.in`  
 Имя выходного файла: `opt.out`  
 Ограничение по времени: 4 seconds  
 Ограничение по памяти: 64 Mebibytes

Тем временем Пол Атридес и его мать Джессика были обнаружены в пустыне отрядом Свободных, предводителем которых был Стилгар. После ряда испытаний было принято решение принять обоих беглецов как полноправных членов отряда.

Прибыв в съетч (так называется поселение Свободных на их языке), Пол поразился тому, как эффективно и экономно организована жизнь Свободных. Например, в ремонтных мастерских работали всего три мастера. Каждое из устройств, нуждавшихся в ремонте, требовало определённого времени на его ремонт вне зависимости от мастера, который этот ремонт производит (при этом работать над одним устройством одновременно несколько мастеров не могут). Зная время на ремонт каждого устройства, мастера распределили имеющиеся устройства так, что работа была закончена довольно быстро. Подошедший Стилгар заметил, что мастера пока ещё довольно неопытные и можно было распределить работу оптимальнее, завершив её за меньшее время. По заданному количеству устройств и времени, требующемуся на ремонт каждого устройства, требуется найти минимальное время, которое понадобится для того, чтобы отремонтировать все устройства.

### Формат входного файла

Входной файл содержит несколько (не более, чем 120) тестовых примеров. Каждый из примеров начинается с новой строки. Первым числом примера идёт целое число  $n$  ( $1 \leq n \leq 40$ ) — количество устройств, требующих ремонта. Далее в той же строке заданы  $n$  целых чисел  $t_i$  ( $1 \leq t_i \leq 30$ ) — время на ремонт  $i$ -го устройства. Ввод заканчивается строкой, содержащей одно число 0. Эту строку обрабатывать не надо.

### Формат выходного файла

Для каждого тестового примера с новой строки выведите в выходной файл минимальное время, за которое может быть закончен ремонт всех устройств.

### Пример

<code>opt.in</code>	<code>opt.out</code>
1 30	30
3 15 10 20	20
5 6 7 8 9 10	15
0	

## Задача С. Набег контрабандистов

Имя входного файла: `run.in`  
 Имя выходного файла: `run.out`  
 Ограничение по времени: 2 seconds  
 Ограничение по памяти: 64 Mebibytes

Один из соратников герцога Лито Атридеса, Гурни Хэллек, после нападения Харконненов попал к контрабандистам, где довольно быстро занял одну из командных позиций.

Сейчас группе под командованием Гурни предстоит собрать гоплан на одном из недавно открытых мест его выхода на поверхность Ерракиса. Схема этого участка выглядит следующим образом.

Имеется  $n$  посадочных площадок, соединённых таким образом, что от одной площадки до другой харвестер (комбайн для сборки гоплана) может доехать только по единственному маршруту. Иначе говоря, схема путей представляет собой дерево, «листьями» которого являются эти  $n$  площадок, а внутренними узлами — не более, чем  $n - 1$  перекрёстков, в которых пути могут сходиться и расходиться (при этом в каждом перекрёстке сходятся не менее 3 путей). С каждого перекрёстка харвестер собирает  $t$  грамм гоплана, а с каждого метра пути —  $r$  грамм. Из-за опасности нападения червя харвестер может пройти путь только от одной посадочной площадки до другой, не разворачиваясь и не проходя никакую точку маршрута дважды.

У контрабандистов имеется таблица, в которую занесены расстояния вдоль путей между каждыми двумя посадочными площадками. Необходимо выбрать такие две посадочные площадки, чтобы количество собранного гоплана при путешествии харвестера между ними было максимально.

### Формат входного файла

Входной файл состоит из нескольких (не более, чем 120) тестовых примеров. Каждый пример начинается с одной строки, содержащей три целых числа:  $n$  ( $1 \leq n \leq 50$ ) — количество посадочных площадок,  $r$  ( $1 \leq r \leq 10$ ) — количество гоплана, собираемого харвестером с метра пути,  $t$  — количество гоплана, собираемого харвестером с каждого перекрёстка. Далее, в следующих  $n$  строках заданы  $n$  чисел — расстояния вдоль путей между  $i$ -й и  $j$ -й посадочными площадками  $d_{ij}$ , то есть  $j$ -е число в  $i$ -й строке задаёт  $1 \leq d_{ij} \leq 1000$ . Гарантируется, что данные корректны, то есть  $d_{ii} = 0$ ,  $d_{ij} = d_{ji}$ , а также существует дерево путей, удовлетворяющее условию задачи и порождающее заданную матрицу расстояний. Входной файл заканчивается строкой, состоящей из одного числа 0, которую обрабатывать не надо.

### Формат выходного файла

Для каждого тестового примера выведите в отдельной строке максимальное количество гоплана, которое сможет собрать харвестер по пути между некоторыми двумя посадочными площадками.

### Пример

run.in	run.out
9 1 5 0 8 22 16 16 13 24 14 11 8 0 20 14 14 11 22 12 9 22 20 0 12 12 11 22 12 23 16 14 12 0 4 5 16 6 17 16 14 12 4 0 5 16 6 17 13 11 11 5 5 0 13 3 14 24 22 22 16 16 13 0 14 25 14 12 12 6 6 3 14 0 15 11 9 23 17 17 14 25 15 0 0	38

## Задача D. Точная сумма

Имя входного файла: `change.in`  
 Имя выходного файла: `change.out`  
 Ограничение по времени: 3 seconds  
 Ограничение по памяти: 64 Mebibytes

Заказчик на собранный контрабандистами гоплан нашёлся быстро. Преподобная Мать Юс Коу, возглавляющая отделение Бене Джессерит, расположенное в торговом городе Vor36, была готова заплатить большую сумму денег за некоторое количество гоплана, требующееся для проведения ежегодной ритуальной церемонии в Vor36.

Для того, чтобы уменьшить шансы перехвата контрабандного гоплана патрулями Харконненов во время совершения сделки, контрабандисты требуют необходимую сумму ровно (то есть без сдачи). Юс Коу не знает, какую конкретно сумму запросят за гоплан, но ей известно, что сумма выражается целым числом безусловных единиц и не превосходит  $C$  безусловных единиц. У Преподобной Матери имеются с собой купюры различного достоинства. Она хочет заранее подготовить минимальное количество купюр, такое, что с их помощью она сможет выплатить любую сумму от 1 до  $C$  безусловных единиц.

### Формат входного файла

Входной файл состоит из нескольких (не более, чем 120) тестовых примеров. Каждый тестовый пример начинается со строки, содержащей два целых числа:  $C$  ( $1 \leq C \leq 10^9$ ) — максимальная сумма, которую надо выплатить и  $m$  ( $1 \leq m \leq 1000$ ) — количество различных номиналов купюр, имеющихся у Юс Коу.

Далее следуют  $m$  строк, в каждой из которых заданы по два целых положительных числа, не превосходящих 1000: номинал  $v_i$  соответствующей купюры и количество  $n_i$  купюр данного номинала у Юс Коу. Входной файл завершается строкой, содержащей единственное число 0, которую обрабатывать не надо.

### Формат выходного файла

Для каждого тестового примера в выходной файл выведите одно число — минимальное количество купюр, которое нужно приготовить, чтобы суметь точно оплатить любую сумму от 1 до  $C$ , выражаемую целым числом безусловных единиц, или `Not possible`, если для какого-то значения суммы точная выплата невозможна.

### Пример

change.in	change.out
4 2	3
2 1	Not possible
1 3	
9 3	
1 5	
8 2	
7 1	
0	

## Задача E. Древние записи

Имя входного файла: `ancient.in`  
 Имя выходного файла: `ancient.out`  
 Ограничение по времени: 2 seconds  
 Ограничение по памяти: 64 Mebibytes

Во время одной из вылазок группа контрабандистов во главе с Гурни Хэллеком была захвачена отрядом Свободных. Каково же было удивление Гурни, когда он увидел, что во главе этого отряда стоит Пол Атридес, принявший у Свободных имя Муад Диб (по названию небольшого зверька, похожего на мышь и обитающего в пустынях Ерракиса).

Оказалось, что информации, которая имеется у Пола и у Гурни, хватает для того, чтобы разработать план возвращения контроля Атридесов над Ерракисом. Правда, для того, чтобы гарантировать невмешательство извне, необходимо было прочитать записи, сделанные ещё в эпоху начала освоения Ерракиса. Предполагалось, что эти записи могут содержать информацию, позволяющую уничтожить весь гоплан на Ерракисе. Подобная угроза остановила бы любое внешнее вмешательство.

Записи, выполненные в виде мозаики на стене подземной пещеры, оказались почти неповреждёнными, за исключением нескольких слов, в которых осыпались все буквы. Часть букв на выпавших камнях можно было прочитать, часть нет (то есть это могла бы быть любая буква).

Из логики текста можно предположить, какие слова могли быть на месте повреждённых. Однако для проверки этих предположений надо знать, можно ли составить данное слово из данного набора камней (с оставшимися или же стёртыми буквами).

### Формат входного файла

Входной файл состоит из нескольких (не более 120) тестовых примеров. Каждый тестовый пример начинается со строки, содержащей целое число  $1 \leq n \leq 1000$  — количество возможных различных слов, которые могут оказаться на месте повреждённого слова. Следующие  $n$  строк содержат сами эти слова, состоящие из заглавных латинских букв. Длина каждого слова не менее 1 буквы и не более 7 букв. Последняя строка примера описывает набор камней: `_`, если буква на камне стёрта, или заглавная латинская буква, сохранившаяся на камне. Общее количество камней также не менее одного и не более 7. Входной файл завершается строкой из одного числа 0, которую обрабатывать не надо.

### Формат выходного файла

В выходной файл для каждого тестового примера выведите одно число — количество слов из списка, которые могли быть составлены из данного набора камней.

### Пример

<code>ancient.in</code>	<code>ancient.out</code>
5	3
GOPLAN	2
ERRAKIS	
WORM	
STORM	
VOR	
_MORT	
3	
WATER	
SAND	
GOPLAN	
A__AA__	
0	

## Задача F. Расстановка войск

Имя входного файла:	go.in
Имя выходного файла:	go.out
Ограничение по времени:	2 seconds
Ограничение по памяти:	64 Mebibytes

И вот началась подготовка решающей атаки на позиции Харконненов. Для начала Пол Муад Диб и Гурни Хэллек решили найти наилучшее расположение для контролируемых ими сил Свободных.

Схема предстоящего сражения была нарисована на листе клетчатой бумаги  $n \times n$ . Боевые единицы Свободных были представлены белыми фишками, расположенными в узлах решётки, боевые единицы Харконненов — чёрными. При этом в некоторых узлах решётки может не быть ни одной фишки, и ни в одном узле решётки нет двух и более фишек. Пол заметил, что по правилам искусства стратегии, восходящего к древней игре «Го», позиция каждой стороны оценивается как количество незанятых узлов, образующих территории, окружённые фишками, изображающими войска данной стороны, а общая оценка позиции — как разность оценок позиций сторон.

Гурни не был знаком с правилами этой игры, и тогда Муад Диб уточнил способ оценки.

Территория — это множество незанятых узлов, между любыми двумя различными элементами которого существует путь по линиям решётки, проходящий только по узлам, принадлежащим данной территории, причём любой незанятый узел, в который существует аналогичный путь из узла, принадлежащего территории, также принадлежит этой территории. Тогда площадью территории называется количество узлов, в неё входящих.

Граница территории — множество всех занятых фишками узлов, непосредственно соединённых с узлами, принадлежащими данной территории (то есть находящихся непосредственно справа, слева, сверху или снизу от какого-нибудь узла, принадлежащего данной территории); территория считается окружённой данным цветом, если все фишки, образующие её границу, имеют этот цвет.

Для того, чтобы помочь в анализе стратегической ситуации, по заданной Вам карте требуется оценить заданную на ней позицию.

### Формат входного файла

Входной файл состоит из нескольких (не более, чем 120) тестовых примеров. Каждый пример начинается со строки, содержащей три числа:  $n$  ( $1 \leq n \leq 19$ ) — длина стороны схемы,  $b$  и  $w$  — количество соответственно чёрных и белых фишек (обозначающих группы войск Харконненов и Свободных) ( $b \geq 0, w \geq 0, 1 \leq b + w \leq n^2$ ). Во второй строке находятся  $b$  пар целых чисел  $r_1 c_1 \dots r_b c_b$  ( $1 \leq r_i, c_i \leq n$ ) — координаты чёрных фишек (первое число — номер ряда, второй — номер колонки), в третьей строке аналогичным образом заданы координаты  $w$  белых фишек. Если чёрных или белых фишек нету, соответствующая строка остаётся пустой. Входной файл завершается строкой, содержащей одно число 0, эту строку обрабатывать не надо.

### Формат выходного файла

Для каждого теста в выходной файл один из трёх ответов: **White wins by N**, в случае, если оценка позиции белых больше на целое положительное  $N$ , **Black wins by N**, если оценка позиции чёрных больше на целое положительное  $N$ , **Draw**, если оценка позиции белых и чёрных совпадает. Более точно формат вывода показан в примере.

**Пример**

go.in	go.out
1 1 0	Draw
1 1	White wins by 3
2 0 1	Black wins by 1
1 1	
5 12 4	
1 1 1 2 1 3 2 1 2 3 3 1 3 3 4 1 4 3 5	
1 5 2 5 3	
1 4 2 4 3 4 3 5	
0	