

## Задача А. Битва движков

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

В одном институте ведутся вечные споры — какой движок лучше: Unity или Unreal? Чтобы прекратить бессмысленную полемику, Алексей решил использовать движки в тандеме.

Движок Unity может обработать только графический пакет величиной в  $a$  кадров, движок Unreal только в  $b$  кадров. Алексею требуется сделать так, чтобы графический пакет был обработан сначала на Unity, а потом на Unreal. Для этого можно использовать преобразователь, который способен увеличить или уменьшить размер пакета в любое количество раз. Но есть загвоздка:

все получаемые величины после преобразований, а также величины  $a$  и  $b$  не должны совпадать между собой.

Помогите Алексею определить последовательность преобразований графического пакета.

### Формат входных данных

В одной строке через пробел заданы два целых числа  $a$  и  $b$  — размеры графических пакетов ( $1 \leq a, b \leq 10000, a \neq b$ ).

### Формат выходных данных

Выведите порядок действий, каждое из которых в отдельной строке и заданное следующим форматом:

- Выведите  $/d$ , если требуется уменьшить графический пакет в  $d$  раз. Величина  $d$  должна быть делителем текущего размера пакета;
- Выведите  $*m$ , если требуется увеличить графический пакет в  $m$  раз.

Количество действий не должно превышать 1000. Величина получаемых графических пакетов не должна превышать  $10^9$ .

Вы можете вывести любую допустимую последовательность преобразований.

### Примеры

стандартный ввод	стандартный вывод
60 77	*11 *7 *2 /120
12 21	*3 *5 *7 /4 /15
5 8	*2 *4 *8 *16 /640

## Задача В. Я не спал 2000 лет

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Отчаянный программист Ноунейм как всегда всё оставил на последний день. Теперь ему требуется написать  $n$  программ, а по срокам остался всего один день (то есть, 24 часа).

У каждой программы указано время на её разработку (в часах). Определите, сколько максимум программ возможно успеть разработать за один день.

### Формат входных данных

В первой строке одно целое число  $n$  — количество программ ( $8 \leq n \leq 100$ ).

Во второй строке через пробел указаны целые числа  $a_i$  ( $1 \leq i \leq n$ ) — количество часов, необходимое для разработки  $i$ -й программы ( $1 \leq a_i \leq 1000$ ).

### Формат выходных данных

Одно целое число — ответ на поставленную задачу.

### Примеры

стандартный ввод	стандартный вывод
10 4 4 4 1 8 11 3 1 1 1	8
8 3 3 3 3 3 3 3 3	8
8 30 21 28 12 13 40 110 27	1

## Задача С. Пробегка по лестнице

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Женя живет на  $n$  этаже. У него  $k$  энергии.

Судьба распорядилась так, что он живет невысоко. Каждый раз, возвращаясь домой, он осознавал, что не успевает устать, пока поднимается по лестничным пролётам до своего этажа. Поэтому, он решил, что будет продолжать подниматься выше, и только потом возвращаться на свой этаж. Это была бы хорошая тренировка выносливости.

Чтобы подняться на один этаж вверх по лестничным пролётам, требуется потратить  $a$  энергии. С каждым новым подъёмом вверх, стоимость увеличивается на  $a$ , то есть для второго подъёма она будет равна  $2a$  и т.д.

Спуск вниз на один этаж всегда требует  $b$  энергии.

До какого максимального этажа Женя сможет подняться, чтобы у него хватило энергии, в том числе и на возвращение на свой этаж. Подъём начинается с 1 этажа.

Если Жене не хватит энергии добраться до своего этажа, выведите  $-1$ .

### Формат входных данных

В единственной строке через пробел заданы 4 числа —  $n, k, a, b$  ( $1 \leq n, k, a, b \leq 10^{18}$ )

### Формат выходных данных

Одно число — ответ на поставленную задачу.

### Примеры

стандартный ввод	стандартный вывод
2 10 2 1	3
2 14 2 1	4
2 13 2 1	3
1 1000000000000000000 1 1	1414213561

### Замечание

Можно считать, что этажей в доме бесконечно много.

## Задача D. Разборки в космосе

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Один не безызвестный космический институт выпустил свой спутник, чтобы наблюдать за бескрайними просторами вселенной. Руководствуясь международными правилами, спутник может находиться только на геостационарной орбите некоторой планеты.

Но без проблем не обошлось. По соседству расположился спутник-спамер, который забивал весь радиоэфир предложениями купить ягоды «Годжи». Не пробуя вежливо отказать от предложения, вы решили телепортироваться от данного спамера как можно подальше.

В космосе  $n$  планет. Их геостационарные орбиты описаны как длина радиуса  $r_i$  и центр орбиты  $\{x_i y_i z_i\}$ . Сами орбиты находятся в некоторой плоскости, параллельной плоскости XY.

Вам требуется определить, на орбиту какой планеты стоит телепортироваться спутнику института, чтобы оказаться в такой точке пространства, чтобы расстояние между ним и спамером стало максимально большим.

Для простоты будем считать, что спутники являются материальными точками.

### Формат входных данных

Спутник-спамер всегда находится в координатах  $\{0\ 0\ 0\}$ .

В первой строке задано одно целое число  $n$  — количество планет ( $1 \leq n \leq 10^5$ ).

В  $i + 1$  строке ( $1 \leq i \leq n$ ) заданы параметры геостационарной орбиты  $i$ -й планеты. Данные указаны в следующем формате:

в одной строке через пробел заданы 4 целых числа  $r, x, y, z$  — радиус орбиты и координаты её центра ( $1 \leq r \leq 1000$ ).

Все значения координат находятся в диапазоне от -1000 до 1000.

### Формат выходных данных

Одно целое число от 1 до  $n$  — номер планеты, на чьей геостационарной орбите окажется спутник института после телепортации.

Если в качества ответа подходит несколько номеров, выведите наименьший.

### Пример

стандартный ввод	стандартный вывод
4 2 -3 1 0 1 5 -1 1 5 -1 -1 4 1 5 4 0	3

## Задача Е. Бесконечные вычисления

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Толе-Карпу приснился кошмар, где он до бесконечности делил число  $n$  на 2. А потом, он вдруг осознал, что его интересует последняя значимая цифра, которая получилась у числа.

Помогите Толе-Карпу пережить данный сон, сообщите ему данную цифру.

### Формат входных данных

В одной строке через пробел заданы 2 целых числа  $n$  и  $k$  — изначальное число, которое делил Толя-Карп и количество делений на 2 ( $1 \leq n \leq 1000, 0 \leq k \leq 100$ ).

### Формат выходных данных

Выведите одну цифру — ответ на поставленную задачу.

### Примеры

стандартный ввод	стандартный вывод
20 2	5
4 3	5
34 1	7

## Задача F. Воспитательница

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 256 мегабайт

### Это интерактивная задача

Сначала выдаётся множество допустимых слов. Описывается оно числом  $n$  — количество слов во множестве, далее идут сами слова.

Вам требуется одолеть Марию Ивановну:

Марья Ивановна загадывает одно слово из множества. Вы можете создать запрос — один строчный символ латинского алфавита. Марья Ивановна ответит, сколько раз в загаданном слове встречается данный символ.

Ваша задача — отгадать загаданное слово не более, чем за **3** запроса.

### Формат входных данных

В первой строке задано число  $n$  — количество слов во множестве ( $1 \leq n \leq 100$ ).

В следующих  $n$  строках указаны слова множества, по одному в каждой строке (длина каждого из них не превосходит 10).

В  $n + 2$  строке задано целое число  $T$  ( $1 \leq T \leq 10^4$ ) — количество раундов, которые нужно сыграть.

Гарантируется, слова во множестве подобраны так, что любое слово однозначно можно определить при помощи 3 запросов.

### Протокол взаимодействия

После считывания входных данных, тестирующая система готова отвечать на запросы вашей программы.

В каждой партии взаимодействие начинает ваша программа запросом строчного латинского символа  $C$ . Запрос задаётся в виде «?  $C$ ». После запроса вы получите ответ — сколько раз встречается символ  $C$  в загаданном слове. Данный запрос можно повторить до 3 раз.

Если вы готовы сообщить загаданное слово  $A$ , выведите строку «!  $A$ ». Если по итогу программа жюри выведет число 1, то предложенный ответ оказался верным и начнётся следующий раунд игры. Если программа жюри выведет число  $-1$ , то ответ неверный.

В случае, если на ваш ответ программа жюри ответила  $-1$ , программа должна немедленно завершиться, в противном случае есть шанс получить вместо Wrong Answer другую ошибку.

### Пример

стандартный ввод	стандартный вывод
5	? a
ab	
bc	? b
cd	
ad	? c
aad	
1	! cd
0	
0	
1	
1	

## Замечание

Для корректной работы программы после каждой операции вывода запроса или вывода ответа требуется выводить символ переноса строки, а также очищать буфер вывода, то есть делать следующие операции:

- В языке Pascal: `flush(output);`
- В C/C++: `fflush(stdout)` или `cout.flush();`
- В Java: `System.out.flush();`
- В Python: `sys.stdout.flush()` из библиотеки `sys`;
- В C#: `Console.Out.Flush();`

## Задача G. Инскрипция

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

В популярной игре Инскрипция есть сложный босс. Битва с ним происходит следующим образом:

1. Вам выдаётся  $k$  карточек с цифрами от 0 до 9;
2. Вы образуете десятичное число из данных карточек путем конкатенации указанных на них значений. Из карточек 2 и 3 можно образовать числа 23 и 32;
3. Боссу наносится такое значение урона, которое получается у созданного вами десятичного числа при взятии остатка от деления на 15.

Порой, наносить много урона не выгодно, босс может перейти во вторую фазу раньше времени. У вас есть артефакт, который случайным образом образует десятичное число из карточек и при этом, позволяет вам повторно совершить ещё один ход. Но использовать его рискованно, поэтому вы прежде всего решили найти среднее арифметическое значение урона при составлении десятичных чисел из заданных карточек.

Напишите программу, которая находит данное значение.

### Формат входных данных

В первой строке одно целое число  $k$  ( $1 \leq k \leq 10^5$ ) — количество карточек.

Во второй строке заданы через пробел  $k$  цифр со значениями от 0 до 9.

### Формат выходных данных

Одно вещественное число — ответ на поставленную задачу с точность не хуже  $10^{-6}$ .

### Примеры

стандартный ввод	стандартный вывод
2 1 5	3
3 0 0 1	7
4 9 8 9 8	8.5

### Замечание

Артефакт случайно берёт любую имеющуюся карточку и ставит её на первую позицию. Далее берёт вторую карточку из оставшихся и так далее.

Во втором примере вы можете составить следующие числа: 001, 001, 010, 010, 100, 100. Итого, среднее арифметическое значение урона будет вычислено как  $42/6 = 7$ .



## Задача Н. Все очень просто

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    1 секунда  
Ограничение по памяти:      256 мегабайт

По небу летит солнечный остров массой  $a$  тонн и воздушный замок массой  $b$  тонн.

Вы находитесь в замке и можете утяжелить его, добавив облако любой массы.

Впереди летит Жар-Птица, которая пикирует на тот объект, которой является более лёгким. Вы хотите сделать так, чтобы Жар-Птица не проткнула своим клювом ваш замок.

Какой минимальной массы облако требуется добавить, чтобы Жар-Птица выбрала в качестве цели солнечный остров? В случае равенства масс, Жар-Птица выберет замок.

### Формат входных данных

В одной строке через пробел заданы два целых числа — масса солнечного острова  $a$  и воздушного замка  $b$ , указанная в тоннах ( $1 \leq a, b \leq 1000$ ).

### Формат выходных данных

Одно целое число — масса облака в тоннах.

### Примеры

стандартный ввод	стандартный вывод
100 50	51
5 1	5

## Задача I. Расписание занятий

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Вам выдана таблица о расписании нового направления ИКИТ. В нём указаны разделы дисциплин и соответствующее им количество зачётных единиц (ЗЕ) в виде таблицы. Таблица была экспортирована в формате csv с разделителем «запятая».

Но вот проблема, колонтитул у таблицы не сохранился. Вам требуется посчитать суммарное количество ЗЕ всех разделов или сообщить о том, что данные невалидны.

Раздел может состоять из подразделов, в этом случае, после указания названия самого раздела следует название подраздела через точку. Подраздел может состоять из своих подразделов, увеличивая вложенность. Суммарно, количество ЗЕ у подразделов должно быть равно количеству ЗЕ самого раздела. Раздел и его подразделы идут непрерывно друг за другом.

### Формат входных данных

В первой строке одно число  $n$  — количество строк в таблице формата csv ( $1 \leq n \leq 1000$ ).

В каждой из следующих  $n$  строк указаны данные — латинские символы, цифры, знак точка и запятая. Запятая является разделителем между столбцами и не присутствует в названии разделов. Количество запятых в каждой строке одинаковое.

В одном из столбцов присутствуют только целые числа  $a_i$  ( $1 \leq a_i \leq 1000$ ), которые указывают на количество ЗЕ. Гарантируется, в таблице нет второго такого столбца, в котором присутствовали бы только целые числа.

Название раздела не повторяется и может присутствовать в любом столбце. Подраздел указывается в том же столбце, что и раздел, в следующем ряду. В каждой ряду таблицы заполненными являются только две ячейки (название и ЗЕ).

Максимальная длина каждой строки - 1000 символов.

### Формат выходных данных

Данные таблицы является невалидными, если возникает ситуация, когда сумма ЗЕ подразделов не равна кол-ву ЗЕ у самого раздела. В этом случае, выведите  $-1$ . Иначе, выведите суммарное количество ЗЕ.

## Примеры

стандартный ввод	стандартный вывод
5 GameCreate,,12,,, ,Math,8,,, ,Math.Profi,6,,, ,Math.Starter,2,,, ,,1,,,PhysicalCulture	21
5 Streaming,,,4 Streaming.SFU,,,2 Streaming.ISIT,,,2 Streaming.ISIT.IS,,,2 Streaming.ISIT.PI,,,2	-1
12 a,18 a.1,6 a.1.1,3 a.1.2,3 a.2,6 a.2.1,3 a.2.2,3 a.3,6 a.3.1,6 a.3.1.1,6 b,10 1,100	128

## Замечание

Во 2 примере у раздела «Streaming.ISIT» всего 2 ЗЕ, но у его подразделов суммарно вышло 4 ЗЕ.

## Задача J. Культ суперкомпьютера

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Для того, чтобы подпитывать вычислительными мощностями суперкомпьютер, рядом с ним работает  $n$  студенческих ноутбуков. Подключены они между собой так, что образуется окружность (смотрите примечание).

Чтобы совершить обряд подпитки, ноутбуки сначала загадывают некоторые натуральные числа, а потом отсылают их своим соседям. Всё это происходит одновременно. Каждый ноутбук складывает пришедшие к нему числа и выводит их на экран. Если по итогу, на всех экранах будут выведены нечётные числа, обряд считается завершённым успешно.

Помогите определить, возможно ли завершить обряд успешно, если подключено  $n$  ноутбуков.

### Формат входных данных

Одно целое число  $n$  — количество ноутбуков ( $3 \leq n \leq 10^5$ ).

### Формат выходных данных

Выведите «YES», если обряд может завершиться успешно. Иначе, выведите «NO».

### Примеры

стандартный ввод	стандартный вывод
3	NO
4	YES
5	NO
6	NO

### Замечание

На изображения показано: слева — какие числа сгенерировали ноутбуки, справа — полученные результаты сумм.

