

Задача А. Распродажа

Имя входного файла: `ain.txt`
 Имя выходного файла: `aout.txt`
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 64 мегабайта

Магазины в рекламных целях часто устраивают распродажи. Так, например, одна из крупных сетей магазинов канцелярских товаров объявила два рекламных предложения: «купи N одинаковых товаров и получи еще один товар бесплатно» и «купи K товаров по цене $K - 1$ товара».

Для проведения олимпиады организаторам требуется распечатать условия для участников, на что уходит очень много бумаги. Каждая пачка стоит B рублей. Какое максимальное количество пачек бумаги можно приобрести на A рублей, правильно используя рекламные предложения?

Формат входного файла

Во входном файле записаны целые числа N , K , A и B ($1 \leq N \leq 100$, $2 \leq K \leq 100$, $1 \leq A \leq 10000$, $1 \leq B \leq 10000$), разделенные пробелами.

Формат выходного файла

Выведите одно целое число — максимальное количество пачек бумаги, которое смогут купить организаторы олимпиады.

Примеры

<code>ain.txt</code>	<code>aout.txt</code>
4 4 13 2	8
3 4 8 3	2
3 4 7 1	9

В первом примере, дважды используя второе рекламное предложение, можно купить 8 пачек бумаги, заплатив за 6.

Во втором примере рекламными предложениями воспользоваться нельзя.

В третьем примере можно по одному разу воспользоваться каждым из двух рекламных предложений и на оставшийся рубль купить еще одну пачку бумаги.

Задача В. Дипломы в папках

Имя входного файла: `bin.txt`
 Имя выходного файла: `bout.txt`
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 64 мегабайта

В этом году Иван заканчивает школу и поступает в вуз. За время своей учебы он часто участвовал в олимпиадах по информатике и у него накопилось много дипломов. Иван раскладывал дипломы по папкам совершенно бессистемно, то есть любой диплом мог оказаться в любой из папок. К счастью, Иван помнит, сколько дипломов лежит в каждой из папок.

Иван хочет принести в приемную комиссию выбранного вуза папку, в которой находится диплом Московской олимпиады по программированию (такой диплом у Ивана ровно один). Для того чтобы понять, что в данной папке нужного диплома нет, Ивану нужно просмотреть все дипломы из этой папки. Просмотр одного диплома занимает у него ровно одну секунду и он может мгновенно переходить к просмотру следующей папки после окончания просмотра предыдущей. Порядок просмотра папок Иван может выбирать.

По заданному количеству дипломов в каждой из папок требуется определить, за какое наименьшее время в худшем случае Иван поймет, в какой папке содержится нужный ему диплом.

Формат входного файла

В первой строке входного файла записано целое число N ($1 \leq N \leq 100$) — количество папок. Во второй строке записаны N целых чисел a_1, a_2, \dots, a_N ($1 \leq a_i \leq 100$) — количество дипломов в каждой из папок.

Формат выходного файла

Выведите одно число — минимальное количество секунд, необходимое Ивану в худшем случае для определения того, в какой папке содержится диплом.

Пример

<code>bin.txt</code>	<code>bout.txt</code>
2	1
2 1	

В примере Иван может просмотреть папку 2 за 1 секунду и, не найдя там диплома, понять, что диплом находится в папке 1.

Задача С. Параллелограмм

Имя входного файла: cin.txt
 Имя выходного файла: cout.txt
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 64 мегабайта

На уроке геометрии семиклассники Вася и Петя узнали, что такое параллелограмм. На перемене после урока они стали играть в игру: Петя называл координаты четырех точек в произвольном порядке, а Вася должен был ответить, являются ли эти точки вершинами параллелограмма.

Вася, если честно, не очень понял тему про параллелограммы, и ему требуется программа, умеющая правильно отвечать на Петины вопросы.

Напомним, что параллелограммом называется четырехугольник, противоположные стороны которого равны и параллельны.

Формат входного файла

В первой строке входного файла записано целое число N ($1 \leq N \leq 10$) — количество заданных Петей вопросов. Каждая из N последующих строк содержит описание четырех точек — четыре пары целых чисел X и Y ($-100 \leq X \leq 100$, $-100 \leq Y \leq 100$), обозначающих координаты точки. Гарантируется, что четыре точки, о которых идет речь в одном вопросе, не лежат на одной прямой.

Формат выходного файла

Для каждого из вопросов выведите «YES», если четыре заданные точки могут образовать параллелограмм, и «NO» в противном случае. Ответ на каждый из запросов должен быть в отдельной строке без кавычек.

Пример

cin.txt	cout.txt
3	YES
1 1 4 2 3 0 2 3	NO
1 1 5 2 2 3 3 0	YES
0 0 5 1 6 3 1 2	

Задача D. Поход

Имя входного файла: din.txt
 Имя выходного файла: dout.txt
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 64 мегабайта

Группа школьников решила сходить в поход вдоль Москвы-реки. У Москвы-реки существует множество притоков, которые могут впадать в нее как с правого, так и с левого берега.

Школьники хотят начать поход в некоторой точке на левом берегу и закончить поход в некоторой точке на правом берегу, возможно, переправляясь через реки несколько раз. Как известно, переправа как через реку, так и через приток представляет собой определенную сложность, поэтому они хотят минимизировать число совершенных переправ.

Школьники заранее изучили карту и записали, в какой последовательности в Москву-реку впадают притоки на всем их маршруте.

Помогите школьникам по данному описанию притоков определить минимальное количество переправ, которое им придется совершить во время похода.

Формат входного файла

Единственная строка содержит описание Москвы-реки между начальной и конечной точкой похода. Длина строки не превосходит 200 символов.

Каждый символ строки может быть одной из трех латинских букв L, R или B. Буква L означает, что очередной приток впадает в реку с левого берега, R — приток впадает в реку с правого берега и B — притоки впадают с обоих берегов реки в одном месте. Поход начинается на левом берегу перед описанной частью реки и заканчивается на правом берегу после описанной части.

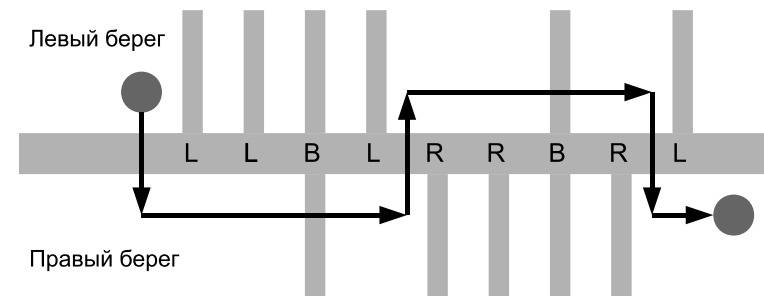
Формат выходного файла

Выведите одно целое число — минимальное количество переправ.

Пример

din.txt	dout.txt
LLBLRRBRL	5

Рисунок к приведенному выше примеру.



Задача Е. Футурама

Имя входного файла: ein.txt
 Имя выходного файла: eout.txt
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 64 мегабайта

С помощью изобретенной профессором машины Фарнсворт и Эми меняются телами с целью осуществить свои мечты: профессор жаждет острых ощущений, а Эми мечтает есть от пуза, не опасаясь за фигуру. Впоследствии выясняется, что обмен разумом между двумя телами возможен не более одного раза, и чтобы вернуться обратно в свои тела нужно произвести промежуточный обмен. Бендер предлагает свою помощь, однако, получив тело Эми, он тут же скрывается, чтобы под чужой личиной украсть корону императора Робо-Венгрии.

Эми, недовольная возможностями тела профессора в плане обжорства, уговаривает поменяться Лилу. Фрай приходит в ужас. Лила обижена и обвиняет Фрая в том, что его заботит только ее внешность. Фрай в отместку меняет телами с Зойдбергом.

Бендер оказывается пойман при попытке ограбления, однако освобождается, убедив императора в том, что он — робот в теле человека. Узнав, что император тайне мечтает пожить немного жизнью простых людей, Бендер предлагает тому на время поменяться телами. Но так как Профессор уехал рисковать жизнью в теле Бендера, пришлось подsunуть императору вместо своего корпуса автоматизированное помойное ведро.

Фрай в теле Зойдберга и Лила в теле Профессора встречаются в ресторане, чтобы выяснить отношения. В конце концов они понимают, что любят друг друга вовсе не за внешность. При виде сцены их бурного примирения Эми, на этот раз уже в теле Гермеса, надолго теряет аппетит.

Бендер, поменявшись телами с правителем Робо-Венгрии, наслаждается жизнью на его яхте. Однако именно в этот вечер заговорщики совершают покушение на императора. Жизнь Бендеру спасает появление профессора Фарнсворта.

После того, как все герои решают свои личные проблемы, профессору с помощью Бубльгума Тэйта и Сладкого Клайда из команды «Ударники» удается вернуть всех в свои тела.

“Футурама”. Десятый эпизод шестого сезона.

В очередной серии Футурамы было проведено несколько обменов разумами между телами героев, но, по крайней мере Бубльгум Тэйт и Сладкий Клайд в обменах не участвовали. Теперь необходимо вернуть разумы всех героев в свои тела. К сожалению, два тела могут участвовать только в одном обмене, поэтому обратные обмены для этого произвести невозможно. Например, если тело 1 поменялось разумом с телом 2, а потом тело 1 поменялось разумом с телом 3, то в теле 1 находится разум третьего героя, в теле 2 — разум первого героя, а в теле 3 — второго. Теперь можно произвести обмен разумами только между телами 2 и 3, тогда разум второго героя вернется в свое тело, а первому и третьему героям могут помочь только Тэйт с Клайдом.

Помогите героям Футурамы вернуться в свои тела.

Формат входного файла

Во входном файле записаны целые числа N ($4 \leq N \leq 20$) и M ($1 \leq M \leq 100$) — количество героев Футурамы и количество произведенных обменов разумами. Герои занумерованы числами от 1 до N , изначально разум каждого из героев находится в своем теле. В последующих M строчках записана последовательность совершенных обменов разумами. Каждый обмен описывается двумя различными числами — номерами тел, которые в этом обмене меняются разумами. Бубльгум Тэйт и Сладкий Клайд, как наиболее разумные герои, имеют номера $N - 1$ и N , и гарантируется, что в исходных обменах они не участвовали.

Решения, верно работающие в случаях, когда каждое тело участвовало в обмене не больше одного раза будут набирать не менее 40 баллов.

Формат выходного файла

Выведите план обменов для возвращения разумов героев в свои тела в виде пар различных чисел — номеров тел которые участвовали в соответствующем обмене. Причем никакие два тела не должны обмениваться между собой разумами более одного раза, включая исходные обмены. Если обменов не требуется, то можно ничего не выводить. Если планов обменов несколько, то выведите любой из них (не обязательно минимальный).

Вернуть разумы героев в свои тела всегда возможно.

Пример

ein.txt	eout.txt
4 1	1 3
1 2	2 4
	1 4
	2 3
	3 4

Приведем таблицу положения героев в телах после каждого из обменов:

Обмен	Тело 1	Тело 2	Тело 3	Тело 4
До обменов	1	2	3	4
1—2	2	1	3	4
1—3	3	1	2	4
2—4	3	4	2	1
1—4	1	4	2	3
2—3	1	2	4	3
3—4	1	2	3	4