

Задача А. Ночное программирование

Имя входного файла: *стандартный ввод* или `input.txt`
Имя выходного файла: *стандартный вывод* или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Многие студенты программируют по ночам. Предположим, за один час студент напишет m строчек кода. Сколько строчек кода будет написано всего, если программирование начинается после полуночи и гарантированно заканчивается в тот же день?

Формат входных данных

В единственной строке содержатся три целых числа: m — количество строчек кода в час ($0 \leq m \leq 1000$), h_1 — час, с которого начинается программирование и h_2 — час, в котором заканчивается ($0 \leq h_1 \leq h_2 \leq 23$).

Формат выходных данных

Выведите единственное число, равное количеству написанных студентом строчек кода.

Примеры

ввод	вывод
20 1 5	80
50 2 5	150

Задача В. Квадрат

Имя входного файла: *стандартный ввод* или `input.txt`
Имя выходного файла: *стандартный вывод* или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дана прямоугольная декартова плоскость. На ней нужно расположить квадрат заданной площади S таким образом, чтобы вершины квадрата располагались в точках с целочисленными координатами, или определить, что это невозможно.

Формат входных данных

В единственной строке содержится одно целое число S ($1 \leq S \leq 10^{12}$) — площадь квадрата.

Формат выходных данных

Если на плоскости возможно расположить квадрат заданным способом, то в первой строке выведите «YES», а в следующих четырёх строках выведите x_i и y_i ($|x_i|, |y_i| \leq 10^9$) — координаты i -й вершины квадрата.

Вершины допускается выводить в произвольном порядке.

Если же подобным образом квадрат расположить невозможно, то в единственной строке нужно вывести «NO».

Примеры

ВВОД	ВЫВОД
4	YES 1 1 -1 -1 1 -1 -1 1
2	YES 1 0 0 1 1 2 2 1
3	NO

Задача С. CS:GO

Имя входного файла: *стандартный ввод* или `input.txt`
Имя выходного файла: *стандартный вывод* или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Поликарп смотрит трансляцию по «Computer Science: Glamorous Optimizations» — новой дисциплине спортивного программирования, сокращённо **CS:GO**.

В матч по **CS:GO** играют две команды, которые соревнуются между собой, применяя не только знания об алгоритмах и структурах данных, но и используя самые невероятные умения в программировании. Матч состоит из раундов, в каждом из которых побеждает какая-то команда.

Победителем матча считается команда, которая первой выигрывает 16 раундов.

В случае, если по истечении 30 раундов команды имеют равное количество выигранных раундов, то назначается серия из 6 дополнительных раундов — овертайм. Победителем в овертайме является команда, которая первая выигрывает 4 раунда. В случае ничейного счёта в овертайме, назначается новый овертайм, и так до определения победителя.

Уже поздно и Поликарп очень хочет спать, поэтому по заданному счёту, который он видит на трансляции, ему интересно минимальное количество раундов, которое должны сыграть команды, для выявления победителя матча.

Помогите ему в этом, чтобы он определился идти ему спать или пытаться досмотреть матч.

Формат входных данных

В единственной строке содержатся два целых числа a и b — текущий счёт ($0 \leq a, b \leq 15^{15}$).

Формат выходных данных

Если Поликарп настолько сонный, что видит невозможный счёт, выведите «`votesleep`». Иначе, выведите единственное число — минимальное количество раундов до определения победителя.

Примеры

ВВОД	ВЫВОД
8 7	8
16 9	0
17 9	votesleep
17 18	1

Задача D. Тайный геном

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

Это интерактивная задача.

Перед вами стоит цель разгадать неизвестный геном. Для простоты будем считать, что он является последовательностью, состоящей из символов «AGCT» и имеющей длину от 1 до 300.

Для любой последовательности символов s разрешается провести эксперимент, отвечающий на вопрос, существует ли в неизвестном геноме **подпоследовательность подряд идущих** символов, совпадающая с s .

Протокол взаимодействия

Каждый вопрос требуется выводить в отдельной строке в виде «? s », где s — непустая последовательность, состоящая из символов «AGCT».

На каждый такой вопрос в отдельной строке будет выведен символ «+», если s входит в геном, и «-», если не входит.

Вы должны задать **не более 850 запросов**, после чего вывести в отдельной строке ответ в виде «! t », где t — разгаданный геном, и завершить работу программы.

Пример

ВВОД	ВЫВОД
+	? CAT
-	? GTA
+	? TAG
-	? CATAG
	! CATTAG

Замечание

Для корректного взаимодействия после каждой операции вывода данных вам необходимо вывести **перенос строки**, а также **очищать буфер вывода**, то есть делать следующие операции:

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

Задача Е. Опытный робототехник

Имя входного файла: *стандартный ввод* или `input.txt`
Имя выходного файла: *стандартный вывод* или `output.txt`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Вова – начинающий инженер, в школе он занимается в кружке робототехники уже второй год. За год он значительно обновил навигацию своего робота и собирается снова его тестировать.

Он подготовил поле, являющееся выпуклым многоугольником. На поле расчерчены линии, разбивающие плоскость на клетки со стороной 1 сантиметр и образующие декартову систему координат. Робот на этом поле в каждый момент времени может перемещаться только в одном из четырёх направлений, параллельных линиям сетки. Скорость робота составляет один сантиметр в секунду.

Вова собирается проверить заложенную в робота программу достижения цели. За один тестовый запуск Вова выбирает две случайные точки внутри поля: в первую он помещает робота, а во вторую роботу нужно попасть. Робот будет двигаться так, что это займёт минимальное время.

Точки в многоугольнике Вова выбирает каждый раз равновероятно и независимо.

Будем считать, что робот является материальной точкой, и что смена направления, разгон и торможение происходят мгновенно.

Вова собирается делать такой тест сегодня много раз, поэтому ему интересно, сколько понадобится энергии для робота, которая зависит от времени перемещения. Определите математическое ожидание времени, необходимое роботу для достижения цели за один тестовый запуск.

Формат входных данных

Первая строка содержит целое число n — количество вершин многоугольника.

Вторая строка содержит n целых чисел x_i , разделённых пробелами.

Третья строка содержит n целых чисел y_i , разделённых пробелами.

$3 \leq n \leq 30\,000$; $|x_i|, |y_i| \leq 10^9$.

Вершины многоугольника заданы в порядке обхода против часовой стрелки.

Никакие три подряд идущие точки многоугольника не лежат на одной прямой.

Формат выходных данных

В единственную строку выведите ожидаемое время. Выведенный ответ будет считаться верным, если абсолютная или относительная погрешность не превосходит 10^{-9} .

Примеры

ВВОД	ВЫВОД
4 0 4 4 0 0 0 2 2	2
4 0 1 0 -1 -1 0 1 0	0.9333333333
3 0 1 3 3 0 2	1.4222222222
5 1 4 6 4 0 0 1 4 5 2	2.8782787990904536169885

Задача F. Центр перехода в СУНЦ

Имя входного файла: *стандартный ввод* или `input.txt`
Имя выходного файла: *стандартный вывод* или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

«ЦПС — это вообще-то центр перехода в СУНЦ МГУ, а не центр профессионального самоопределения»
(с) Егор Бочаров | Ученик СУНЦ МГУ с 2018.

Рома работает тренером в Центре профессионального самоопределения. У него достаточно профессионализма для того, чтобы вести занятия одновременно в n комнатах! В комнате номер i занимаются a_i человек. Занятия идут полным ходом, но вот беда: Роме сообщили, что в ЦПС направляется комиссия СУНЦ МГУ, которая приехала лишь с одной целью — прорекламировать свою «чудесную» школу в «чудесном» городе Москва. В этой комиссии отлично знают, что здесь учатся такие ученики, как Егор, Наташа, Илья, а также Давид, в которых они очень заинтересованы.

Для Романа это совершенно типичная ситуация, из которой он с легкостью выходил уже не раз. Решать её в очередной раз ему совершенно не интересно, поэтому он решил поручить эту задачу вам. Вы должны как можно быстрее эвакуировать всех учеников ЦПС. Вы можете делать следующие операции сколь угодно раз:

- Встать перед какой-то комнатой j . На это уходит 0 секунд.
- Объявить эвакуацию, при этом, вас будет слышно в комнате, перед которой вы встали, а также в a комнатах слева от неё и в b комнатах справа. На это уходит 1 секунда.

К сожалению, из одной комнаты не может выйти больше чем x человек за одну операцию, так как администрация может заподозрить что-то неладное и остановить вашу тайную эвакуацию.

У вас очень мало времени, вы уже видите сотрудников СУНЦ в окнах. Стоит поторопиться!

Формат входных данных

В первой строке входных данных находятся четыре целых числа n, a, b, x ($1 \leq n \leq 10^5$; $0 \leq a, b \leq 10^9$; $1 \leq x \leq 10^9$).

Вторая строка выходных данных содержит n натуральных чисел a_i ($1 \leq a_i \leq 10^9$) — количество учеников, которые сидят в каждой из комнат.

Формат выходных данных

Выходные данные должны содержать одно число — минимальное время в секундах, за которое возможно эвакуировать всех учеников ЦПС.

Пример

ВВОД	ВЫВОД
7 1 1 1 2 3 3 2 3 3 2	6

Пояснение к примеру

В тестовом примере выше ответа 6 можно достичь следующим образом:

$[2, 3, 3, 2, 3, 3, 2] \Rightarrow [2, 3, 3, 1, 2, 2, 2] \Rightarrow [2, 2, 2, 0, 2, 2, 2] \Rightarrow [1, 1, 1, 0, 2, 2, 2]$

$\Rightarrow [0, 0, 0, 0, 2, 2, 2] \Rightarrow [0, 0, 0, 0, 1, 1, 1] \Rightarrow [0, 0, 0, 0, 0, 0, 0]$, где числа в квадратных скобках характеризуют текущее состояние в ЦПС, i -е число равно количеству людей в i -е комнате. Для вашего удобства, затронутые комнаты при каждой операции помечены жирным шрифтом.

Задача G. Крестики-нолики

Имя входного файла: *стандартный ввод* или `input.txt`
Имя выходного файла: *стандартный вывод* или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Играем на поле размером $W \times H$ клеток. Вначале в верхнем левом углу стоит X, в правом нижнем – O. Игроки ходят по очереди, первым ходит X.

Ставить X можно только рядом с другим X (сверху, снизу, слева, справа). Аналогично O можно ставить только рядом с другим O.

Проигрывает тот, кто поставит три X или O подряд (вертикально или горизонтально) или не сможет сделать ход.

Определите, кто выиграет, если оба игрока играют оптимально.

Формат входных данных

В единственной строке содержатся два целых числа W и H ($1 \leq W, H \leq 100$; $2 \leq W \times H$).

Формат выходных данных

Выведите фразу «X WINS», если выиграют крестики, или «O WINS», если выиграют нолики.

Пример

ВВОД	ВЫВОД
3 3	X WINS

Задача Н. Врёт ли Петя?

Имя входного файла: *стандартный ввод* или `input.txt`
Имя выходного файла: *стандартный вывод* или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Петя загадал последовательность $a_1, a_2, \dots, a_n$, состоящую из n целых чисел, и утверждает, что она является перестановкой последовательных натуральных чисел от 1 до n . Своему другу Васе он передаёт только последовательность $b_1, b_2, \dots, b_{n-1}$, для которой выполняется $b_i = a_i + a_{i+1}$.

Вася не доверяет своему другу и думает, что тот хочет его надурить, поэтому Вася хочет проверить, существует ли такая перестановка $a_1, a_2, \dots, a_n$, для которой последовательность b совпадает с последовательностью, переданной Петей.

Так как для него эта задача показалась слишком сложной, он поручил её вам.

Формат входных данных

В первой строке содержится единственное целое число n ($2 \leq n \leq 10^5$) — размер загаданной последовательности a .

Вторая строка содержит $n - 1$ целых чисел $b_1, b_2, \dots, b_{n-1}$ ($1 \leq b_i \leq 10^{18}$).

Формат выходных данных

В единственной строке выведите n целых чисел $a_1, a_2, \dots, a_n$, где a_i — значение i -го элемента найденной последовательности a .

Если никакая последовательность, удовлетворяющей условиям, не существует, выведите единственное число -1 .

Примеры

ВВОД	ВЫВОД
4 4 5 6	1 3 2 4
3 1 3	-1

Задача I. Музыкальный редактор

Имя входного файла:	<i>стандартный ввод</i> или <code>input.txt</code>
Имя выходного файла:	<i>стандартный вывод</i> или <code>output.txt</code>
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Илья — начинающий музыкант, недавно он начал изучать нотную грамоту.

Так как играть на музыкальных инструментах мальчик ещё не умеет, он установил программу, которая по заданной нотной записи может воспроизвести желаемую мелодию. В контексте этой задачи будем считать, что нотную запись можно представить как упорядоченное множество нот, где каждой ноте соответствует определённая частота.

Перед воспроизведением программа генерирует аудиофайл по следующему алгоритму:

1. Перебираются все ноты с частотами от 1 до h , где h — максимальное значение возможной частоты, это неизвестное значение, которое всегда больше частоты любой ноты в произведении.
2. При совпадении очередной ноты в записи с нотой в цикле эта нота считается сгенерированной, и генерация продолжается со следующей за ней ноты, если таковая имеется. Изначально генерируется самая первая нота.
3. Если после окончания цикла остались не сгенерированные ноты, то запускается новый цикл.

Например, если на генерацию попала запись $[2, 3, 3, 1]$, то произойдёт три запуска цикла: на первой итерации сгенерируется последовательность $[2, 3]$, на второй — $[3]$, на третьей — $[1]$.

Таким образом, на каждой итерации может быть сгенерировано более одной ноты, при этом одна итерация всегда занимает одну секунду, то есть в указанном выше примере генерация займёт три секунды.

Программа также может выполнять операции двух типов:

1. Изменить частоты всех нот на некотором отрезке, прибавив к ним число x .
2. Воспроизвести отрывок мелодии из нот, лежащих на некотором отрезке, порядок нот при этом сохраняется.

Экспериментируя над своей новой мелодией, Илья заметил, что для запросов второго типа программа не умеет определять, через какое время воспроизведётся нужный отрывок произведения. Это ему не понравилось, и, так как он не силен в программировании, он решил попросить вас написать ему новую программу, которая по заданным ей данным об изначальной мелодии и запросах смогла бы для каждого запроса второго типа определить время, требуемое для генерации очередного отрывка мелодии. Сможете ли вы помочь ему?

Формат входных данных

Первая строка входных данных содержит одно число n ($1 \leq n \leq 10^6$) — количество нот в изначальной версии произведения Ильи.

Вторая строка содержит n чисел a_i , означающих частоты нот ($1 \leq a_i \leq 10^9$).

Третья строка содержит одно число q — количество запросов Ильи ($1 \leq q \leq 2 \times 10^5$).

Каждая из следующих q строк содержит информацию об очередном запросе:

- запрос первого типа задаётся в формате «1 l_i r_i x_i », где входные параметры — позиции левой и правой нот изменяемого отрывка произведения, а также число, на которое меняются частоты изменяемых нот, соответственно ($1 \leq l_i \leq r_i \leq n$; $1 \leq |x_i| \leq 10^9$).
- запрос второго типа задаётся в формате «2 l_i r_i », где входные параметры — позиции левой и правой нот отрывка произведения, который нужно воспроизвести ($1 \leq l_i \leq r_i \leq n$).

Все числа во входных данных являются целыми. Гарантируется, что существует хотя бы один запрос второго типа и частоты всех нот в любой момент времени положительны.

Формат выходных данных

На каждый запрос второго типа нужно вывести единственное число — время, которое займёт генерация требуемого отрывка мелодии.

Отвечать на запросы следует в том же порядке, в котором они идут во входных данных.

Пример

ВВОД	ВЫВОД
5	1
1 3 2 3 5	2
6	2
2 1 2	2
1 3 3 2	
2 3 5	
1 2 3 -1	
2 3 4	
2 1 5	

Пояснение к примеру

На первом запросе второго типа генерируется последовательность [1, 3]. На втором запросе второго типа генерируются последовательности [4], [3, 5]. На третьем запросе второго типа генерируются последовательности [3], [3]. На четвертом запросе второго типа генерируются последовательности [1, 2, 3], [3, 5].

Задача J. Микроскоп

Имя входного файла: *стандартный ввод* или `input.txt`
Имя выходного файла: *стандартный вывод* или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Министерству образования и науки досталась от военных структур партия списанных, но ещё работающих атомно-силовых микроскопов. Найти им новое применение поручили подчинённой госструктуре под названием Ассоциация Содружеств Муниципалитетов (АСМ). Там приняли решение передать приборы в образовательные учреждения, где о такой технике могли только мечтать и где – как задумывалось – она будет использована для научных изысканий. Примерно так и случилось.

Первый экземпляр на один семестр достался Союзу Фиктивной Учёбы (СФУ), студенты которого моментально нашли применение бесценной технике: разобрали её с целью изучения внутреннего устройства. Разбирали аккуратно, поэтому в процессе разбили только одну линзу. Но прибор-то в конце семестра предстояло вернуть.

Тут надо сказать, что студентам СФУ уже доводилось использовать в учебном процессе разную оптическую технику, поэтому они умели изготавливать в учебных лабораториях плоские куски стекла круглой формы любого диаметра с точностью до нанометра. Проблема была только с определением этого диаметра: оригинальная линза разлетелась на мелкие осколки и определить её размер было невозможно. Тогда студенты решили сделать и засунуть в прибор самый большой стеклянный блин, какой туда поместится.

Корпус атомно-силового микроскопа имеет в основании форму правильного креста с размерами сторон l и w (см. иллюстрацию), и линза должна располагаться параллельно основанию. Размеры корпуса уже тщательно измерены. Вы – тот самый студент, которому дали задание рассчитать диаметр будущей «линзы». Результат нужен с точностью до нанометра, ведь наши герои не знают слова «халтура»!

Формат входных данных

Ввод содержит два целых числа l и w — размеры корпуса в нанометрах ($10^6 \leq l, w \leq 10^9$).

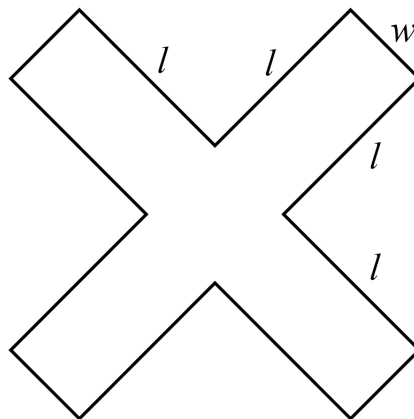
Формат выходных данных

Вывести диаметр куса стекла, который должны изготовить студенты. Ответ должен быть верным хотя бы с точностью до нанометра.

Пример

ВВОД	ВЫВОД
4000000 2000000	2828427.125

Иллюстрация



Задача К. Статья

Имя входного файла: *стандартный ввод* или `input.txt`
Имя выходного файла: *стандартный вывод* или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Студент Поликарп недавно закончил писать статью по своей научной работе. Он показал её своему научному руководителю, Роману Петровичу, который остался доволен статьёй, но дал указание вставить в текст ссылки на используемую литературу. Ссылка является списком целых чисел, разделённых запятыми, и помещённым в квадратные скобки.

Поликарп в спешке расставил ссылки в тексте, но уже перед тем, как ещё раз показывать статью Роману Петровичу, обнаружил, что некоторые списки с номерами не упорядочены, а в некоторых проблемы с пробелами. По ГОСТу, пробелы в списках должны быть только после запятой, разделяющих числа, а сами числа должны идти в порядке возрастания.

Поликарп знает, что если в этот раз статья не понравится Роману Петровичу, то за такую неверную попытку ему придётся стоять планку в течение времени, пропорциональному размеру статьи!

Так как статья очень большая, он решил автоматизировать процесс... с вашей помощью.

Формат входных данных

Статья состоит из не более чем 30 000 символов. В тексте встречаются только символы латинского алфавита, цифры, пробелы, а так же знаки «.,!?». Перенос строки считается за один символ.

Гарантируется, что каждый список ссылок не разрывается переносом, последовательность чисел в списке непустая и содержит различные целые числа из диапазона от 1 до 1 000.

Между любыми двумя соседними числами в списке присутствует ровно одна запятая. Запятые в списке присутствуют только между числами. По бокам от числа в списке может быть любое количество пробелов. Никакой пробел не будет находиться посреди числа.

Формат выходных данных

Выведите статью с отредактированными ссылками.

Весь текст, кроме ссылок, должен остаться неизменным.

Пример

ВВОД	ВЫВОД
Hello there! [1,2]	Hello there! [1, 2]
Some meaningful text... [6, 5, 4]	Some meaningful text... [4, 5, 6]
OK [9], good bye then.	OK [9], good bye then.