

## Задача А. Рассадка

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

Олимпиада всегда приносит много хлопот для жюри и оргкомитета, и очный тур Открытой олимпиады не является исключением из этого правила. Пока пишутся условия, готовятся тесты, создаются разборы и настраиваются компьютеры, очень легко забыть правильно рассадить участников олимпиады по аудиториям.

В этот раз, в целях рационального использования рабочих мест, оргкомитет олимпиады поставил перед собой задачу — создать *компактную* рассадку школьников по  $M$  имеющимся аудиториям. Рассадка называется *компактной*, если для любых двух различных аудиторий количества детей в них отличаются не более чем на один.

В жюри олимпиады имеется человек с экстрасенсорными способностями, который может очень точно предсказывать результаты будущих соревнований по программированию. Хотя результаты заочного тура ещё не известны, он без труда определил, что на очный тур пройдут ученики из  $K$  школ. Кроме того, было предсказано, что из  $i$ -й школы ( $1 \leq i \leq K$ ) пройдёт в точности  $n_i$  учеников. Хотя отношение большей части оргкомитета к подобным прогнозам зачастую скептическое, они имеют странное свойство сбываться, поэтому вам предлагается на всякий случай вычислить максимально возможное количество участников, которые попадут в одну аудиторию при компактной рассадке, если предсказание окажется верным.

### Формат входных данных

В первой строке ввода находятся два целых числа  $M$  и  $K$  — количество аудиторий и количество школ, участвующих в очном туре олимпиады согласно предсказанию ( $1 \leq M \leq 10^9$ ,  $1 \leq K \leq 10^5$ ).

В следующей строке находятся  $K$  чисел  $n_i$ , разделённых пробелами,  $i$ -е из которых обозначает, сколько участников предположительно приедет из  $i$ -й школы ( $1 \leq n_i \leq 10^9$ ).

### Формат выходных данных

Выведите одно число — максимальное количество людей в аудитории при компактной рассадке.

### Примеры

stdin	stdout
2 2 5 9	7
1 2 20 10	30

### Замечание

В первом тесте существует только одна компактная рассадка — в каждую из аудиторий рассадить по 7 человек.

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и **всех тестов предыдущих групп**.

Группа	Тесты	Баллы	Ограничения			Комментарий
			$K$	$M$	$n_i$	
0	1–2	0	–	–	–	Тесты из условия.
1	3–13	25	$K \leq 2$	$M \leq 2$	$n_i \leq 100$	
2	14–20	25	$K \leq 2$	$M \leq 3$	$n_i \leq 10^9$	
3	21–27	25	$K \leq 4$	$M \leq 10^9$	$n_i \leq 10^9$	
4	28–34	25	$K \leq 10^5$	$M \leq 10^9$	$n_i \leq 10^9$	Дополнительных ограничений нет.

## Задача В. Расписание

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

Как же Илье надоело учиться! Сначала школа, потом университет... Вот, наконец, наступил тот долгожданный день, когда Илье утром не надо ехать на учебу. Но, к несчастью для Ильи, оказалось, что после окончания университета начинается самое трудное — надо устраиваться на работу.

Во всемирно известной фирме «Goondex», в которую устроился Илья, принято очень много работать, в частности, для сотрудников установлена шестидневная рабочая неделя. Но, в качестве бонуса, «Goondex» каждый год предлагает своим сотрудникам выбрать любой день недели в качестве выходного. В свою очередь, оставшиеся шесть дней недели будут рабочими.

Илья сообразил, что с учётом государственных праздников (которые всегда являются выходными) с помощью правильного выбора выходного дня недели можно варьировать количество рабочих дней в году. Теперь он хочет знать, какой день недели ему следует выбрать в качестве выходного, чтобы отдыхать как можно больше дней в году, или, наоборот, демонстрировать чудеса трудолюбия, работая по максимуму.

### Формат входных данных

В первой строке входных данных находится одно целое число  $N$  ( $0 \leq N \leq 366$ ) — количество государственных праздников.

Во второй строке содержится одно целое число  $year$  ( $1800 \leq year \leq 2100$ ) — год, в который необходимо помочь Илье.

В каждой из последующих  $N$  строк расположено по паре чисел  $day\ month$  ( $day$  — целое число,  $month$  — слово, между  $day$  и  $month$  ровно один пробел), обозначающих, что день  $day$  месяца  $month$  является государственным праздником.

В последней строке расположено слово  $dayOfWeek$  — день недели первого января в год  $year$ .

Гарантируется, что все даты указаны корректно (в том числе указанный день недели первого января действительно является днём недели первого января соответствующего года  $year$ ) и все дни государственных праздников различны.

### Формат выходных данных

Выведите через пробел два дня недели — лучший и худший варианты дней недели для выходного (то есть дни недели, для которых достигается соответственно максимальное и минимальное количество выходных дней в году). Если возможных вариантов ответа несколько, выведите любой из них.

### Примеры

stdin	stdout
2 2015 1 January 8 January Thursday	Monday Thursday

### Замечание

Соответствие названий месяцев и дней недели в английском и русском языках:

Месяцы			
January	Январь	July	Июль
February	Февраль	August	Август
March	Март	September	Сентябрь
April	Апрель	October	Октябрь
May	Май	November	Ноябрь
June	Июнь	December	Декабрь

Дни недели	
Monday	Понедельник
Tuesday	Вторник
Wednesday	Среда
Thursday	Четверг
Friday	Пятница
Saturday	Суббота
Sunday	Воскресенье

Соответствие названий месяцев и количества дней в них:

Январь	31	Июль	31
Февраль	28 (29)	Август	31
Март	31	Сентябрь	30
Апрель	30	Октябрь	31
Май	31	Ноябрь	30
Июнь	30	Декабрь	31

В феврале 29 дней только в високосные года. Год является високосным, если он кратен 400, либо кратен 4 и не кратен 100. Например, 1996 и 2000 являются високосными, а 1999 и 1900 — нет.

Тесты к этой задаче состоят из трёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и **всех тестов предыдущих** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Тесты	Баллы	Дополнительные ограничения		Комментарий
			$N$	$year$	
0	1	0	—	—	Тесты из условия.
1	2–15	20	$N \leq 3$	$2000 \leq year \leq 2015$	
2	16–54	80	—	—	<b>Offline-проверка.</b>

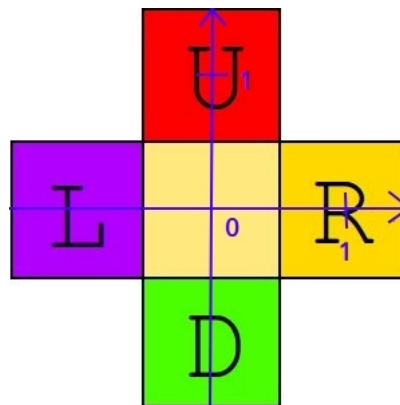
## Задача С. DDR

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

Вася играет в «Dance Dance Revolution». Задача игрока состоит в том, чтобы нажимать ногами расположенные на игровом полу стрелочки в порядке их появления на экране. Игровой процесс сопровождается энергичной музыкой, и невнимательному зрителю может показаться, что Вася просто танцует. На самом же деле он очень сосредоточен на том, чтобы минимизировать суммарное перемещение своих ног, не допустив при этом ни одной ошибки.

Уточним описание игры:

1. Игровое поле состоит из пяти одинаковых по размеру квадратных платформ: центральная без стрелки, а также четыре платформы с различными стрелками, находящиеся по разные стороны от центральной. Будем называть их левой, правой, верхней и нижней стрелками, и обозначим заглавными буквами латинского алфавита 'L', 'R', 'U' и 'D' соответственно. Для лучшего понимания рекомендуется посмотреть на картинку.



2. Песня состоит из  $N$  позиций. Для каждой позиции известно, какие стрелки должны быть нажаты. Поскольку Вася всего лишь человек, то это либо одна, либо две одновременно нажатых стрелки.
3. Вася уже достаточно опытный игрок, чтобы стоять на игровом поле как угодно: любая нога Васи может находиться на любом из пяти элементов поля вне зависимости от положения другой ноги (в том числе он может поставить обе ноги на одну и ту же клетку).
4. Вася любит наблюдать за осьминогами, поэтому для него не проблема переместить ноги из любого исходного положения в любое другое.

Цель Васи — нажать на все стрелки в требуемой последовательности. При этом он имеет право стоять на стрелке даже тогда, когда она не должна быть нажата: например, если в данный момент требуется нажать только на стрелку 'L', то он может поставить одну ногу на 'L', а другую — куда угодно. Также Вася в любой момент может поставить одну или обе ноги на центральный элемент без стрелки. Изначально Вася стоит обеими ногами на центральном элементе, а в конце песни его ноги могут находиться где угодно.

В рамках данной задачи будем считать, что ноги Васи являются точками. Для нажатия на платформу Васе нужно поставить ногу в центр соответствующей платформы. Обратите внимание, что это относится и к случаю, когда Вася пользуется своим правом поставить какую-либо ногу на центральную платформу. Возможные координаты ног Васи:  $(0, 0)$ ,  $(1, 0)$ ,  $(0, 1)$ ,  $(-1, 0)$  и  $(0, -1)$ . Координаты перечислены в порядке: центральная, правая, верхняя, левая, нижняя платформы.

Помогите Васе найти минимальное расстояние, которое в сумме придётся пройти его ногам во время песни. Имеется в виду евклидово расстояние для точек на плоскости. Например, расстояние

от центральной платформы до любой другой равно 1, но расстояние между верхней и правой равно  $\sqrt{2}$ .

## Формат входных данных

В первой строке содержится одно целое число  $N$  ( $1 \leq N \leq 100\,000$ ) — количество позиций в песне.

Следующие  $N$  строк содержат описания позиций в следующем формате: сначала следует число  $k_i$  — количество стрелок, которые должны быть нажаты в данный момент времени ( $k_i \in \{1, 2\}$ ). Затем следуют  $k_i$  различных символов (разделённых пробелом, если  $k_i = 2$ ) из множества  $\{'L', 'R', 'U', 'D'\}$ , обозначающих стрелки, участвующие в  $i$ -й позиции.

## Формат выходных данных

Выведите одно вещественное число — минимальное суммарное перемещение ног, которого может достичь Вася, правильно нажав на все стрелки в требуемой последовательности. Ваш ответ будет считаться правильным, если его абсолютная или относительная ошибка не будет превосходить  $10^{-9}$ .

Более формально, пусть ваш ответ равен  $A$ , а ответ жюри —  $B$ . Проверяющая программа сочтёт ваш ответ правильным, если  $\frac{|A-B|}{\max(1, B)} \leq 10^{-9}$ .

## Примеры

stdin	stdout
3 1 U 1 D 2 L R	4.8284271247
2 1 U 1 U	1.0000000000

## Замечание

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы.

Группа	Тесты	Баллы	Дополнительные ограничения		Комментарий
			$N$	$k_i$	
0	1–2	0	—	—	Тесты из условия.
1	3–10	30	$N \leq 20$	$k_i = 2$	Стрелка вниз не встречается.
2	11–19	30	$N \leq 20$	—	
3	20–29	40	—	—	Дополнительных ограничений нет.

## Задача D. Мультиплеер в змейке

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

Компания Konia, специализирующаяся на выпуске бюджетных мобильных устройств, в последнее время теряет свои позиции на рынке. Ради спасения компании совет директоров пошел на отчаянный шаг: было решено внести изменения в самую известную игру, которая доступна на всех устройствах от Konia — Змейку.

Правила этой игры очень просты. На клетчатом поле, состоящем из  $N$  строк и  $M$  столбцов, находится змейка, изначально имеющая длину 1 (т.е. представляющая собой квадратик  $1 \times 1$ ). На каждом ходу змейка перемещается на одну клетку в одном из четырёх направлений: вверх, вниз, влево или вправо. Если при этом змейка пересекает какую-либо границу экрана, то она появляется с противоположной стороны. Например, сделав шаг влево из первой клетки четвертой строки (будем говорить, что такая клетка имеет координаты  $(4, 1)$ ) на поле размером 10 строк на 8 столбцов, змейка окажется в клетке с координатами  $(4, 8)$ . Иногда на поле появляется еда, которая может увеличить длину змейки, но это не имеет отношения к данной задаче.

По предложению совета директоров в Змейку было решено добавить режим парной игры. Изначально змейки двух игроков появляются в разных клетках и имеют длину 1, то есть состоят только из клеток, в которых они появились. Каждый ход состоит из того, что сначала перемещается змейка первого игрока, а затем змейка второго игрока. Правила перемещения аналогичны описанным выше для одиночной игры, но, в отличие от одиночного режима, змейки имеют возможность столкнуться друг с другом, после чего игра заканчивается. Маркетинговые исследования показали, что пользователи предпочитают как можно более длинные игры, поэтому перед разработчиками стоит задача исследовать, как зависит длительность игры от начальных координат змеек  $(a_1, b_1)$  и  $(a_2, b_2)$ . Так как в режиме парной игры пока реализована только возможность передвигаться, еда на поле не появляется, и фактически игра сводится к управлению двумя одноклеточными змейками.

Вам, как ведущему инженеру компании, поручено контролировать деятельность коллег. Для этого необходимо написать программу, которая по заданным размерам поля  $N$  и  $M$  и координатам двух различных клеток  $(a_1, b_1)$  и  $(a_2, b_2)$  вычисляет, на каком по счёту ходу впервые *может возникнуть* ситуация, что змейки столкнутся, то есть одна змейка совершит ход в клетку, в которой находится другая.

### Формат входных данных

В первой строке ввода содержатся целые числа  $N$  и  $M$  ( $1 \leq N, M \leq 1\,000\,000$ ), задающие соответственно количество строк и количество столбцов на игровом поле. Обратите внимание, что строки нумеруются от 1 до  $N$  снизу вверх, а столбцы — от 1 до  $M$  слева направо.

Во второй строке записаны два целых числа  $a_1$  и  $b_1$  ( $1 \leq a_1 \leq N$ ,  $1 \leq b_1 \leq M$ ) — начальные координаты змейки первого игрока: сначала номер строки, затем номер столбца.

В третьей строке аналогично задаются  $a_2$  и  $b_2$  — начальные координаты змейки второго игрока.

Гарантируется, что в начале игры змейки игроков находятся в разных клетках.

### Формат выходных данных

Выведите номер первого хода, на котором змейки могут столкнуться.

## Примеры

stdin	stdout
5 8 5 5 5 7	1
10 10 1 1 10 10	1
10 4 3 1 10 3	3

## Замечание

В первом тесте существует единственная клетка в которой они могут встретиться через один ход — клетка с координатами 5, 6.

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп.

Группа	Тесты	Баллы	Дополнительные ограничения	Комментарий
			$N, M$	
0	1–3	0	–	Тесты из условия.
1	4–16	30	$N, M \leq 4$	
2	17–49	30	$N, M \leq 100$	
3	50–69	40	–	Дополнительных ограничений нет.





## Формат выходных данных

В первой строке выведите два числа: номер цвета, в который будет покрашено *наибольшее* число узлов на плаще, и количество узлов, покрашенных в этот цвет.

Во второй строке выведите два числа: номер цвета, в который будет покрашено *наименьшее* число узлов на плаще, и количество узлов, покрашенных в этот цвет.

Если правильных ответов несколько, выведите любой из них.

## Примеры

stdin	stdout
7 4 3	2 14 3 13
2 3 2	1 6 2 6

## Замечание

Плащ из первого примера изображён на рисунке из условия.

Во втором примере в каждый из двух цветов покрашено одинаковое число узлов плаща.

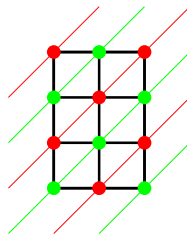


Рис. 3: Плащ размером  $2 \times 3$ , покрашенный в два цвета.

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу тестов ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп.

Группа	Тесты	Баллы	Дополнительные ограничения		Комментарий
			$N, M$	$K$	
0	1 – 2	0	–	–	Тесты из условия.
1	3 – 37	25	$N, M \leq 1000$	–	
2	38 – 55	25	$N, M \leq 1\,000\,000$	–	
3	56 – 68	25	–	$K \leq 1\,000\,000$	
4	69 – 85	25	–	–	Дополнительных ограничений нет.

## Задача F. Анархия в Берляндии

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

С незапамятных времен Берляндией правили добрые и справедливые цари, а власть передавалась по наследству. Но вот настали тяжёлые времена, по воле рока династия прервалась, и страна погрузилась в хаос. Некоторые города были захвачены враждующими друг с другом бандами головорезов, в других же городах какая-либо власть исчезла вовсе.

Всё началось с того, что в некоторых городах установилась власть преступных банд, причём под властью каждой из банд изначально оказалось ровно по одному городу. Каждая банда избрала в качестве столицы город своего первоначального зарождения. Начиная с того времени, каждый год *ровно одна* из банд решала расширить свои владения, захватывая некоторый город, до этого не бывший ни в чьём подчинении или находящийся под властью другой банды.

Для осуществления захвата вооружённый отряд данной банды выходил из своей столицы и двигался к выбранному городу, захватывая все города на своём пути. Также известно, что столицы всех банд были так серьёзно защищены, что через них невозможно было проехать (и уж тем более не могло идти речи об их захвате). Поэтому на пути от столицы банды-захватчика до выбранного города не могло быть столиц других банд.

Как уже неоднократно отмечалось различными историками, в описываемые времена в Берляндии было  $N$  городов, соединённых между собой  $N - 1$  дорогой с двусторонним движением, при этом от каждого города можно было доехать до любого другого. Подобное устройство транспортной системы естественным образом налагало ограничения на возможные маршруты путешествия.

События эти происходили уже очень давно, а хроники тех лет были утрачены в вихре смутного времени. Теперь же к вам в руки попала карта Берляндии тех времён. Вы обнаружили, что на карте отмечены столицы каждой из банд, а также для каждого города указано, какая банда им владела на определённый момент времени (свободных городов к тому году уже не осталось).

Вычислите, при какой возможной последовательности захватов могло возникнуть описанное этой картой распределение контроля банд над городами, или же определите, что карта содержит ошибку, так как указанное на карте распределение власти банд над городами не могло быть получено с помощью последовательности описанных выше захватов. При этом, так как Вам не хочется заниматься трудоёмкой проверкой своего ответа, Вам необходимо найти последовательность, состоящую из не более чем  $N$  захватов.

### Формат входных данных

В первой строке входных данных через пробел указаны два целых числа  $N$  и  $M$  — количество городов на карте Берляндии и количество преступных банд ( $1 \leq M \leq N \leq 300\,000$ ).

В следующих  $N - 1$  строках описываются дороги Берляндии. Каждая строка содержит два различных целых числа  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq N$ ), обозначающих, что между городами с номерами  $a_i$  и  $b_i$  существовала дорога с двусторонним движением.

В следующей строке входных данных через пробел указаны  $M$  различных целых чисел  $c_i$  ( $1 \leq c_i \leq N$ ): согласно карте, в городе с номером  $c_i$  находилась столица банды  $i$ .

В последней строке входных данных находятся  $N$  разделённых пробелами целых чисел  $d_i$ , обозначающих, что город под номером  $i$  ( $1 \leq i \leq N$ ) отмечен на карте как находящийся под властью банды с номером  $d_i$  ( $1 \leq d_i \leq M$ ). Гарантируется, что  $d_{c_i} = i$  для всех  $i$ .

### Формат выходных данных

Если существует такая последовательность захватов, которая приводит к указанному на карте распределению контроля банд над городами, то в первой строке выведите слово «YES» (без кавычек). Затем в следующей строке выведите единственное целое число  $S$  — длину найденной последовательности захватов (при этом должно выполняться неравенство  $0 \leq S \leq N$ ). В следующих  $S$  строках выведите по два различных числа  $x_i$  и  $y_i$  — номера городов, являющихся столицей банды, захватывающей города в  $i$ -м году, и выбранным городом-концом захвата в  $i$ -м году соответственно. Если

же карта содержит ошибку и получить распределение контроля банд над городами, указанное на карте, невозможно, то в единственной строке вывода выведите слово «NO» (без кавычек).

Гарантируется, что если существует последовательность захватов, приводящая к указанному на карте распределению контроля банд над городами, то такое распределение контроля банд над городами всегда могло возникнуть не более чем за  $N$  лет.

## Примеры

stdin	stdout
4 2 1 3 2 3 3 4 1 2 1 2 2 1	YES 2 1 4 2 3
5 3 1 3 2 3 3 4 3 5 1 2 3 1 2 3 1 3	NO

## Замечание

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и **всех тестов предыдущих** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Тесты	Баллы	Дополнительные ограничения		Комментарий
			$N$	$M$	
0	1-2	0	—	—	Тесты из условия.
1	3–30	30	$N \leq 10$	—	
2	31–60	30	$N \leq 1000$	—	
3	61–90	40	—	—	<b>Offline-проверка.</b>