

## Задача А. Нападение

Имя входного файла:	<code>war.in</code>
Имя выходного файла:	<code>war.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Всем известно о существовании вампиров. Однако, никто не задумывается о непосредственной близости этих существ. В сумеречном королевстве имеется  $n$  городов. Между некоторыми городами есть двусторонние дороги. В каждом городе есть свой вампирский клан, в который входит  $k_i$  вампиров.

В свое время вампиры изгнали из королевства всех оборотней, сделав их тем самым своими лютыми врагами. Оборотням это, естественно, не понравилось. Они решили объединиться в один отряд и напасть на какой-нибудь город. Отряд, нападающий на город, содержит  $w$  оборотней.

Однако, приняв в расчет то, что вампиры всегда действуют сообща, оборотни начали сомневаться в успешности их нападения. Теперь они пришли к вам за помощью. Известно, что вампиры могут перемещаться между двумя городами, соединенными дорогой, за один день. Также известно, что за один день сражения погибает  $\min(t, a, b)$  вампиров, обороняющих осажденный город, и столько же оборотней, где  $a$  — текущая численность отряда вампиров,  $b$  — текущая численность отряда оборотней,  $t$  — константа. По данной вам информации, требуется узнать, смогут ли вампиры защитить город. Город считается осажденным, если существует момент времени, когда число оборотней, напавших на город, больше нуля, а число вампиров, обороняющих город, равно нулю.

### Формат входного файла

В самой первой строке написано три числа:  $n$  ( $1 \leq n \leq 10^5$ ) — количество городов,  $m$  ( $1 \leq m \leq 10^5$ ) — количество дорог,  $t$  ( $1 \leq t \leq 10^3$ ) — количество вампиров и оборотней, погибающих за один день сражения. Во второй строке написано ровно  $n$  чисел:  $k_i$  ( $1 \leq k_i \leq 10^4$ ) — количество вампиров в  $i$ -м городе. Следующие  $m$  строк описывают дороги между городами: в каждой строке написано два числа — номера городов, соединенных дорогой. В последней строке написан номер города, на который было произведено нападение, и  $w$  ( $1 \leq w \leq 10^4$ ) — размер армии оборотней.

### Формат выходного файла

Выведите «Vampires win», если вампиры смогут отстоять свой город, и «Werewolves win» иначе.

## Пример

war.in	war.out
5 6 5 4 5 4 4 1 1 5 1 3 2 1 3 5 2 5 4 3 1 8	Werewolves win
5 5 1 3 1 5 1 1 1 2 1 5 1 4 4 5 2 3 1 11	Vampires win

## Комментарий

Обратите внимание, что в случае, если в городе погибают все вампиры, а новый отряд из соседнего города приходит на следующий день, вампиры все равно проигрывают.

## Задача В. Свидание

Имя входного файла: `dinner.in`  
Имя выходного файла: `dinner.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Эдвард и Белла пришли на свидание в ресторан «У Стефани». Этот ресторан подает всегда один и тот же набор блюд за фиксированную стоимость.

Когда Эдвард выложил деньги и ушел поудрить нос, Белла задумалась, хотел ли он заплатить за нее. Когда Эдвард хочет заплатить только за себя, он выкладывает на стол такой набор купюр, что при удалении из него любой купюры сумма денег будет меньше стоимости обеда. Обратная ситуация означает, что он хотел заплатить также и за Беллу.

К сожалению, Белле очень трудно дается устный счет, поэтому она просит вас помочь.

### Формат входного файла

В первой строке входного файла дано число  $k$  ( $1 \leq k \leq 10^9$ ) — стоимость обеда в этом ресторане. Во второй строке дано число  $n$  ( $1 \leq n \leq 1000$ ) — количество купюр, которые выложил Эдвард. В следующей строке даны  $n$  чисел  $a_i$  ( $1 \leq a_i \leq 1000$ ) — стоимости купюр.

### Формат выходного файла

В единственной строке выходного файла выведите «YES», если Эдвард точно собирается платить за Беллу, и «NO» в противном случае.

### Пример

<code>dinner.in</code>	<code>dinner.out</code>
169 2 200 100	YES
169 1 300	NO

### Комментарий

В первом тесте Эдвард не стал бы выкладывать вторую купюру, если бы собирался платить только за себя.

## Задача С. Чтение

Имя входного файла: `reading.in`  
Имя выходного файла: `reading.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В необычной и насыщенной событиями жизни Беллы иногда находится место самым обыденным занятиям. Например, иногда она читает абсолютно обыкновенные книги. Однако, в последнее время Белле начали сниться странные сны, связанные с этим занятием.

В своих снах Белла также читала книгу, но текст в этой книге был не обычным художественным текстом, а случайным бредом, не несущим никакой смысловой нагрузки. При этом, сны были настолько реалистичными, что Белла стала периодически терять ощущение того, где она находится — во сне или наяву.

Помогите Белле по тексту, который напечатан в книге, которую она читает в данный момент, определить, находится ли она во сне или наяву.

### Формат входного файла

Входной файл содержит некоторое количество символов, ASCII-код которых не превышает 127 — текст, напечатанный в книге, которую в данный момент читает Белла. Гарантируется, что количество символов в каждом тесте не превышает 50 000. Кроме этого, гарантируется, что нет ни одного теста, количество символов в котором меньше 10 000.

Известно, что этот текст является или отрывком некоторого художественного произведения на английском языке (возможно, переведенным с русского языка), или сгенерирован программой-генератором случайного текста. Код программы-генератора, написанный на языке Java, вы можете найти, перейдя по ссылке <http://neerc.ifmo.ru/school/io/reading/>

### Формат выходного файла

Выведите в выходной файл фразу «`Real life`», если данный вам текст является художественным текстом, или же «`Dreaming`», если данный текст сгенерирован программой-генератором.

### Пример

<code>reading.in</code>	<code>reading.out</code>
<code>Having thanked Anna Pavlovna for her charming...</code>	<code>Real life</code>
<code>Lnх. Wlqytv gmudkiup fhtem, wyhprlrd...</code>	<code>Dreaming</code>

### Комментарий

По понятной причине, привести тест полностью в рамках условия не представляется возможным. Первые два теста доступны по ссылкам <http://neerc.ifmo.ru/school/io/reading/01.in> и <http://neerc.ifmo.ru/school/io/reading/02.in>.

В тестах, являющихся настоящими английскими текстами, могут встречаться опечатки и нарушения прафил орфографии и пунктуации, принятых в английском языке. Однако, жюри гарантирует, что они встречаются в количестве, не оказывающем существенного влияния на возможные решения этой задачи.

## Задача D. Вампирский баскетбол

Имя входного файла:	basket.in
Имя выходного файла:	basket.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Как известно, вампиры любят и хорошо умеют играть в разные спортивные игры. Но так как сейчас осень, и играть на улице в бейсбол уже прохладно, вампиры решили собраться в спортивном зале и поиграть в баскетбол. Как только они попали в зал, то сразу поделились на две команды. Эдвард всегда играет на позиции разыгрывающего защитника, потому что любит давать пасы. Для определенной расстановки игроков на поле он хочет знать, кто из его команды находится в наиболее выгодной позиции, а кто нет.

Поле представляет из себя полуплоскость, каждая точка которой задается целыми координатами  $x$  и  $y$  такими, что координата  $y$  неотрицательна. Кольцо находится в точке  $(0, 0)$ . Также есть трехочковая дуга, представляющая полуокружность с радиусом  $r$  и центром в точке, в которой находится кольцо. Если расстояние игрока до кольца строго больше, чем  $r$ , то его забитый бросок считается за три очка, иначе за два. Эдвард знает, что все игроки его команды бросают одинаково. Каждый игрок команды противника обладает защитными навыками, измеряемыми

Каждый вампир имеет майку, на которой написан его номер, причем все номера для игроков одной команды различны и идут подряд. Таким образом, если в команде Эдварда три вампира, а в команде противников два, то они будут иметь номера 1, 2, 3 и 1, 2. Сам Эдвард носит майку с номером 0, и собственная позиция ему безразлична. Эдвард считает выгодность позиции вампира из его команды под номером  $i$  по формуле

$$b_i = \frac{p_i}{d_i^2 \cdot (1 + \sum z_j)}$$

, где  $p_i$  — количество очков, которые можно получить, забив с этой точки,  $d_i$  — расстояние от вампира до кольца, а  $\sum z_j$  — сумма мер защитных навыков противников, расстояние от которых до нападающего не превышает  $l$ .

Чтобы помочь Эдварду, нужно найти последовательность номеров вампиров из его команды в порядке убывания выгодности их позиции. При равенстве выгодностей позиций вампиры идут по возрастанию номера. Тогда Эдвард наконец-то сможет решить, кому же он хочет отдать пас.

### Формат входного файла

В самой первой строке заданы числа  $n, m, r, l$  ( $1 \leq n, m \leq 10^3, 1 \leq r, l \leq 10^5$ ) — количество вампиров в команде Эдварда, количество противников, радиус трехочковой дуги, расстояние, определяющее влияние защитников на игроков.

В следующих  $n$  строках перечислены перечислены координаты сокомандников Эдварда  $x_i$  и  $y_i$  ( $1 \leq y_i \leq 10^5, -10^5 \leq x_i \leq 10^5$ ) в порядке возрастания номеров на майках. В следующих  $m$  строках перечислены координаты противников  $x_j, y_j$  ( $1 \leq y_j \leq 10^5, -10^5 \leq x_j \leq 10^5$ ) также в порядке возрастания номеров и  $z_j$  ( $1 \leq z_j \leq 100$ ) — мера защитного умения противника под номером  $j$ .

Все позиции всех игроков различны.

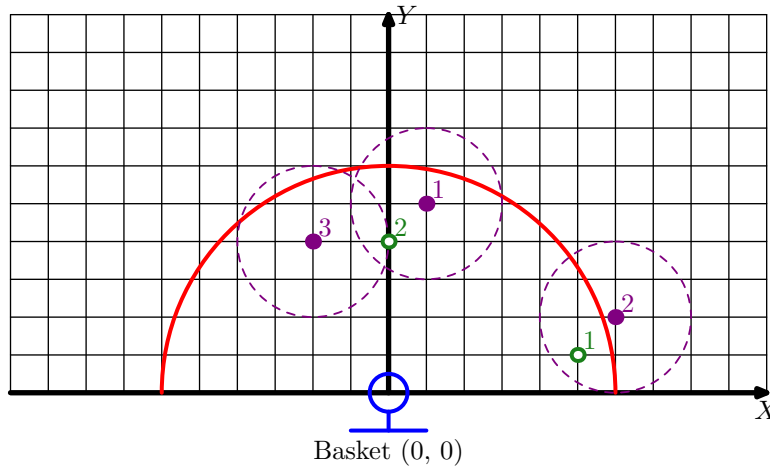
### Формат выходного файла

Выведите в выходной файл  $n$  чисел — последовательность номеров вампиров в нужном порядке.

## Пример

basket.in	basket.out
3 2 6 2	2 3 1
1 5	
6 2	
-2 4	
5 1 2	
0 4 3	

## Комментарий



На картинке изображен пример.  $b_1 = \frac{2}{26 \cdot 4} = \frac{1}{52}$ ,  $b_2 = \frac{3}{40 \cdot 3} = \frac{1}{40}$ ,  $b_3 = \frac{2}{20 \cdot 4} = \frac{1}{40}$ .  
Обратите внимание, что кольцо является точкой.

## Задача Е. Ликантропия

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Это интерактивная задача.

Джейкоб — оборотень, умеющий превращаться в волка. Сейчас он в теле человека и хочет превратиться в волка. Но его волчьей сущности надоели частые превращения туда-обратно и она пытается сопротивляться превращению.

Для превращения в волка, мозг Джейкоба посылает сигнал в нервную систему. Сигнал распространяется по нервной системе вниз, от мозга к сердцу. Нервная система Джейкоба состоит из  $n$  нейронов и  $m$  синапсов. Мозг генерирует сигнал в первом нейроне, после чего начинается его распространение до сердца (нейрона с номером  $n$ ). Каждый синапс может передавать сигнал из  $a_i$ -го нейрона в  $b_i$ -й ( $a_i < b_i$ ). После того, как сигнал приходит в нейрон, он должен отправиться дальше по какому-нибудь синапсу. Если синапса, по которому сигналу можно отправиться, нет, сигнал затухает и превращения в волка не происходит. Если сигнал доходит до сердца, начинается превращение в волка.

Волчья сущность хочет помешать Джейкобу начать превращение в волка. Для этого она, каждый раз, сразу после того, как сигнал приходит в нейрон, может временно дестабилизировать работу не более, чем  $k$  синапсов, и они не смогут переносить сигнал, пока он не уйдёт из того нейрона, в котором он находится в данный момент. После ухода сигнала, волчья сущность снова сможет дестабилизировать (возможно, другие) синапсы, и так далее, до затухания сигнала, либо до начала превращения в волка. Работоспособность ранее дестабилизированных синапсов при этом восстанавливается.

Помогите волчьей сущности провалить попытку превращения.

### Протокол взаимодействия с программой жюри:

В первой строке заданы числа  $n$ ,  $m$  и  $k$  ( $2 \leq n \leq 50$ ,  $0 \leq m \leq 10\,000$ ,  $0 \leq k \leq m$ ) — число нейронов, число синапсов и максимальное число дестабилизируемых волчьей сущностью синапсов.

Далее, в  $m$  строках заданы описывающие синапсы числа  $a_i$  и  $b_i$  ( $1 \leq a_i < b_i \leq n$ ) — откуда и куда может транспортироваться  $i$ -м синапсом сигнал.

Если не существует способа помешать превращению в волка, дестабилизируя при каждом перемещении сигнала не более  $k$  синапсов, чтобы не дать сигналу дойти до сердца, выведите «NO» и завершите программу.

Иначе, много раз повторяется следующее:

- Выведите число  $l$  ( $0 \leq l \leq k$ ) — количество синапсов, работу которых необходимо дестабилизировать на этом этапе, и затем  $l$  различных чисел  $s_i$  ( $1 \leq s_i \leq m$ ) — номера дестабилизируемых синапсов.
- Считайте очередное  $v$  — новое местоположение сигнала. Если  $v = -1$ , то завершите работу программы. Если сигнал дошёл до нейрона  $n$ , то волчья сущность проиграла.

## Пример

stdin	stdout
4 4 2 1 2 1 3 2 4 3 4 -1	2 1 2
4 4 2 1 2 1 3 2 4 3 4 2 -1	0 1 3
2 3 2 1 2 1 2 1 2	NO

## Комментарий

Для корректной работы программы после каждой операции вывода данных вам необходимо делать следующие операции:

- В языке Pascal: `flush(output);`
- В C/C++: `fflush(stdout);`
- В Java: `System.out.flush();`
- В Python: `sys.stdout.flush();`

Кроме этого, не забывайте после каждой выведенной строки ставить перевод строки.



## Задача F. Мощность вампира

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайта

Джейкоб думает очень громко.

---

Эдвард

Это интерактивная задача.

Все вампиры обладают уникальной способностью читать мысли, и Эдвард очень часто пользовался ей для защиты своей жены Беллы Свон. Поэтому, когда на нее напали несколько вампиров клана Вольтури, он тут же решил прочитать их мысли, дабы узнать мощь каждого напавшего вампира.

Но каждый вампир этого клана обладает ментальным щитом, поэтому узнать мощь отдельного вампира невозможно. Эдвард может лишь узнать сумму всех чисел, выражающих их мощь, но при этом он может брать каждое число с нужным ему знаком (плюс или минус).

Эдварду срочно необходима ваша помощь, иначе Беллу съедят вампиры из клана Вольтури.

### Протокол взаимодействия с программой жюри:

Программа жюри выводит число  $n$  в отдельной строке ( $1 \leq n \leq 100$ ). После этого не более чем  $n$  раз повторяются следующие действия.

Ваша программа выводит в отдельной строке строку, состоящую из символов  $a_i$  ( $a_i = +$  или  $a_i = -$ ), разделенных пробелами, где  $i$ -й символ означает, с каким знаком мы просуммируем мощь  $i$ -го вампира  $p_i$  ( $-100 \leq p_i \leq 100$ ).

Когда вы найдете мощь каждого вампира, вашей программе необходимо вывести в строчку их все по порядку:

`answer: a1 a2 ... an`

где  $a_i$  — мощь  $i$ -го вампира.

### Примеры

stdin	stdout
2	+ +
3	+ -
-1	answer: 1 2

### Комментарий

Для корректной работы программы после каждой операции вывода данных вам необходимо делать следующие операции:

- В языке Pascal: `flush(output);`
- В C/C++: `fflush(stdout);`
- В Java: `System.out.flush();`
- В Python: `sys.stdout.flush();`

Кроме этого, не забывайте после каждой выведенной строки ставить перевод строки.

## Задача G. Пещеры

Имя входного файла: `cave.in`  
Имя выходного файла: `cave.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Как известно, много лет назад оборотни, предки Джейкоба и его семьи, жили в пещерах неподалеку от залива. Причем дабы избежать всевозможных проблем все они жили по одиночке.

Однако, это еще не все странности оборотней того времени. Каждый раз, на протяжении суток после очередного солнечного затмения, оборотни, возвращаясь с охоты, могли заходить в свои пещеры только по очень странному правилу, описанному далее.

Издравле сложилось, что оборотни всегда охотятся стаями, поэтому на момент их возвращения с охоты все пещеры пусты. Для удобства понимания правила, по которому оборотни заходят в пещеры, пронумеруем пещеры натуральными числами от 1 до  $n$ , начиная с самой дальней от залива пещеры.

Процесс заселения пещер состоит в следующем: сначала выбирается пещера, которую оборотни хотят заселить, или наоборот, из которой они хотят выселить оборотня. После этого, если там есть оборотень, то он выходит, а если нет, то оборотень туда заселяется.

Однако, вся сложность в том, что оборотни могут выбрать не любую пещеру, а либо пещеру с номером один, либо пещеру с номером  $x + 1$ , где  $x$  — первая занятая пещера.

Определите, в каком порядке оборотни могли заселять пещеры.

### Формат входного файла

В единственной строке дано одно натуральное число  $n$  — количество пещер и оборотней ( $1 \leq n \leq 20$ ).

### Формат выходного файла

В первой строке выведите число  $k$  — количество совершенных действий. Во второй строке выведите последовательность действий оборотней. Если пещера освобождается, то номер пещеры должен быть выведен со знаком минус. Иначе выводите просто номер заселяемой пещеры. Количество действий не должно превышать  $10^6$ . Если существует несколько возможных решений задачи, то разрешается вывести любое.

### Пример

<code>cave.in</code>	<code>cave.out</code>
3	5 1 2 -1 3 1

## Задача Н. Магические стержни

Имя входного файла:            `rod.in`  
Имя выходного файла:        `rod.out`  
Ограничение по времени:    2 секунды  
Ограничение по памяти:      256 мегабайта

У Беллы есть  $n$  пронумерованных магических стержней. Для получения магической энергии Белла поджигает каждый стержень с двух концов (возможно, в различные моменты времени). Огонь распространяется со скоростью один сантиметр в секунду. В момент времени, когда огонь охватывает стержень целиком, его магическая энергия освобождается.

Вам известно что длина  $i$ -го стержня равна  $l_i$  сантиметрам. Помогите Белле выбрать моменты времени поджога концов каждого стержня так, чтобы магическая энергия со всех стержней освободилась в один и тот же момент времени. При этом Белла хочет, чтобы в  $i$ -м стержне последняя точка, которая будет охвачена огнем, находилась на расстоянии  $d_i$  от первого конца стержня.

Времена поджогов всех концов всех стержней должны быть целыми положительными числами. Среди всех возможных вариантов ответа выберете тот, который минимизирует момент времени, когда вся магическая энергия освободится.

### Формат входного файла

В первой строке задано одно целое число  $n$  ( $1 \leq n \leq 10\,000$ ) — количество стержней у Беллы. В следующих  $n$  строках записано по два числа  $l_i$  и  $d_i$  ( $2 \leq l_i \leq 1\,000\,000$ ,  $0 < d_i < l_i$ ) — длина  $i$ -го стержня, а также расстояние от первого конца  $i$ -го стержня до точки на этом стержне, которая должна загореться последней.

### Формат выходного файла

В  $i$ -й строчке выходного файла выведите два положительных целых числа  $t_{i1}$ ,  $t_{i2}$  — моменты времени, в которые надо поджечь первый и второй концы  $i$ -го стержня соответственно.

### Примеры

<code>rod.in</code>	<code>rod.out</code>
2	2 2
2 1	2 1
3 1	