# Задача A. Автоспорт

| | |
|---|---|
| Имя входного файла: | `cars.in` |
| Имя выходного файла: | `cars.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

В кружке автомоделирования проводятся соревнования. Они проходят на прямом участке трассы длиной $L$ и шириной $W$. $N$ участников выставляют свои модели на стартовые позиции. По свистку все модели начинают движение.

Цель состязания — быстрее всех доехать до линии финиша. Однако в пути каждую модель подстерегают опасности. А именно, модель может врезаться в борта трассы и тем самым прекратить движение, или две и более модели могут столкнуться и также остановиться.

Введем систему координат, в которой ось $x$ будет направлена вдоль протяжения трассы, а ось $y$ — поперек ему. Тогда трасса является прямоугольником, ограниченным прямыми $y = 0$, $y = W$, $x = 0$, $x = L$.

Модель считается врезавшейся в борт трассы, если в некоторый момент времени её $y$-координата будет равняться либо 0, либо $W$. Две модели считаются столкнувшимися, если в некоторый момент времени их координаты совпадают. Модели можно считать материальными точками.

Модель побеждает в соревновании, если она успешно преодолела линию финиша (прямую $x = L$) и сделала это не позже любой другой модели. Возможно, что побеждает сразу несколько моделей, в этом случае, как говорится, «побеждает дружба». Возможно также, что ни одна модель по тем или иным причинам не сумеет преодолеть линию финиша.

Заметим отдельно, что если модель проходит через какую-либо из точек $(L, 0)$ или $(L, W)$, то считается, что она врезается в борт.

Поведение большинства моделей довольно предсказуемо, поэтому часто можно предугадать ход состязания. В данной задаче мы будем приближённо считать, что все модели начинают двигаться из своих начальных точек с заданными, постоянными на протяжении соревнования, векторами скорости.

Вам дана информация, известная на начало соревнования. Предскажите исход соревнования.

## Формат входного файла

В первой строке входного файла находятся три числа $N$, $L$, $W$. ($1 \le N \le 1000$, $1 \le L \le 10^4$, $2 \le W \le 10^4$).

В последующих $N$ строках находятся описания соревнующихся моделей — по 4 целых числа $x_i$, $y_i$, $vx_i$, $vy_i$, $1 \le i \le N$. $(x_i, y_i)$ — это координаты стартовой точки модели с номером $i$, $(vx_i, vy_i)$ — вектор скорости этой модели.

Гарантируется, что стартовые точки всех моделей различны, находятся на трассе, и не располагаются ни на каком-либо борту трассы, ни на линии финиша. Координаты векторов скорости не превышают $10^4$ по абсолютному значению.

## Формат выходного файла

В первой строке выходного файла выведите количество победителей. Во второй строке выведите их номера в порядке возрастания.

## Примеры

| cars.in | cars.out |
|---|---|
| 1 1 2<br>0 1 1 0 | 1<br>1 |
| 2 10 3<br>0 1 2 0<br>5 2 1 0 | 2<br>1 2 |

# Задача B. Игра с шоколадкой

| | |
|---|---|
| Имя входного файла: | `choco.in` |
| Имя выходного файла: | `choco.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

Петя с Андреем играют в интересную игру. Для этой игры им необходимо $k$ шоколадок, каждая из которых имеет форму прямоугольника — $i$-ая из них имеет размер $w_i$ на $h_i$ долек.

Петя с Андреем ходят по очереди, за один ход игрок должен взять со стола одну шоколадку, разломать ее на две неравных непустых части, и их обе положить обратно на стол. При этом обе получающиеся части должны иметь форму прямоугольника, длины сторон которого выражаются целым числом долек.

Первый ход в этой игре делает Петя. Проигрывает игрок, который не может сделать ход. Победитель после этого получает весь шоколад, лежащий на столе.

Ваша задача — определить, кто из игроков выиграет при оптимальной игре обоих.

## Формат входного файла

Входной файл содержит несколько наборов входных данных. Первая строка входного файла содержит количество $T$ наборов входных данных ($1 \le T \le 1000$).

Каждый набор входных данных описывается следующим образом. Первая строка описания содержит число $k$ ($1 \le k \le 100$) — количество шоколадок, лежащих на столе в начале игры. Последующие $k$ строк описывают шоколадки: $i$-ая из этих строк содержит два числа: $w_i$ и $h_i$ — размеры соответствующей шоколадки ($1 \le w_i, h_i \le 100$).

## Формат выходного файла

Для каждого набора входных данных выведите ответ. Следуйте формату, приведенному в примерах.

## Примеры

| choco.in | choco.out |
|---|---|
| 3<br>1<br>1 1<br>1<br>3 1<br>2<br>1 1<br>3 1 | Game 1: Andrew wins.<br>Game 2: Petr wins.<br>Game 3: Petr wins. |

# Задача C. Изысканные числа

| | |
|---|---|
| Имя входного файла: | `delicate.in` |
| Имя выходного файла: | `delicate.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

Шах одной азиатской страны роскошно завтракает каждое утро. Так как он считает себя математиком, как и некоторые из вас, он приказывает украсить свой стол... числами. Каждое число подается аккуратно написанным на прямоугольнике из глянцевой бумаги на маленькой серебряной тарелочке.

Однажды шах соизволил изучить двоичную систему счисления. С этого дня он приказал подавать ему *изысканные* числа. У таких чисел есть одна особенность: их запись в десятичной системе счисления является подстрокой их записи в двоичной системе счисления.

Каждый вечер он заказывает все изысканные числа из отрезка $[K; L]$ себе на завтрак. Конечно, у него много слуг, и они могут найти все такие числа за ночь, но вот тарелочки... Изысканные серебряные тарелочки должны также быть выкованы за ночь придворным кузнецом, так как шах не любит однообразия и хочет каждый день видеть разные тарелочки.

Помогите слугам и кузнецу — посчитайте необходимое количество тарелочек.

## Формат входного файла

Во входном файле в первой строке два числа — $K$ и $L$. ($0 \le K \le L \le 2^{63} - 1$).

## Формат выходного файла

Выведите количество изысканных чисел из отрезка $[K; L]$.

## Пример

| delicate.in | delicate.out |
|---|---|
| 0 10 | 3 |

# Задача D. Скидка

| | |
|---|---|
| Имя входного файла: | discount.in |
| Имя выходного файла: | discount.out |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

Межгалактическая сеть гипермаркетов $OJ$ устраивает сезонную распродажу. В связи с этим на каждую покупку предоставляется скидка. Однако правила предоставления скидки весьма необычны.

Во-первых, для предоставления скидки количество товаров в покупке должны быть не меньше $m$. Во-вторых, скидка предоставляется в размере 50% на выбранные покупателем три товара из покупки, но при этом суммарная стоимость этих трех товаров не должна превышать $c$ межгалактических денежных единиц.

Задан список товаров, входящих в покупку. Необходимо найти, за какую минимальную сумму можно их приобрести, воспользовавшись скидкой (если ей, конечно, можно воспользоваться) или не воспользовавшись. При этом все товары должны остаться в одной покупке.

## Формат входного файла

Первая строка входного файла содержит три целых числа: $n$, $m$, $c$ ($1 \le n, m \le 100$, $1 \le c \le 10^8$).

Вторая строка входного файла содержит $n$ целых чисел: стоимости товаров, входящих в покупку. Стоимость товара — неотрицательное число, не превосходящее $10^8$.

## Формат выходного файла

В выходной файл выведите ответ на задачу.

## Примеры

| discount.in | discount.out |
|---|---|
| 6 6 6<br>2 2 2 2 2 2 | 9 |
| 6 7 8<br>2 2 2 2 2 2 | 12 |

# Задача E. Сумма делителей

| | |
|---|---|
| Имя входного файла: | divsum.in |
| Имя выходного файла: | divsum.out |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

Пусть $x$ — натуральное число. Назовем $y$ его *делителем*, если $1 \le y \le x$ и остаток от деления $x$ на $y$ равен нулю.

Задано число $x$. Найдите сумму его делителей.

## Формат входного файла

Первая строка входного файла содержит целое число $x$ ($1 \le x \le 10^{18}$). Все простые делители числа $x$ не превосходят тысячу.

## Формат выходного файла

В выходной файл выведите ответ на задачу.

## Примеры

| divsum.in | divsum.out |
|---|---|
| 12 | 28 |
| 239 | 240 |

# Задача F. Три поросенка

| | |
|---|---|
| Имя входного файла: | `game.in` |
| Имя выходного файла: | `game.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

Три поросенка играют в следующую игру.

Первый поросенок придумывает натуральное число $N_1$ и сообщает его второму поросенку.

После этого второй поросенок придумывает такое натуральное число $N_2$, что существуют такие $A_1 \geq 2$, $B_1 \geq 2$, что $N_2 = A_1 + B_1$ и $A_1 \times B_1 = N_1$, и сообщает его третьему поросенку. Тот аналогично придумывает число $N_3$ и сообщает его первому поросенку, и т.д.

Если какой-нибудь поросенок называет число $N_i$, для которого следующий поросенок не может придумать числа $A_i$ и $B_i$, то он выигрывает. Если же игра никогда не заканчивается, то считается, что наступила ничья.

Дано число $N$. Определите, кто выигрывает, если первый поросенок начинает и придумывает число $N$.

Поросята играют оптимально. То есть, если для поросенка существует ход, приводящий к его выигрышу, то он его делает; если такого хода не существует, но существует ход, ведущий к ничье, то поросенок делает этот ход; иначе, он делает ход, максимизирующий его число $N_i$.

## Формат входного файла

В первой строке дано натуральное число $N$. $2 \leq N \leq 10^6$.

## Формат выходного файла

В выходной файл выведите порядковый номер поросенка, выигрывающего игру, или 0, если наступает ничья.

## Примеры

| game.in | game.out |
|---|---|
| 5 | 1 |
| 16 | 3 |
| 4 | 0 |

# Задача G. NEERC

| | |
|---|---|
| Имя входного файла: | `neerc.in` |
| Имя выходного файла: | `neerc.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

В полуфинале студенческого чемпионата мира по программированию NEERC (http://neerc.ifmo.ru) участвуют команды из $n$ институтов. Участники для проведения соревнований распределяются по $k$ залам, каждый из которых может вместить ограниченное число команд. При этом по правилам соревнований в одном зале может находиться не более одной команды от института.

Многие институты уже подали заявки на участие в полуфинале. Оргкомитет полуфинала хочет допустить до участия максимально возможное количество команд. При этом, разумеется, должна существовать возможность рассадить их по залам без нарушения правил.

Напишите программу, определяющую максимальное количество команд, которые можно допустить до участия в полуфинале.

## Формат входного файла

Первая строка входного файла содержит число $n$ — число институтов, подавших заявки ($1 \le n \le 100$). Вторая строка входного файла содержит $n$ чисел $a_1, \ldots, a_n$ ($a_i$ — это количество команд, заявленных от института номер $i$, $1 \le a_i$). Сумма всех $a_i$ не превосходит 2000.

Третья строка входного файла содержит число $k$ — количество залов, в которых проходят соревнования ($1 \le k \le 100$). Последняя строка входного файла содержит $k$ целых чисел $c_1, \ldots, c_k$ — вместимости залов ($\forall j\ 1 \le c_j$, сумма всех $c_j$ не превосходит 2000).

## Формат выходного файла

Выведите в выходной файл ответ на задачу.

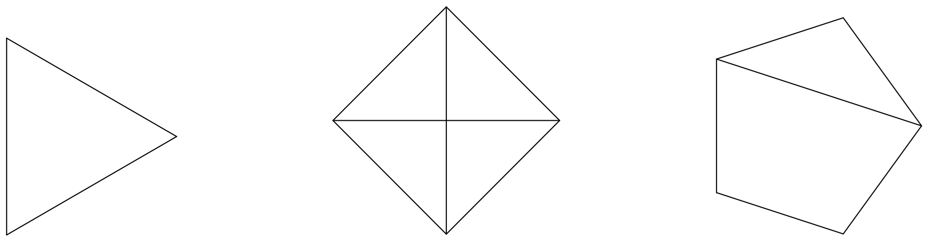
## Примеры

| neerc.in | neerc.out |
|---|---|
| 3<br>1 2 4<br>3<br>3 3 3 | 6 |
| 3<br>1 2 4<br>4<br>3 3 3 1 | 7 |

# Задача H. Части многоугольника

| | |
|---|---|
| Имя входного файла: | `parts.in` |
| Имя выходного файла: | `parts.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

В правильном $n$-угольнике провели $m$ диагоналей. Необходимо найти количество связных областей, которые при этом образовались.



## Формат входного файла

Первая строка входного файла содержит два целых числа $n$ ($3 \le n \le 40$) и $m$ ($0 \le m \le \frac{n \cdot (n-3)}{2}$).

Каждая из последующих $m$ строк содержит по два числа $u$ и $v$ ($1 \le u, v \le n$, $\min(|u - v|, n - |u - v|) > 1$) — номера вершин, соединенных соответствующей диагональю. Вершины нумеруются натуральными числами от 1 до $n$ против часовой стрелки.

Каждая диагональ упомянута во входном файле не более одного раза.

## Формат выходного файла

В выходной файл выведите ответ на задачу.

## Примеры

| parts.in | parts.out |
|---|---|
| 3 0 | 1 |
| 4 2<br>1 3<br>2 4 | 4 |
| 5 1<br>1 3 | 2 |

# Задача I. Ладья в лабиринте

| | |
|---|---|
| Имя входного файла: | `rook.in` |
| Имя выходного файла: | `rook.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

*Ладья* — это шахматная фигура, которая за один ход может переместиться на любое количество клеток по горизонтали или вертикали. При этом она не может «перепрыгивать» через стоящие на ее пути фигуры.

Вася недавно соорудил на шахматной доске своеобразный лабиринт — в некоторые клетки доски он поставил пешки (самые «слабые» шахматные фигуры). Теперь он хочет знать, за какое минимальное количество ходов ладья может добраться из одной клетки в другую.

Он размышляет над этим вопросом уже несколько дней, однако найти ответ не может. Поэтому он решил обратиться за помощью к Вам. Напишите программу, находящую ответ на Васину задачу.

## Формат входного файла

Первая строка входного файла содержит два натуральных числа: $n$ и $m$ ($1 \le n, m \le 500$) — размеры лабиринта.

Каждая из последующих $n$ строк содержит $m$ символов. $j$-ый символ $i$-ой из этих строк соответствует клетке с координатами $(i, j)$. Он равен «`.`» (точка), если клетка пуста, `P`, если занята пешкой, `S`, если это начальная клетка для ладьи, `F`, если это конечная клетка.

## Формат выходного файла

В выходной файл выведите минимальное количество ходов, требуемое ладье для того, чтобы из начальной клетки попасть в конечную. Если конечная клетка недостижима из начальной — выведите -1.

## Примеры

<table>
<tr><th>rook.in</th><th>rook.out</th></tr>
<tr><td><pre>4 4
F.PS
.PP.
.PP.
....</pre></td><td><pre>3</pre></td></tr>
<tr><td><pre>4 4
F.PS
.PP.
.PP.
.P..</pre></td><td><pre>-1</pre></td></tr>
</table>

# Задача J. Коробка с сахаром

| | |
|---|---|
| Имя входного файла: | `sugar.in` |
| Имя выходного файла: | `sugar.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

Маша — хозяйственная девочка и часто помогает маме по дому.

Сегодня мама купила новую коробку для сахара. Она имеет форму выпуклой прямоугольной призмы. То есть, в основании коробки лежит выпуклый многоугольник, а стенки коробки и ее ребра перпендикулярны дну.

Несложно посчитать, сколько сахара может поместиться в коробку, если рассчитывается, что ее можно закрыть крышкой. Объем сахара в этом случае равен площади многоугольника, помноженной на высоту коробки.

Но коробку необязательно закрывать, и в этом случае туда можно насыпать больше сахара.

Маше известно, что горка сахара не будет осыпаться, если наклон ее стенок относительно горизонтали не превышает $K$ градусов.

Помогите Маше узнать, насколько больше сахара можно насыпать в коробку, если ее не закрывать.

## Формат входного файла

В первой строке входного файла два целых числа $N$ и $K$, $3 \le N \le 100, 0 \le K \le 89$, где $N$ — число вершин в многоугольнике, лежащем в основании коробки.

Далее следует $N$ строк, в каждой из которых находятся координаты вершины многоугольника $X_i, Y_i$. Координаты вершин не превосходят по модулю $10^4$.

Гарантируется, что многоугольник выпуклый, его вершины заданы в порядке обхода против часовой стрелки, площадь его ненулевая и длина каждого ребра не меньше единицы.

## Формат выходного файла

Выведите ответ на задачу с абсолютной или относительной точностью $10^{-6}$. Это значит, что если правильный ответ равен $T$, ваш ответ равен $U$, и верно хотя бы одно из этих неравенств:

- $T - 10^{-6} \le U \le T + 10^{-6}$
- $T \cdot (1 - 10^{-6}) \le U \le T \cdot (1 + 10^{-6})$

то ваш ответ будет засчитан как правильный.

## Примеры

| sugar.in | sugar.out |
|---|---|
| 4 45<br>0 0<br>1 0<br>1 1<br>0 1 | 0.1666666666667 |
| 3 0<br>1 1<br>-1 -1<br>1 -1 | 0.0 |