

## Задача 1. Раз – горох, два – горох...

Входной файл: input.txt

Выходной файл: output.txt

Время на тест: 1 секунда

Приближалась зима, и Хома с Сусликом решили запастись горохом. Весь день они бегали в амбар и таскали по несколько стручков: Хома по четыре, а Суслик по два. К вечеру они пересчитали все стручки, что они натаскали, и задумались, как теперь этот горох делить. Хома утверждал, что если он за раз таскал в два раза больше, чем Суслик, то и гороха ему должно достаться в два раза больше. Суслик на это резонно возражал, что, во-первых, скорость у Хома заметно меньше, чем у Суслика, а, во-вторых, кто его знает, может Хома всего раз-два сбегал, а остальное время бездельничал...

Помогите друзьям хоть немного разобраться в этой сложной ситуации. Определите все возможные варианты того, сколько стручков притащил Суслик, а сколько Хома.

### Входные данные

В первой строке натуральное четное число  $M$  – количество украденных стручков,  $2 \leq M \leq 1000$ .

### Выходные данные

Все возможные сочетания количеств стручков, принесенных Сусликом и Хомой по одному сочетанию в строке. Каждое сочетание представляет собой два целых неотрицательных числа через пробел: первое число – количество стручков, принесенных Сусликом, второе – принесенных Хомой. Сочетания упорядочить по убыванию первого числа.

### Пример

input.txt

6

output.txt

6 0

2 4

## Задача 2. Грипп АН1N1

Входной файл: input.txt

Выходной файл: output.txt

Время на тест: 3 секунды

Любая эпидемия опасна. А тем более эпидемия такой легко передающейся болезни, как грипп. Поэтому правительства практически всех стран разрабатывают планы специальных мероприятий на случай эпидемий. Одной из наиболее эффективных мер при эпидемии является изоляция заболевших. При этом в масштабах страны изолировать иногда приходится целые города (что делать внутри каждого изолированного города, пусть решает мэрия этого города). Очевидно, что на случай изоляции города необходимо сохранить все коммуникации между остальными городами. Естественно, если коммуникации не сохраняются, то необходимо построить новые дороги (желательно, как можно меньше).

Президент одной из стран с развитой сетью дорог вызвал вашу команду для оценки ситуации. Он просит определить, останутся ли коммуникации между всеми городами (будем считать, что в этой стране коммуникации между всеми городами изначально есть), если изолировать один любой город (т.е. перекрыть все ведущие к нему дороги). При этом, если есть хотя бы один город, изоляция которого нарушает коммуникации, необходимо определить, достаточно ли построить всего одну дорогу, чтобы изоляция любого города не нарушала коммуникаций. Поскольку информация о всех дорогах страны является стратегической, президент не указал на сети дорог названия городов, а просто пронумеровал их.

### Входные данные

В первой строке через пробел два целых числа через ( $N$  – количество городов,  $2 \leq N \leq 20000$  и  $M$  – количество дорог,  $1 \leq M \leq 200000$ ). При этом два города могут соединяться только одной дорогой.

Далее в каждой строке по два целых числа – номера городов, которые соединяет дорога (количество строк равно количеству дорог).

### Выходные данные

Если изоляция любого города возможна (с построением не более чем одной дороги) в первой строке выдать текст YES, в противном случае в первой строке выдать текст NO

Если для возможности изоляции любого города необходимо построить всего одну дорогу, то во второй строке выдать два целых числа (номера городов) через пробел, описывающие любую такую дорогу.

### Пример

#### *input.txt*

```
4 4
1 2
2 3
1 3
3 4
```

#### *output.txt*

```
YES
1 4
```

### Задача 3. Доктор Хаус в морге

Входной файл: input.txt

Выходной файл: output.txt

Время на тест: 1 секунда

Герой популярного телесериала доктор Хаус в очередной раз разыграл свою начальницу, доктора Кадди, и она решила немного наказать-разыграть его. Для этого она хочет отправить Хауса в морг рассортировать трупы, лежащие в нумерованных ячейках. Трупы бывают трёх типов мужские (обозначим их буквой m), женские (f) и детские (c). От Хауса требуется, чтобы сначала лежали все детские трупы, после них – все женские, и только потом – мужские.

Напомним, что доктор Хаус – инвалид с больной ногой, и таскать трупы (даже на каталках) ему нелегко. Поэтому сердобольная доктор Кадди решила проверить, какое минимальное число переключений трупов ему придётся совершить (зная доктора Хауса как весьма неглупого человека, она догадывается, что он будет действовать оптимальным образом). По её просьбе санитар проверил содержимое ячеек и посчитал трупы. В представленном санитаром документе указано, сколько трупов какого типа лежат в ячейках подряд. Теперь Лизе Кадди осталось только написать соответствующую программу, для чего она решила нанять вас. Не желая раскрывать секрет своего розыгрыша (вдруг вы тоже поклонники Хауса), она формулирует для вас следующее задание.

Дана последовательность букв, в которой буквы всего трех типов: c, f, m. При этом для сокращения записи последовательность задана блоками одинаковых символов, идущих подряд. Требуется определить минимальное число перестановок букв, которое необходимо для упорядочивания последовательности по возрастанию кода буквы.

#### Входные данные

В первой строке одно натуральное число  $N$  ( $1 \leq N \leq 1000$ ). Далее  $N$  строк, в каждой из которых описание блока: буква ('c', 'f' или 'm'), из которой состоит данный блок, и через пробел натуральное число  $K$  ( $1 \leq K \leq 1000$ ) – размер блока.

#### Выходные данные

В первой строке одно число – требуемое минимальное количество перестановок.

#### Пример

*input.txt*

3

f 3

m 3

c 3

*output.txt*

6

## Задача 4. Задача Итана Ханта

**Входной файл:** input.txt

**Выходной файл:** output.txt

**Время на тест:** 1 секунда

Герой сериала «Миссия невыполнима» Итан Хант известен больше как диверсант. Однако однажды ЦРУ привлекло его в качестве эксперта по оценке надёжности кодов дверей в ЦРУ. Как известно (в основном, по популярным фильмам), кодовые замки в ЦРУ открываются с помощью магнитной карточки. Будем считать, что на этой карточке записана последовательность битов длины  $M$ . Последовательность битов считывается, и если записанные биты совпадают с кодом двери, дверь открывается. У Ханта спросили, как бы он вскрывал такой замок, и тот предложил следующую идею. Возьмём длинную магнитную ленту, запишем на неё произвольную последовательность длины  $N$  ( $N \geq M$ ) из 0 и 1, такую, что вероятность встретить в ней 0 или 1 одинаковы. Затем проведём этой лентой по замку. Если в этой последовательности где-то встретился код замка, замок откроется (напомним, что дело происходит в голливудском фильме, и то, что физическая кодировка сплошной ленты и магнитной карты никак не совпадают, не столь важно). В дирекции ЦРУ подумали и решили проверить все коды замков на хантоустойчивость.

### Входные данные

В первой строке число  $N$  - длина последовательности битов, записанных на ленте ( $0 < N < 61$ ). Во второй строке – код замка, т.е. непустая строка, состоящая из символов «0» и «1», длина которой не превышает  $N$ .

### Выходные данные

Вероятность наличия кода замка в произвольно взятой последовательности битов длины  $N$ . Вероятность выдается в виде несократимой дроби, числитель и знаменатель которой разделены символом «/».

### Пример

*input.txt*

3

00

*output.txt*

3/8

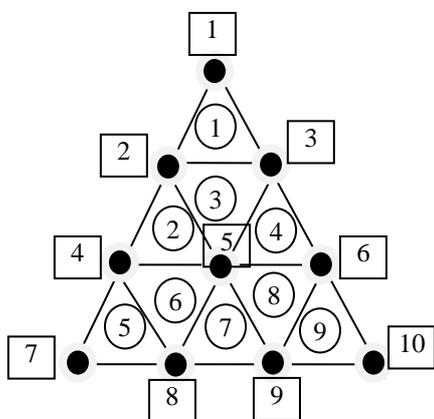
## Задача 5. Три поросёнка

Входной файл: input.txt

Выходной файл: output.txt

Время на тест: 1 секунда

Напомним, что после победы над волком все три поросёнка поселились у Наф-Нафа. Долгими зимними вечерами в тёплом каменном доме они читали книжки, играли... Но зима длинная, а все игры когда-нибудь надоедают. Решили поросята придумать новую настольную игру, похожую на шашки или шахматы, но на троих. Нарисовали они треугольную доску, на сторонах треугольника на равном расстоянии поставили по одинаковому количеству точек. Далее точки, поставленные на смежных сторонах, соединили отрезками, параллельными третьей стороне. В результате получилось треугольное поле с мелкими треугольными клеточками (его небольшой фрагмент изображен на рисунке).



А после этого поросята вспомнили, что когда они играют, то всегда спорят. И поэтому решили, что в их новой игре понадобится записывать ходы, а для этого нужно занумеровать треугольники и их вершины. Поросята пронумеровали вершины, выбрав одну за стартовую (на рисунке нумерация вершин показана в прямоугольных рамках), и треугольники (нумерация треугольников на рисунке в кружочках). А затем умный Наф-Наф сообразил, что номера вершин треугольного поля легко определить по его номеру, и поэтому при записи ходов не нужно записывать номера вершин. Например, у треугольника 7 вершины 5, 8 и 9, у треугольника 11 вершины 7, 8, 12 и т.д. Однако Ниф-Ниф и Нуф-Нуф не сразу поняли его идею, и им понадобилась программа, которая вычисляет но-

мера вершин треугольника по его номеру.

### Входные данные

В первой строке одно целое число – номер треугольника, положительный, не более 1 000 000 000 (будем считать, что поросята соорудили достаточно большую доску, чтобы такой треугольник на ней был).

### Выходные данные

Три целых числа через пробел – номера вершин указанного треугольника в порядке возрастания номеров.

### Пример

*input.txt*

6

*output.txt*

4 5 8

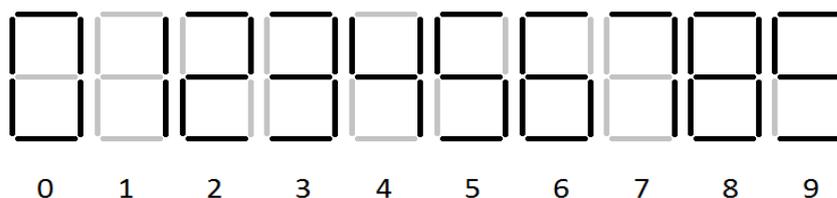
## Задача 6. Прогресс в артиллерии

Входной файл: input.txt

Выходной файл: output.txt

Время на тест: 1 секунда

Артиллерия всегда была родом войск, в котором интенсивно использовались различные счетные устройства. Когда-то они были механическими и показывали результаты вычислений посредством колёс с нарисованными цифрами. Однако прогресс не стоял на месте. Однажды конструкторы разработали электрическое устройство, которое показывало результаты с помощью сегментных индикаторов:



К его быстрдействию и точности претензий не возникло, а вот что касается способа отображения... Некий генерал-ретроград отнёсся к нему с большим недоверием. Эти индикаторы (рассуждал он) — довольно-таки ненадёжная штука. Стоит не загореться всего лишь одному нужному сегменту или, наоборот, загореться ненужному — и результат окажется искажённым. Хорошо ещё, если в результате искажения будет видна бессмыслица — оператор заметит. А если результат будет по-прежнему выглядеть правдоподобно, как нормальная последовательность цифр? Искажение всего одного разряда способно превратить 888 в 808. Два разряда, и 888 превращается в 800. Так недолго открыть огонь и по собственным позициям!

Необходимо разработать программу, которая по заданному числу определяет наибольшее отклонение, которое может возникнуть в его отображении при «ошибке одного сегмента» в одном и двух разрядах.

### Входные данные

В первой строке целое неотрицательное число  $N$ , которое исследуется на возможность внесения искажений,  $0 \leq N \leq 10^{1000}$ .

### Выходные данные

В первой строке целое положительное число, равное максимальному отклонению от истинного показания при ошибке одного сегмента.

Во второй строке целое положительное число, равное максимальному отклонению от истинного показания при двух ошибках одного сегмента, произошедших в разных разрядах.

Если в каком-то из этих случаев отклонение невозможно, то вместо числа должно стоять слово «NO».

Необходимо рассматривать только ситуации, при которых результат после искажения выглядит корректно. Предполагается, что устройство не отображает незначащие нули в начале числа, так что искажение типа «вместо 888 отобразилось 088» не рассматривается.

### Пример

input.txt

1

output.txt

6

NO

(Пояснение: для единицы возможно только одно искажение, когда она превращается в семёрку. Соответственно, отклонение составляет  $|1-7|=6$ . Искажение двух разрядов невозможно в принципе, так как имеется лишь один разряд.)

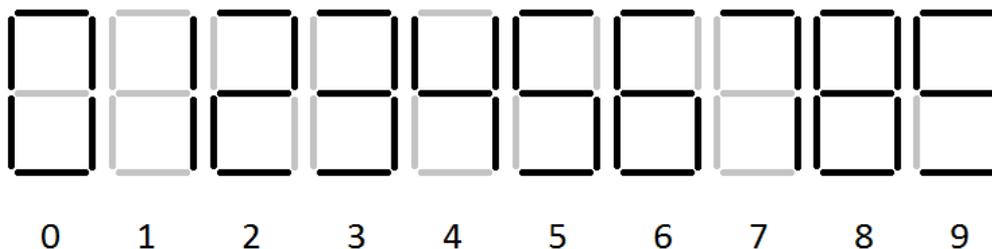
## Задача 7. Прогресс в артиллерии продолжается

**Входной файл:** input.txt

**Выходной файл:** output.txt

**Время на тест:** 1 секунда

На следующем заседании комиссии по перевооружению выступил другой генерал, который наоборот, всячески приветствовал технические новшества, в том числе и новый метод отображения числа с помощью сегментных индикаторов:



Он рассказал, что конструкторы внесли усовершенствование в счётное устройство, и теперь ошибки индикации могут быть легко обнаружены.

По его словам, прибор теперь снабжён контрольным указателем, который показывает последнюю цифру суммы цифр отображаемого числа. Например, для результата 987 будет показана контрольная цифра 4 ( $9+8+7=24$ , контрольная цифра - 4). Причём для большей надёжности это показание производится старым способом, лишённым возможности искажения. О том, что конструкторы сочли такое усовершенствование бредом и внесли его лишь после долгих уговоров, генерал-новатор тактично умолчал.

Генерал-ретроград тут же возразил, что это новшество не защищает от ошибки, и привёл в пример число 17. При односегментной ошибке (то есть в том случае, когда не загорелся один нужный сегмент или, наоборот, загорелся один ненужный) в двух разрядах оно превратится в 71, а показание 8 контрольного счётчика создаст у оператора иллюзию правильности. А если исказятся не два, а больше разрядов, то для ошибок открывается ещё больше возможностей.

Необходимо разработать программу, которая по указанному числу определяет, возможно ли такое его искажение в результате ошибки отображения не более чем одного сегмента в каждом разряде, при котором контрольный указатель не позволяет обнаружить ошибку (искажения, приводящие к появлению незначущих нулей, не рассматриваются).

### Входные данные

В первой строке целое неотрицательное число  $N$ , которое исследуется на возможность внесения искажений,  $0 \leq N \leq 10^{1000}$ .

### Выходные данные

В выходном файле слово «YES» или «NO» в зависимости от того, возможно или нет описанное искажение.

### Пример

input.txt

17

output.txt

YES

## Задача 8. Время раскладывать камни

Входной файл: input.txt

Выходной файл: output.txt

Время на тест: 1 секунда

На столе в ряд расположено  $N$  ящиков, каждый из которых либо пуст, либо содержит несколько камней. За один ход можно сделать одно из следующих действий:

- 1) взять один камень из крайнего ящика и переложить в соседний,
- 2) взять два камня из «некрайнего» ящика и положить их по одному в соседние с ним ящики.

Требуется определить, можно ли достичь таким образом определенного расположения камней в ящиках, и, если можно, вычислить минимальное количество ходов, которое для этого требуется

### Входные данные

В первой строке одно натуральное число  $N$  – количество ящиков,  $2 \leq N \leq 20$ .

Во второй строке  $N$  целых неотрицательных чисел  $a_i$  через пробел, где  $a_i$  – исходное количество камней в  $i$ -ом ящике,  $0 \leq a_i \leq 100$ . Сумма всех  $a_i$  не больше 100.

В третьей строке  $N$  целых неотрицательных чисел  $b_i$  через пробел, где  $b_i$  – желаемое количество камней в  $i$ -ом ящике,  $0 \leq b_i \leq 100$ .

### Выходные данные

Если желаемого распределения камней достичь нельзя, в первой строке вывести «No solution». Иначе, в первой строке вывести одно целое число – минимальное количество ходов, за которое можно достичь желаемого распределения.

### Пример

input.txt

3

1 2 3

3 2 1

output.txt

6

## Задача 9. Медианные последовательности

**Входной файл:** input.txt

**Выходной файл:** output.txt

**Время на тест:** 1 секунда

Будем называть неубывающую целочисленную последовательность  $\{a\}$  длины  $N$  медианной последовательностью (1-го порядка) последовательности  $\{b\}$  длины  $N + 1$ , если для произвольного  $i$  такого, что  $1 \leq i \leq N$ , выполняется равенство

$$a_i = (b_i + b_{i+1}) / 2.$$

Если последовательность  $\{b\}$  в свою очередь является медианной (а значит, в том числе и неубывающей целочисленной) последовательностью некоторой последовательности  $\{c\}$  длины  $N + 2$ , то последовательность  $\{a\}$  будет медианной последовательностью 2-го порядка для последовательности  $\{c\}$ . Аналогичным образом можно определить медианную последовательность  $k$ -го порядка.

Требуется по заданной неубывающей целочисленной последовательности из двух чисел и натуральному числу  $k$  определить, для скольких неубывающих целочисленных последовательностей исходная последовательность является медианной последовательностью  $k$ -го порядка.

### Входные данные

В первой строке два целых числа  $a_1$  и  $a_2$  через пробел,  $-20 \leq a_1 \leq a_2 \leq 20$ , определяющие исходную последовательность длины 2.

Во второй строке натуральное число  $k$ ,  $1 \leq k \leq 20$ , – порядок медианной последовательности  $\{a_1, a_2\}$ .

### Выходные данные

В первой строке одно целое число – количество целочисленных неубывающих последовательностей, для которых последовательность  $\{a_1, a_2\}$  является медианной последовательностью  $k$ -го порядка.

### Пример

input.txt

1 3

2

output.txt

5

### Пояснение к примеру:

Эти 5 последовательностей следующие:

1 1 1 9

0 0 4 4

-5 3 3 3

-1 1 3 5

-2 2 2 6

## Задача 10. Экономная домохозяйка.

Входной файл: input.txt

Выходной файл: output.txt

Время на тест: 1 секунда

Одна семья очень любит домашнее печенье круглой формы. Хозяйке дома приходится постоянно готовить такое печенье, потому она уже набила на этом деле руку и действует быстро и ловко. Она раскатывает тесто в виде прямоугольника и намечает на нем горизонтальные и вертикальные полоски на расстоянии, равном диаметру печенюшки. Затем формой из каждого полученного квадратного кусочка теста она вырезает круг, который и становится печенюшкой, после чего оставшееся тесто она формирует снова в виде некоторого прямоугольника такой же толщины и аналогичным образом вырезает печенье. Процесс заканчивается, когда либо израсходовано все тесто, либо остался кусочек теста, из которого уже нельзя формой вырезать печенье, в какой прямоугольник (с той же толщиной теста) его ни раскатай.

Требуется, зная площадь исходного прямоугольника и радиус печенюшки, определить, сколько печенюшек сможет приготовить хозяйка.

### Входные данные

В первой строке два целых числа через пробел:  $S$  – площадь исходного прямоугольника из теста ( $\text{см}^2$ ),  $R$  – радиус печенюшки (см),  $1 \leq S, R \leq 2 \times 10^9$ .

### Выходные данные

В первой строке одно целое число – максимальное количество печенюшек заданного радиуса, которое можно изготовить из имеющегося теста.

### Пример

input.txt

36 3

output.txt

1

## Задача 11. Древний лабиринт

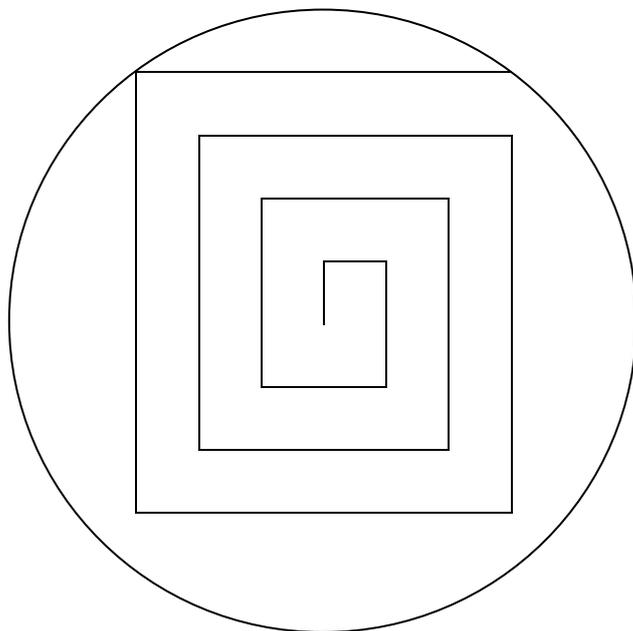
Входной файл: input.txt

Выходной файл: output.txt

Время на тест: 1 секунда

Одной из загадок многотысячелетней истории человеческой цивилизации является загадка древних лабиринтов. Археологи обнаружили их в различных частях света: в Европе, Южной Африки, в Новой Зеландии. Учёных удивило то, что все найденные лабиринты были построены поразительно одинаково.

Все лабиринты представляли собой ломаную, расположенную внутри круга. Она начиналась в центре круга, а далее закручивалась по определенным правилам. Из центра круга ломаная шла строго на север на длину  $h$ . Затем лабиринт поворачивает на восток под прямым углом, причем длина отрезка ломаной оставалась прежней. Потом, поворачивая на юг, длина отрезка ломаной увеличивалась на  $h$ , и, наконец, лабиринт поворачивал на запад, оставляя прежнюю длину отрезка. Далее, ломаная снова шла на север, увеличив на  $h$  длину отрезка, и так далее. Лабиринт заканчивался, когда соответствующий отрезок ломаной пересекал (а не касался) границу круга.



Требуется написать программу, вычисляющую длину лабиринта по начальному шагу  $h$  и радиусу круга  $R$ , и тем самым помочь учёным в изучении загадки этих лабиринтов.

### Входные данные

В первой строке два целых числа через пробел:  $R$  – радиус лабиринта (м),  $1 \leq R \leq 10$ ,  $h$  – шаг лабиринта (см),  $1 \leq h \leq 1000$ .

### Выходные данные

В первой строке выводится одно число – длина лабиринта в метрах с точностью три знака после запятой.

### Пример

input.txt

3 100

output.txt

24.236

## Задача 12. Города Нила.

Входной файл: input.txt

Выходной файл: output.txt

Время на тест: 2 секунды

Известно, что поселения на берегах Нила очень древние, и найдены ещё далеко не все. Один из археологов решил проанализировать обрывочные записи древних купцов и обнаружил, что они достаточно часто упоминают время, требуемое на доставку товара из одного поселения в другое. Зная скорость движения по реке в те далёкие времена, он легко перевёл время в расстояние между поселениями. Учёный знал, что на Ниле в изучаемое время было  $N$  поселений, одно из которых, в устье Нила, было известно. Он принял его положение за точку отсчёта. Из анализа записей о торговле он получил ровно  $N*(N-1)/2$  расстояний между поселениями, т.е. расстояния между любыми двумя поселениями. Ему осталось только сопоставить расстояния парам поселений, и найти расположение поселений вдоль Нила. Потом, правда, он сообразил, что расположение поселений, найденное по указанной информации, может оказаться не единственным, и его энтузиазм несколько угас. Однако, информация лишней не бывает, и он решил всё же заказать программу, выдающую хотя бы одно расположение поселений.

### Входные данные

В первой строке одно целое число  $N$  (число посёлков,  $3 \leq N \leq 25$ ).

Далее в  $N*(N-1)/2$  строк все расстояния между посёлками, по одному целому положительному числу в строке, упорядоченные по возрастанию. Каждое расстояние не превышает 100 000.

### Выходные данные

Расстояния от устья Нила до каждого из  $N$  поселений, по одному в строке, упорядоченные по возрастанию. Расстояние до первого поселения всегда равно 0.

### Пример

*input.txt*

5  
2  
5  
7  
7  
9  
9  
14  
14  
16  
23

*output.txt*

0  
9  
14  
16  
23