

Задача А. Сладкоежка

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Любитель сладостей Жора в свободное от учебы время зашел в кафе, где продают его любимые пирожные и фруктовый сок. Одно пирожное стоит A рублей, а один стакан фруктового сока – B рублей. Жора сильно голоден и хочет потратить на еду все свои деньги – S рублей. Сколько пирожных и стаканов сока может купить Жора?

Формат входных данных

В единственной строке задаются три натуральных числа S , A и B , не превышающих 10^6 .

Формат выходных данных

Выведите два числа – количество пирожных и количество стаканов сока, которые может купить Жора. Если существует несколько решений, выберите из них то, при котором Жора купит больше пирожных. Если решений не существует, выведите одно число -1 .

Примеры

<code>standard input</code>	<code>standard output</code>
20 6 7	1 2
21 7 1	3 0
1 7 11	-1

Задача В. Футбольный матч

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В рамках торжественных мероприятий, посвященных празднованию государственных праздников двух дружественных республик (Десятично-Натуральной Республики и Линейно Независимой Республики), был проведен товарищеский матч по футболу между сборными этих двух государств.

Молодому, но очень перспективному спортивному журналисту Григорию было поручено подготовить статью об этом матче в газету. К сожалению, из-за большого ажиотажа Григорий не смог попасть на стадион и посмотреть матч, однако ему удалось раздобыть протокол матча, в котором были указаны номера игроков обеих команд, сыгравших в этом матче, а также номера игроков, отличившихся голами в этом матче, однако без указания команды, за которую этот игрок выступает.

Помогите Григорию узнать, с каким результатом мог закончиться матч, в предположении, что не было автоголов (т.е. ни один из игроков не забивал в свои ворота).

Формат входных данных

В первой строке задаются 11 целых чисел в диапазоне от 1 до 99 – номера игроков сборной Десятично-Натуральной Республики, принявших участие в матче. Во второй строке аналогичным образом задаются номера игроков сборной Линейно Независимой Республики. В третьей строке записано целое число K ($0 \leq K \leq 10$) – количество голов, забитых в матче. В четвертой строке задаются K целых чисел. i -ое из этих чисел определяет номер игрока забившего i -ый гол в матче.

Гарантируется, что нет двух игроков с одинаковыми номерами в одной сборной, а голы забивали только игроки, принимающие участие в матче.

Формат выходных данных

Выведите все возможные (с учетом имеющейся информации) результаты матча в формате “ $x:y$ ”, где x и y – количество мячей, забитых игроками сборной Десятично-Натуральной Республики и Линейно Независимой Республики соответственно. Ни один из результатов не должен быть выведен более одного раза. Каждый результат должен быть записан в отдельной строке.

Пример

standard input	standard output
1 2 5 9 10 11 17 25 33 49 77	2:2
1 3 4 10 11 12 17 28 44 69 92	3:1
4	
9 10 4 25	

Задача С. Локомотивное программирование

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Институт железнодорожного транспорта разработал прототип нового локомотива с программным управлением. Программа для него представляет собой последовательность пар чисел. Первое число пары является целым в диапазоне от -1 до 1 и определяет режим работы механизмов локомотива, а второе – положительное вещественное, определяющее время работы в соответствующем режиме.

В режиме 0 локомотив движется равномерно со скоростью, которую имел в момент включения этого режима.

В режиме 1 локомотив разгоняется (движется равноускоренно) с ускорением a_+ , а в режиме -1 – тормозит (движется равнозамедленно) с ускорением a_- .

При этом конструктивные особенности локомотива накладывают следующие ограничения:

1. Во время движения скорость локомотива должна все время сохраняться в диапазоне от 0 до v_{max} .
2. До переключения в режим 1 после завершения работы в режиме -1 (или наоборот) должно пройти время, не меньшее t_0 .

Если эти требования не будут соблюдены, локомотив может выйти из строя, чего нельзя допускать.

Требуется запрограммировать локомотив, стоящий в начальной момент времени, таким образом, чтобы он проехал расстояние L и остановился в конце пути. Общее время движения при этом должно быть минимально возможным.

Формат входных данных

В единственной строке задаются пять вещественных чисел L , a_+ , a_- , v_{max} , t_0 . Величины L , a_+ , a_- , v_{max} – положительные, t_0 – неотрицательная. Все числа не превышают 10^3 и имеют не более 3 знаков после десятичной точки.

Формат выходных данных

В первой строке выведите размер K оптимальной программы – количество пар чисел в последовательности. В каждой из последующих K строк должны быть выведены по два числа – режим работы и время работы в этом режиме (с точностью не ниже 10^{-6}).

Примеры

standard input	standard output
28.0 2.0 1.0 4.0 4.5	3 1 2.000000 0 5.000000 -1 4.000000
10 0.9 0.9 5 0	2 1 3.333333 -1 3.333333

Задача D. Ребус на максимум

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Если вы увлекаетесь математическими головоломками, то вероятно сталкивались уже с числовыми ребусами. Числовой ребус представляет из себя корректный арифметический пример, в котором все цифры были заменены буквами (одинаковые цифры заменяются одинаковыми буквами, а различные – различными). От решателя ребуса обычно требуется восстановить полную запись примера, то есть выяснить, какой цифре соответствует каждая буква в ребусе.

В данной задаче указанным выше образом закодированы N чисел, которые будут суммироваться. Но в отличие от стандартного числового ребуса результат сложения неизвестен. Ваша задача – подобрать такое соответствие букв цифрам, чтобы сумма чисел была максимально возможной.

Формат входных данных

В первой строке задается целое число N ($1 \leq N \leq 10^5$), определяющее количество зашифрованных чисел. В каждой из последующих N строк задается закодированное число, состоящее не более чем из 12 заглавных букв латинского алфавита.

Формат выходных данных

В единственной строке выведите максимальную сумму, которой можно добиться при корректной замене символов на десятичные цифры. Если невозможно заменить корректно символы на цифры, выведите одно число -1 .

Примеры

standard input	standard output
6 TO BE OR NOT TO BE	1357
10 AA BB CC DD EE FF GG HH II JJ	-1

Замечание

Во втором примере есть 10 различных букв. Одна из них должна очевидно соответствовать цифре 0. Но тогда среди заданных строк будет такая, которая имеет расшифровку “00”, что не является корректной десятичной записью ни для какого целого числа.

Задача Е. Декодирование UTF-8

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

UTF-8 является в настоящее время одним из наиболее распространенных стандартов для кодирования текста, позволяющим компактно хранить и передавать Unicode-символы, используя переменное количество байтов (от 1 до 6) и обеспечивающим обратную совместимость с 7-битной кодировкой ASCII.

Длина представления Unicode-символа в кодировке UTF-8 (количество байтов) определяется в зависимости от количества значащих битов в двоичном представлении Unicode-кода этого символа в соответствии со следующей таблицей:

Число значащих битов	Длина	Шаблон UTF-8
0..7	1	0xxxxxxx
8..11	2	110xxxxx 10xxxxxx
12..16	3	1110xxxx 10xxxxxx 10xxxxxx
17..21	4	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
22..26	5	111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
27..31	6	1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx

Таким образом, однобайтовые представления используются только для символов ASCII (с кодами от 0 до 127). Старший бит для этих символов всегда равен 0.

Символы с кодами, большими 127, представляются последовательностями из нескольких байтов (один ведущий байт и один или несколько продолжающих байтов). Ведущий байт имеет две или более единицы в старших разрядах, а продолжающие байты содержат последовательность 10 в двух старших битах.

Количество единиц в старших разрядах ведущего байта многобайтовой последовательности определяет количество байтов в ней. Остальные биты представления (обозначенные символом “x” в приведенных выше шаблонах) используются для битов Unicode-кода кодируемого символа, дополняя его при необходимости нулями в старших разрядах. Старшие биты Unicode-кода записываются в ведущем байте, а остальные – последовательно в продолжающих байтах.

Стандарт определяет, что при правильном кодировании используется минимальное количество байтов, необходимых для записи кода символа. Более длинное представление считается недопустимым. Это правило обеспечивает взаимно однозначное соответствие между символами и допустимыми UTF8-представлениями. Это гарантирует, что сравнения строк и поисковые запросы будут корректно определены.

В настоящее время стандарт RFC2639 в целях обеспечения совместимости с UTF-16 не допускает использование 5- и 6-байтовых UTF-8 представлений (а также некоторых 4-байтовых), однако в этой задаче мы будем считать эти представления допустимыми.

Ваша задача – написать декодер UTF-8, который по заданной UTF-8 последовательности выдаст последовательность кодов соответствующих символов.

Формат входных данных

В первой строке задается целое число N ($1 \leq N \leq 10^5$). Вторая строка содержит значения N байтов, представленных в виде двухзначных шестнадцатеричных чисел (в диапазоне от 00 до FF). Символы 0-9 соответствуют цифрам от нуля до девяти, а A, B, C, D, E, F представляют цифры от десяти до пятнадцати. Значения байтов отделяются друг от друга одним пробелом.

Формат выходных данных

Если последовательность из N байтов может быть успешно декодирована в последовательность L Unicode-символов, выведите в первой строке число L , а во второй – L чисел, определяющих коды соответствующих символов (в стандартной десятичной записи без ведущих нулей).

Если же заданная последовательность не может быть декодирована, выведите сообщение “Epic Fail”.

Примеры

standard input	standard output
1 24	1 36
2 C2 A2	1 162
3 E2 82 AC	1 8364
4 F0 90 8D 88	1 66376
4 F0 80 80 AA	Epic Fail
4 00 01 02 03	4 0 1 2 3

Задача F. Доставка пиццы

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В начале самой длинной улицы города находится пиццерия “Благородный Дон”, которая выполняет заказы на приготовление и доставку пиццы. Вдоль этой улицы расположены дома с номерами от 1 до 10^9 . Расстояния между всеми парами соседних домов (с номерами, отличающимися на 1), а также расстояние от пиццерии до дома номер 1 одинаковы, и для того, чтобы преодолеть это расстояние на автомобиле, требуется одна единица объема топлива.

В пиццерию поступило N заказов, которые выполнил шеф-повар, и теперь их необходимо доставить по назначению. В пиццерии есть достаточно курьеров с автомобилями. Каждому курьеру (кроме последнего) шеф-повар выдает случайным образом по M пицц с адресами, куда их необходимо доставить, последнему курьеру может остаться для доставки от 1 до M пицц. Каждый курьер садится в свой автомобиль и едет по улице в сторону увеличения номеров домов до тех пор пока не достигнет какого-либо дома, в который необходимо доставить одну или несколько из выданных ему пицц. Тогда он останавливается, отдает все пиццы, заказанные жителями соответствующего дома, после чего возвращается в машину и, если еще не все заказы доставлены, продолжает свой путь.

Определите, какое суммарное количество топлива может потребоваться для того, чтобы доставить все заказы.

Формат входных данных

В первой строке задаются два целых числа N и M ($1 \leq M, N \leq 10^5$). Во второй строке задаются N натуральных чисел, i -ое из которых определяет номер домов, из которого поступил i -ый заказ. Все числа не превышают 10^9 .

Формат выходных данных

Выведите два числа. Первое из них – это минимальный суммарный объем топлива, который потребуются курьерам для доставки заказов, если шеф-повар будет выдавать курьерам задания оптимальным образом. А второе – минимальный суммарный объем топлива, которого будет достаточно даже в том случае, если шеф-повар будет выдавать задания курьерам наихудшим из возможных вариантов.

Примеры

standard input	standard output
3 2 1 2 3	4 5
6 3 2 5 8 3 5 7	13 15

Задача G. Короткая спичка

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

N студентов университета отмечали в общежитии день факультета. В разгар праздника оказалось, что закончился весь фруктовый сок. Докупить сока не было бы проблемой – магазин находится не очень далеко от общежития, но никто из студентов не хотел идти. Поэтому они решили бросить жребий.

Достав из спичечного коробка N спичек и надломив одну из них, N -ый студент тщательно перемешал их и, выстроив в ряд, взял в руку таким образом, чтобы остальные не видели длину спичек. Далее все студенты с 1 по $N - 1$ по очереди вытягивают по одной спичке. Как только студент вытянул спичку, остальные спички сдвигаются к началу, чтобы не нарушалась непрерывность ряда. После того, как все вытянули по спичке, остается одна спичка, которая достается студенту N .

Определите, кому досталась короткая спичка.

Формат входных данных

В первой строке задается целое число N ($1 \leq N \leq 2 \cdot 10^5$). Вторая строка содержит N символов, определяющих длины соответствующих спичек: `l` соответствует длинной спичке, а `s` – короткой. В третьей строке задаются $N - 1$ чисел, i -ое из которых лежит в диапазоне от 1 до $N + 1 - i$ и определяет номер спички в оставшемся к тому моменту ряду, которую вытаскивает студент i .

Формат выходных данных

Выведите одно число – номер студента, которому достанется короткая спичка.

Примеры

<code>standard input</code>	<code>standard output</code>
5 llsll 4 1 2 1	3
4 slll 4 3 2	4

Задача Н. Покраска роз

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Все знают о том, что в саду Червонной Королевы рос ряд розовых кустарников. Однако мало кто догадывается о том, что этот ряд был бесконечным в обе стороны. Таким образом, каждый куст был пронумерован целым числом, и для каждого целого числа был ровно один куст с соответствующим номером. К сожалению, карты-садовники по ошибке посадили вместо красных роз белые. И теперь, когда они выросли и расцвели, садовникам, чтобы избежать гнева Королевы, предстоит решить сложнейшую задачу – перекрасить розы в красный цвет.

Однако нелегко перекрасить бесконечное количество кустов, поэтому N садовников договорились действовать следующим образом. Первый садовник покрасит каждый k_1 -ый куст, то есть им будут перекрашены в красный цвет все кусты с номерами, делящимися на число k_1 . Второй будет красить каждый k_2 -й куст, и так далее.

После того, как покраска была завершена, Червонная Королева вышла в свой сад и прошлась от куста с номером a до куста с номером b включительно. Определите, сколько кустарников с белыми розами она увидела на своем пути.

Формат входных данных

В первой строке задается целое число N ($0 \leq N \leq 20$). Во второй строке задается N натуральных чисел k_1, k_2, \dots, k_N , не превосходящих 10^9 . В третьей строке задаются два целых числа a, b ($-10^{18} \leq a \leq b \leq 10^{18}$).

Формат выходных данных

Выведите одно число – количество белых роз с номерами в диапазоне от a до b .

Примеры

standard input	standard output
2 3 5 1 10	5
3 4 13 3 24 28	1

Задача I. Уверенная ставка

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В финал одного из самых престижных теннисных турниров Двоичной Республики вышли два известных теннисиста – Сипли́сис и Паска́лис. Есть N букмекерских контор, которые принимают ставки на их поединок. Разные конторы по-разному оценивают шансы теннисистов на победу, и поэтому могут предлагать разные коэффициенты на один и тот же исход. Коэффициент x обозначает, что если вы поставите 1 байт-коин на соответствующий исход и финал завершится с этим исходом, букмекерская контора выплатит вам x байт-коинов. Конечно же, если финал завершится с иным исходом, вы не получите ничего. Заметьте, что в обоих случаях поставленный 1 байт-коин вам не возвращается.

Вам известно, что i -ая букмекерская контора предлагает на победу Сиплисиса коэффициент a_i , а на победу Паскалисиса – b_i . Вы можете ставить на любые исходы у любых букмекеров. Вы можете даже поставить на оба исхода у одного букмекера. Однако каждая ставка должна быть равна в точности 1 байт-коину, и нельзя ставить многократно на один и тот же исход у одного букмекера.

Конечно же, вы хотели бы сделать такие ставки, чтобы ваша прибыль была максимально возможной независимо от того, чьей победой завершится финал.

Определите максимальный гарантированный выигрыш, которого можно добиться, если выполнить ставки оптимальным способом.

Формат входных данных

В первой строке задается натуральное число N ($1 \leq N \leq 10^5$). В каждой из последующих N строк задаются по два вещественных числа a_i и b_i . Эти числа лежат в диапазоне от 0.1 до 1001 и имеют не более 4 знаков после десятичной точки.

Формат выходных данных

Выведите максимальную гарантированную прибыль с точностью до 4 знаков после десятичной точки.

Пример

standard input	standard output
4	0.5000
1.4 3.7	
1.2 2	
1.6 1.4	
1.9 1.5	

Пояснение к примеру

Оптимальной стратегией будет поставить на победу Сиплисиса у третьего и четвертого букмекера, а также на победу Паскалисиса у первого букмекера, потратив 3 байт-коина. В случае победы Сиплисиса выигрыш составит 3.5 байт-коина, а в случае победы Паскалисиса – 3.7 байт-коина. Таким образом, мы гарантированно получим прибыль, не меньшую 0.5 байт-коина. Если делать другие ставки, то конечно при благоприятном исходе можно получить и большую прибыль, но нельзя гарантировать, что большая прибыль будет при любом исходе.

Задача J. Вскрытие сейфа

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Опытный медвежатник Мартин за свою профессиональную карьеру вскрыл не один сейф, но на этот раз ему пришлось столкнуться с сейфом очень необычной конструкции.

На этом сейфе расположены N замков в ряд. Каждый замок может находиться в одном из двух состояний: открытом или закрытом. Состояние первого замка можно изменять независимо от состояния других, а состояние i -ого (при $i > 1$) можно изменять только тогда, когда $(i - 1)$ -ый закрыт, а все замки с номерами, меньшими $i - 1$, открыты.

Мартин передал вам начальное состояние всех замков и просит определить, какое минимальное количество действий (изменений состояния одного из замков) потребуется, чтобы открыть сейф (т.е. добиться того, чтобы все замки оказались в открытом состоянии).

Формат входных данных

В первой строке задается одно натуральное число N ($1 \leq N \leq 5000$). Во второй строке задается N чисел, каждое из которых определяет состояние соответствующего замка (0 – открыт, 1 – закрыт).

Формат выходных данных

Выведите одно целое число – количество действий, необходимых для вскрытия сейфа.

Примеры

<code>standard input</code>	<code>standard output</code>
4 1 0 1 0	6
2 1 1	2

Задача К. Создание треугольника

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Есть стальная проволока длиной l метров. Случайным образом на этой проволоке выбираются две точки и в этих точках проволока разрезается. В результате проволока распадается на три куска.

Определите вероятность того, что левый кусок будет иметь длину в диапазоне от a_1 до a_2 , средний – от b_1 до b_2 , правый – от c_1 до c_2 , и при этом из этих кусков можно будет составить треугольник.

Выбор каждой точки подчиняется равномерному закону распределения, и выбор одной точки выполняется независимо от выбора другой.

Формат входных данных

В первой строке задается одно число l ($1 \leq l \leq 10^6$). Во второй строке задается два числа a_1 и a_2 ($0 \leq a_1 \leq a_2 \leq l$), в третьей – b_1 и b_2 ($0 \leq b_1 \leq b_2 \leq l$), в четвертой – c_1 и c_2 ($0 \leq c_1 \leq c_2 \leq l$). Все числа – целые.

Формат выходных данных

Выведите одно число – искомую вероятность с точностью не ниже 10^{-6} .

Примеры

standard input	standard output
10 0 10 0 10 0 10	0.25
20 5 15 0 10 5 15	0.125