

Информация о проведении второго отборочного тура Московской олимпиады 6-9 (квалификационного отбора)

Второй отборочный этап Московской олимпиады 6-9 классов пройдет в форме квалификационного отбора. Участие во втором отборочном этапе участников, прошедших на Московскую олимпиаду по результатам первого отборочного этапа не обязательно.

Квалификационный отбор заключается в том, что нужно решить и сдать в проверяющую систему три задачи из одной из Московских олимпиад по программированию для 6-9 классов 2008, 2009, 2010, 2011, 2012, 2013 или 2014 года. При этом эти задачи обязательно должны быть из одной олимпиады (то есть сдать по одной задаче из трех разных олимпиад - нельзя).

Задачи должны быть сданы в проверяющую систему на сайте informatics.mosolymp.ru (в разделе "Квалификационный отбор на Московскую олимпиаду по программированию 2015 года") под личным логином участника. В информации о пользователе должны быть корректно прописаны имя и фамилия участника, e-mail должен совпадать с адресом, указанным при регистрации на олимпиаду. Если у вас уже сданы эти задачи (и вы видите себя в соответствующей таблице результатов) - сдавать задачи еще раз не нужно.

Для участия во втором отборочном туре 6-9 классов (квалификационном отборе) необходимо:

1. Заполнить регистрационную форму на сайте ЕСР. Можно использовать свой логин для Московских олимпиад по другим предметам. Указывайте корректный e-mail!
2. Сдать три задачи в разделе квалификационного отбора на сайте informatics.mosolymp.ru. Проверьте, что e-mail вашего пользователя совпадает с адресом из предыдущего пункта!

Вопросы по отбору и олимпиаде 6-9 классов можно задать по адресу: inf-moscow-9@mosolymp.ru

Во всех задачах ввод данных происходит из стандартного потока ввода, вывод результата – в стандартный поток вывода (соответственно, ввод – с клавиатуры, вывод – на экран), либо ввод из файла `input.txt`, вывод – в файл `output.txt` (на ваше усмотрение). В решениях на `qbasic` возможен только ввод из файла и вывод в файл. В решениях на `turbo pascal` не используйте модуль `crt`.

Во всех задачах:

Максимальное время работы на одном тесте: 1 секунда

Максимальный объем используемой памяти: 64 мегабайта

Задача А. Детали

Имеется N кг металлического сплава. Из него изготавливают заготовки массой K кг каждая. После этого из каждой заготовки вытачиваются детали массой M кг каждая (из каждой заготовки вытачивают максимально возможное количество деталей). Если от заготовок после этого что-то остается, то этот материал возвращают к началу производственного цикла и сплавляют с тем, что осталось при изготовлении заготовок. Если того сплава, который получился, достаточно для изготовления хотя бы одной заготовки, то из него снова изготавливают заготовки, из них – детали и т.д.

Напишите программу, которая вычислит, какое количество деталей может быть получено по этой технологии из имеющихся исходно N кг сплава.

Формат входных данных

Вводятся N, K, M . Все числа натуральные и не превосходят 200.

Формат выходных данных

Выведите одно число — количество деталей, которое может получиться по такой технологии.

Примеры

Входные данные	Выходные данные
10 5 2	4
13 5 3	3
14 5 3	4
13 9 4	2

Задача В. Налог

«Курс валюты Зимбабве опустился накануне до рекордно низкого уровня - 1,2 млрд. зимбабвийских долларов за один доллар США»

(Новости от 7.06.2009)

В некоторой стране инфляция достигла таких размеров, что доходы граждан стали выражаться числами, количество знаков в десятичной записи которых доходит до 200. Это сильно усложнило задачу взимания налогов.

Один из налогов на доходы составляет 1%. Напишите программу, которая по введенному числу D (величине дохода гражданина) вычислит этот налог.

При этом применяются следующие правила округления:

1. Если налог выражается целым числом, то он не округляется.
2. Если налог выражается дробным числом, то он округляется в сторону большего целого (в пользу государства).

Формат входных данных

Вводится одно число D (натуральное, $10^5 \leq D < 10^{200}$) – величина дохода гражданина.

Формат выходных данных

Выведите одно натуральное число – величину налога.

Примеры

Входные данные	Выходные данные
1000001	10001
12345600	123456

Частичные ограничения

Решения, верно работающие при $10^5 \leq D < 10^9$, будут оцениваться из 40 баллов.

Решения, верно работающие при $10^5 \leq D < 10^{15}$, будут оцениваться из 60 баллов.

Задача С. Игра

Мальчик Вася играет в свою любимую RPG. Он нашел сундук с M ячейками, в каждой из которых лежит по одной бутылке с зельем лечения. У его героя на поясе есть N карманов, в каждом из которых также лежит по одной бутылке. Каждая бутылка восстанавливает фиксированное число очков здоровья.

Вася хочет заменить часть бутылок, находящихся в кармане на поясе, бутылками из сундука так, чтобы суммарное количество очков здоровья, восстанавливаемых бутылками, которые окажутся на поясе после этого, было максимальным. Ему доступна одна операция: поменять бутылку из указанного кармана пояса с бутылкой из указанной ячейки сундука.

Вам нужно указать последовательность операций, после которой суммарный запас очков здоровья у Васи на поясе будет максимальный.

Формат входных данных

Сначала вводятся N, M ($1 \leq N \leq 1000, 1 \leq M \leq 1000$). Далее идут N чисел, причём i -е равно количеству очков здоровья, восстанавливаемых бутылкой из i -го кармана пояса. Далее – M чисел, j -е из которых равно количеству очков здоровья, восстанавливаемых бутылкой из j -й ячейки сундука. Все очки – натуральные числа, не превосходящие 10000.

Формат выходных данных

Вначале выведите K – количество операций обмена. Оно не должно превышать 100000. Далее выведите K пар чисел, описывающих, какие бутылки нужно поменять: первое из чисел от 1 до N – задает номер кармана на поясе, второе – от 1 до M – номер ячейки в сундуке. Если существует более одного варианта, выведите любой.

Примеры

Входные данные	Выходные данные
1 2 1 2 3	1 1 2
2 2 3 1 4 5	2 1 1 2 2

Задача D. Строительство школы

В деревне Интернетовка все дома расположены вдоль одной улицы по одну сторону от нее. По другую сторону от этой улицы пока ничего нет, но скоро все будет – школы, магазины, кинотеатры и т.д.

Для начала в этой деревне решили построить школу. Место для строительства школы решили выбрать так, чтобы суммарное расстояние, которое проезжают ученики от своих домов до школы, было минимально.

План деревни можно представить в виде прямой, в некоторых целочисленных точках которой находятся дома учеников. Школу также разрешается строить только в целочисленной точке этой прямой (в том числе разрешается строить школу в точке, где расположен один из домов – ведь школа будет расположена с другой стороны улицы).

Напишите программу, которая по известным координатам домов учеников поможет определить координаты места строительства школы.

Формат входных данных

Сначала вводится число N — количество учеников ($1 \leq N \leq 100000$). Далее идут в строго возрастающем порядке координаты домов учеников — целые числа, не превосходящие $2 \cdot 10^9$ по модулю.

Формат выходных данных

Выведите одно целое число — координату точки, в которой лучше всего построить школу. Если ответов несколько, выведите любой из них.

Примеры

Входные данные	Выходные данные
4 1 2 3 4	2
3 -1 0 1	0

Частичные ограничения

Решения, верно работающие при $1 \leq N \leq 1000$ для координат, не превосходящих по модулю 1000, будут оцениваться из 30 баллов.

Решения, верно работающие при $1 \leq N \leq 100000$ для координат, не превосходящих по модулю 10^5 , будут оцениваться из 70 баллов.

Задача Е. Поклейка обоев

Однажды майор Пронин затеял в квартире ремонт. В одной из стен на кухне по плану потребовалось последовательно проделать $(N-1)$ прямоугольных вентиляционных отверстий с горизонтальными и вертикальными сторонами ($1 \leq N \leq 100$). Если оказывалось, что очередное отверстие пересекается с уже проделанными, то майор вырезал только нетронутую часть соответствующего прямоугольника.

Следующая стадия после ремонта – это поклейка обоев. В магазине напротив майор может заказать не более $(2N-1)^2$ прямоугольных кусков обоев любых размеров с ненулевой площадью. Он хочет обклеить стену кусками обоев так, чтобы:

1. Вентиляционные отверстия не были заклеены даже частично.
2. Никакие два куска не пересекались (касаться сторонами они при этом могут).
3. На стене не осталось бы непокрытой области.

Формат входных данных

Рассмотрим декартову систему координат, оси которой параллельны сторонам отверстий и стены.

Сначала вводится число N ($1 \leq N \leq 100$), далее – описание N прямоугольников. Первый прямоугольник описывает положение стены в нашей системе координат, остальные $(N-1)$ — положения отверстий в порядке их появления. Стороны всех прямоугольников параллельны осям координат. Каждый прямоугольник задаётся координатами своих левого нижнего и правого верхнего углов: x_1, y_1, x_2, y_2 . Координаты — целые числа, не превосходящие по модулю 31000, $x_1 < x_2, y_1 < y_2$.

Прямоугольники, обозначающие положение отверстий, **могут** пересекаться и касаться, поскольку это могло быть необходимо в ходе ремонта. Разумеется, все вентиляционные отверстия находятся в стене, то есть не выходят за границы первого прямоугольника.

Формат выходных данных

Вначале выведите количество кусков обоев K , которое нужно заказать в магазине (K должно быть не больше $(2N-1)^2$). Далее выведите схему поклейки: K прямоугольников, обозначающих места расположения заказанных кусков. Для каждого прямоугольника нужно вывести координаты его левого нижнего и правого верхнего углов. **Все координаты должны быть целыми числами.** Гарантируется, что решение существует.

Если возможных способов несколько, выведите любой.

Примеры

Входные данные	Выходные данные
2 -1 -1 2 2 0 0 1 1	5 -1 -1 2 0 -1 0 0 2 0 1 1 2 1 0 2 1 1 1 2 2

Частичные ограничения

Решение, верно работающее для координат, не превышающих по модулю 200, будет оцениваться из 60 баллов.

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.

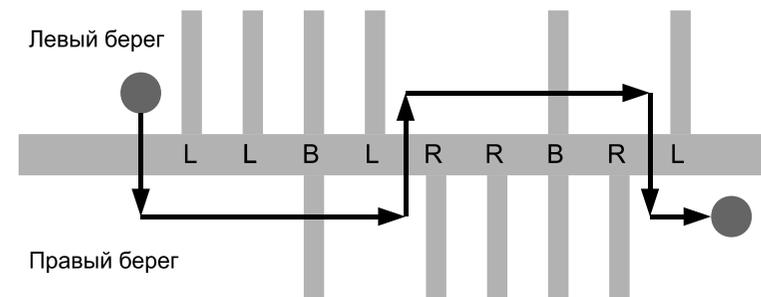
Формат выходного файла

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 вывести. 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

Программа может читать данные с клавиатуры (стандартного ввода) и выводить результат на экран (стандартный вывод), либо программа может использовать файловый ввод-вывод (имена файлов указаны в условии задачи).

Во время тура каждая задача тестируется на части тестов, так называемая online-группа тестов. На тестах online-группы программа может набрать от 40 до 60 баллов в зависимости от задачи (смотрите условия). Результат проверки решения на online-группе будет доступен во время тура.

После окончания тура последнее сданное решение по каждой задаче будет проверено на тестах offline-группы. Суммарное количество баллов за тесты online-группы и offline-группы равно 100 по каждой задаче.

Всего за тур вы можете отправлять решения на проверку не более 60 раз (суммарно по всем задачам).

Задача А. Кола

Имя входного файла: `ain.txt`
Имя выходного файла: `aout.txt`
Ограничение по времени: 0,2 секунды
Ограничение по памяти: 64 мегабайта

Завод по производству колы изготавливает ее не только для магазинов, но и для всемирно известной сети ресторанов быстрого питания.

Ежедневно завод отгружает один и тот же объем колы в литрах. Служба доставки сети ресторанов обычно использует для транспортировки колы емкости объемом или только 50 литров, или только 70 литров. Если доставка осуществляется с помощью емкостей в 50 литров, то для перевозки имеющегося объема колы необходимо A емкостей. А если с помощью емкостей в 70 литров, то необходимо B емкостей. При этом в каждом из случаев одна из емкостей может быть заполнена не полностью.

Недавно сеть ресторанов решила утвердить новый объем емкостей для доставки колы — 60 литров. Сколько емкостей теперь может понадобиться для доставки того же самого объема колы?

Формат входного файла

Входные данные содержат 2 числа A и B , расположенных каждое в отдельной строке ($1 \leq A, B \leq 10\,000\,000$).

Формат выходного файла

Выведите все возможные значения для количества емкостей по 60 литров, которые окажутся заполненными (в том числе одна возможно частично), в порядке возрастания или число -1 , если значения A и B противоречат друг другу, то есть они были записаны неверно.

Примеры

<code>ain.txt</code>	<code>aout.txt</code>
3 2	2 3
1 2	-1

В первом примере колы могло быть, например, 115 литров, в этом случае понадобится две емкости в 60 литров, а могло быть — 135 литров, в этом случае понадобятся уже три емкости по 60 литров. Четыре емкости не могут понадобиться никогда.

Система оценивания

Online-группа тестов оценивается в 60 баллов, в этой группе $1 \leq A, B \leq 1\,000$.

Offline-группа тестов оценивается в 40 баллов.

Задача В. Числа

Имя входного файла: `bin.txt`
Имя выходного файла: `bout.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Саша и Катя учатся в начальной школе. Для изучения арифметики при этом используются карточки, на которых написаны цифры (на каждой карточке написана ровно одна цифра). Однажды они пришли на урок математики, и Саша, используя все свои карточки, показал число A , а Катя показала число B . Учитель тогда захотел дать им такую задачу, чтобы ответ на нее смогли показать и Саша, и Катя, каждый используя только свои карточки. При этом учитель хочет, чтобы искомое число было максимально возможным.

Формат входного файла

Во входном файле записано два целых неотрицательных числа A и B (каждое число в одной строке). Длина каждого из чисел не превосходит 100 000 цифр.

Формат выходного файла

Выведите одно число — максимальное целое число, которое можно составить используя как цифры первого числа, так и цифры второго числа. Если же ни одного такого числа составить нельзя, выведите -1 .

Примеры

<code>bin.txt</code>	<code>bout.txt</code>
280138 798081	8810
123 456	-1

Система оценивания

Online-группа тестов оценивается в 60 баллов, в этой группе числа A и B содержат не более 1000 цифр каждое. При этом решения, правильно работающие для случая, когда A и B содержат не более 6 цифр, будут оценены не менее, чем в 20 баллов. Решения, правильно работающие для случая, когда A и B содержат не более 9 цифр, будут оценены не менее, чем в 40 баллов.

Offline-группа тестов оценивается в 40 баллов.

Задача С. Расписание турнира

Имя входного файла: `cin.txt`
Имя выходного файла: `cout.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Как известно, командные спортивные соревнования часто проводятся по круговой системе, когда любые две команды должны сыграть между собой ровно один матч. Круговой турнир проводится в несколько туров, в одном туре каждая команда может сыграть не более одного матча. Например, если в турнире участвуют 4 команды, то турнир можно провести в три тура: в первом туре команда 1 играет с командой 2, а команда 3 играет с командой 4, во втором туре 1 играет с 3, а 2 играет с 4, в третьем туре — 1 играет с 4, а 2 играет с 3.

Организаторам олимпиады Сочи-2014 необходимо организовать несколько командных турниров по круговой системе с участием различного числа команд. График олимпиады очень плотный, поэтому каждый турнир нужно провести в минимально возможное число туров. Для составления расписания каждого турнира они решили обратиться за помощью к программистам.

Формат входного файла

Во входном файле записано одно натуральное число N — количество команд, участвующих в турнире ($2 \leq N \leq 100$).

Формат выходного файла

В первой строке выведите минимальное количество туров K , необходимых для проведения кругового турнира из N команд. Каждая из K следующих строк содержит описание одного тура. В начале строки выведите количество игр n_i , которое необходимо сыграть в i -м туре. Далее идет n_i пар чисел — команды, которые играют в этом туре. Команды, играющие между собой, разделяются символом “-” (минус), а разные игры разделяются пробелом.

Примеры

<code>cin.txt</code>	<code>cout.txt</code>
4	3 2 1-2 3-4 2 1-3 2-4 2 1-4 2-3
3	3 1 1-2 1 2-3 1 3-1

Система оценивания

Online-группа тестов содержит тесты для $N \leq 10$ и оценивается в 40 баллов.
Offline-группа тестов оценивается в 60 баллов.

Задача D. Питон

Имя входного файла: `din.txt`
Имя выходного файла: `dout.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Витя изучает новый язык программирования Питон. Пока он только успел изучить арифметические операции и условную инструкцию `if`, но он уже полюбил этот язык за красоту и лаконичность синтаксиса.

Отличительной особенностью языка Питон является то, что блоки после инструкций `if` и `else` (а также в циклах `for` и `while`, но Витя еще не успел изучить циклы) выделяются не ключевыми словами (например, в языке Паскаль используются слова `begin` и `end`) и не скобками (например, в языке С используются фигурные скобки), а величиной отступа от начала строки, то есть количеством пробелов, которые идут в начале строки. Например, в такой программе:

```
if a < 0:
    print("Число a - отрицательное")
    a = -a
print("Теперь a - положительное")
```

если условие `a < 0` будет истинно, то выполнятся две следующие строки `print("Число a - отрицательное")` и `a = -a`, а вот следующая строка `print("Теперь a - положительное")` уже находится вне блока условной инструкции `if` и будет выполнена после этой условной инструкции независимо от истинности проверенного условия.

Более формально правила расстановки пробелов в программе такие. Первая строка программы, а также все инструкции в программе, если они не находятся внутри блоков условных инструкций, не содержат отступа, то есть пробелов в начале строки. Если в программе встречается условная инструкция `if`, то блок после этой инструкции пишется с отступом. Величина отступа может быть произвольной (1, 2, 3 и более пробелов), но для всех инструкций внутри блока отступ должен быть одинаковым. Если после инструкции `if` идет инструкция `else`, то она должна иметь такой же отступ, что и соответствующая ей инструкция `if`, после инструкции `else` идет блок из одной и более инструкций с дополнительным отступом. При этом отступ у блока `if` и блока соответствующего ему `else` может быть различным (смотрите примеры верных программ ниже), но внутри одного блока отступ должен быть одинаковым.

Каждой инструкции `if` может соответствовать не более одной инструкции `else`. Не допускаются инструкции `else`, перед которыми нет инструкции `if`. После каждой инструкции `if` и `else` обязательно следует хотя бы одна инструкция с отступом.

Также допускаются вложенные условные инструкции, у блоков вложенных условных инструкций отступ должен быть большим, чем у объемлющей инструкции, но при этом может быть произвольным.

Правильно (разрешается разный отступ в блоках <code>if</code> и <code>else</code>)	Правильно (вложенные условные инструкции)	Неправильно (разный отступ в одном блоке)	Неправильно (нет блока с отступом после инструкции <code>if</code>)
<pre>if x > 0: print("x > 0") print(x) else: print("x < 0") print(-x) print("Bye")</pre>	<pre>if a > b: if a > c: print(a) else: print(c) else: if b > c: print(b) else: print(c)</pre>	<pre>if x > 0: print(x) print("x > 0")</pre>	<pre>if x > 0: else: print(x)</pre>

Витя хочет написать компилятор языка Питон, и для начала он решил реализовать анализатор корректности расстановки отступов в условных инструкциях. Помогите ему в решении этой задачи.

Формат входного файла

Во входном файле записан некоторый текст, содержащий не более 100 строк. Длина каждой строки не превосходит 100 символов. Каждая строка состоит из символов, ASCII-коды которых не менее 32 и не более 126.

Строка считается инструкцией `if`, если первыми непробельными символами строки является слово `if`, после которого идет пробел, а затем — любое число любых символов. Строка считается инструкцией `else`, если она содержит только одно слово `else:` (с двоеточием после него) и, возможно, отступ в начале строки.

Любая строка содержит хотя бы один непробельный символ. Последняя строка программы обязательно содержит ровно одно слово `exit()` без пробелов, завершающееся символом конца строки.

Формат выходного файла

Если отступы в этой программе расставлены правильно, то программа должна вывести одно число 0. Если отступы расставлены неправильно, то нужно вывести минимальный номер строки, в которой нарушаются правила расстановки отступов.

Примеры

din.txt	dout.txt
<pre>a, b, c = map(int, input().split()) if a > b: if a > c: print(a) else: print(c) else: if b > c: print(b) else: print(c) exit()</pre>	0
<pre>x = int(input()) if x < 0: print("Negative") x = -x else: print("Positive") exit()</pre>	4

Система оценивания

Online-группа тестов оценивается в 50 баллов. Тесты этой группы не содержат инструкции `else`.

Offline-группа тестов оценивается в 50 баллов.

Программа, которая выдает правильный ответ только на тех примерах, в которых ответ 0, будет оцениваться в 0 баллов (то есть для получения ненулевого числа баллов за задачу программа должна выдавать правильный ответ хотя бы на одном тесте, помимо теста из условия, в котором ответ не 0).

Задача Е. К-квест

Имя входного файла: `ein.txt`
Имя выходного файла: `eout.txt`
Ограничение по времени: 0,5 секунды
Ограничение по памяти: 64 мегабайта

В одной из компьютерных игр-квестов есть следующее задание. На карте игрового мира размещены N персонажей, с каждым из которых может встретиться игрок. От общения с i -м персонажем карма игрока меняется на величину a_i , которая может быть как положительной, так отрицательной или даже нулем.

Изначально карма игрока равна нулю. Для того чтобы пройти на следующий уровень, нужно чтобы карма была в точности равна значению K , при этом карма также может принимать как положительные, так и отрицательные значения.

Комнаты, в которых находятся персонажи, соединены односторонними магическими порталами, поэтому игроку придется встречать персонажей в определенной последовательности: после персонажа номер i он попадает к персонажу номер $i + 1$, затем к персонажу номер $i + 2$, и т.д. В комнате последнего персонажа с номером N портала к другому персонажу нет.

Для перемещения между персонажами можно использовать еще и заклинания телепортации, но к сожалению у героя осталось всего лишь два свитка с заклинаниями. Поэтому один из этих свитков придется использовать для того, чтобы телепортироваться к любому из персонажей, а второй свиток — чтобы покинуть игровой мир, после того, как карма героя станет равна K .

Помогите игроку определить, в какую комнату надо телепортироваться в начале и из какой комнаты нужно покинуть игровой мир, чтобы достичь кармы K или сообщите, что это невозможно.

Формат входного файла

В первой строке входных данных записаны два числа: количество персонажей N и необходимый уровень кармы K ($|K| \leq 10^9$, $K \neq 0$). Во второй строке через пробел записаны N целых чисел a_1, a_2, \dots, a_N — величины, на которые меняется карма героя после общения с персонажами с номерами 1, 2, ..., N соответственно.

Формат выходного файла

Выведите номер комнаты, в которую надо войти игроку и номер комнаты, из которой надо выйти, чтобы набрать карму K . Если возможных вариантов несколько, то необходимо вывести самый короткий путь, а если и таких несколько, то путь, начинающийся в комнате с как можно большим номером. Если достичь кармы K последовательно общаясь с персонажами невозможно, то выведите одно число -1 .

Примеры

<code>ein.txt</code>	<code>eout.txt</code>
5 3 -2 2 -1 2 4	2 4
7 1 1 -1 1 -1 1 -1 2	5 5
4 3 2 2 2 2	-1

Система оценивания

Тесты по этой задаче разбиты на группы. На 1-3 группах тестов проверка проводится во время тура (online), на последней группе — после окончания тура (offline).

В первой группе тестов $1 \leq N \leq 100$, $|a_i| \leq 100$. Баллы начисляются только при прохождении всех тестов группы, группа оценивается в 20 баллов.

Во второй группе тестов $1 \leq N \leq 2000$, $|a_i| \leq 1\,000\,000$. Баллы начисляются только при прохождении всех тестов группы, группа оценивается в 20 баллов.

В третьей группе тестов $1 \leq N \leq 200\,000$, $0 \leq a_i \leq 10^9$. Баллы начисляются только при прохождении всех тестов группы, группа оценивается в 20 баллов.

В четвертой группе тестов $1 \leq N \leq 200\,000$, $|a_i| \leq 10^9$. Каждый тест этой группы оценивается отдельно. Общее число баллов за тесты этой группы равно 40.

Задача А. Олимпиада в Хогвартсе

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В Хогвартсе проходит традиционная ежегодная олимпиада по теории магии среди младшекурсников. Завхозу школы Аргусу Филчу поручили заняться распределением студентов по аудиториям.

Каждый факультет выставил своих лучших учеников на олимпиаду. От Гриффиндора участвует G студентов, от Слизерина S студентов, Пуффендуй представляет H студентов и Когтевран — R студентов. В распоряжении Филча находится M аудиторий. На аудитории наложено особое заклятие расширения, поэтому при необходимости они могут вместить любое количество студентов. При рассадке необходимо учесть, что ученики одного факультета, находящиеся в одной аудитории, могут, воспользовавшись случаем, начать жульничать, обмениваясь идеями по решению задач. Поэтому в любой аудитории количество студентов с одного факультета, попавших в нее, следует свести к минимуму. Назовем рассадку, удовлетворяющую такому требованию, оптимальной.

Помогите посчитать, какое минимальное количество студентов с одного факультета все же придется посадить в одной аудитории даже при оптимальной рассадке.

Формат входного файла

В первой строке идут четыре целых числа G , S , H и R ($1 \leq G, S, H, R \leq 1000$) — количество учеников, представляющих каждый из факультетов школы.

Во второй строке идет целое число M ($1 \leq M \leq 1000$) — количество классов в распоряжении у Филча.

Формат выходного файла

Выведите минимальное количество студентов с одного факультета, которое Филчу придется посадить в одну аудиторию даже при оптимальной рассадке.

Примеры

Входные данные	Выходные данные
4 3 4 4 2	2
15 14 13 14 5	3

Примечание

Пояснение к первому примеру. Из четырёх студентов, например, Гриффиндора двое в любом случае окажутся в одной аудитории, иначе в другой аудитории будет больше двух гриффиндорцев, что противоречит требованию минимизировать количество студентов одного факультета в каждой из аудиторий. Студентов Слизерина можно рассадить по двум аудиториям как 2 и 1, но в одной аудитории все равно придется посадить двоих, то есть и для Слизерина ответ 2. Ответ в данном случае два.

Задача В. Порядочные числа

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

У Миши развитое эстетическое чувство. Он считает, что не все числа одинаково порядочные. Когда ему грустно, он начинает придумывать числа и приводить их в порядок.

Миша очень любит рассматривать сумму цифр числа. Для того чтобы привести в порядок число A , он сначала записывает само число. Потом он пишет сумму цифр этого числа. Затем — сумму цифр суммы цифр и так далее, до тех пор, пока очередное число не станет однозначным. Он считает, что результатом приведения в порядок числа A является сумма всех выписанных чисел, включая само число A .

Миша настолько любит этот процесс, что он даже заменяет ему счёт овец, когда долго не получается заснуть. Он помнит, что вчера ночью, когда он в уме привёл в порядок число A , у него получилось число B . Но вот беда — он не помнит, какое именно он взял число A ! Помогите ему в отыскании этого числа.

Формат входного файла

На ввод подаётся единственное целое число B ($1 \leq B \leq 1\,000\,000\,000$)

Формат выходного файла

Если существует такое число A , что после приведения его в порядок, получается B , то выведите любое такое число. Если же Миша где-то ошибся в расчётах и такого числа не существует, то выведите -1.

Примеры

Входные данные	Выходные данные
42	29
20	-1

Примечание

Пояснение к первому примеру. Последовательность сумм цифр для 29 состоит из чисел 29, 11, 2. Соответственно, после приведения в порядок число 29 превращается в число $42 = 29 + 11 + 2$.

Задача С. Скучная лекция

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Лёша сидел на лекции. Ему было невероятно скучно. Голос лектора казался таким далеким и незаметным...

Чтобы окончательно не уснуть, он взял листок и написал на нём свое любимое слово. Чуть ниже он повторил своё любимое слово, без первой буквы. Ещё ниже он снова написал своё любимое слово, но в этот раз без двух первых и последней буквы.

Тут ему пришла в голову мысль — времени до конца лекции все равно ещё очень много, почему бы не продолжить выписывать всеми возможными способами это слово без какой-то части с начала и какой-то части с конца?

После лекции Лёша рассказал Максy, как замечательно он скоротал время. Максy стало интересно посчитать, сколько букв каждого вида встречается у Лёши в листочке. Но к сожалению, сам листочек куда-то запропастился.

Макс хорошо знает любимое слово Лёши, а ещё у него не так много свободного времени, как у его друга, так что помогите ему быстро восстановить, сколько раз Лёше пришлось выписать каждую букву.

Формат входного файла

На вход подаётся строка, состоящая из строчных латинских букв — любимое слово Лёши.

Длина строки лежит в пределах от 5 до 100 000 символов.

Формат выходного файла

Для каждой буквы на листочке Лёши, выведите её, а затем через двоеточие и пробел сколько раз она встретилась в выписанных Лёшей словах (см. формат вывода в примерах). Буквы должны следовать в алфавитном порядке. Буквы, не встречающиеся на листочке, выводить не нужно.

Примеры

Входные данные	Выходные данные
hello	e: 8 h: 5 l: 17 o: 5
abacaba	a: 44 b: 24 c: 16

Примечание

Пояснение к первому примеру. Если любимое Лёшино слово — **"hello"**, то на листочке у Лёши будут выписаны следующие слова: **"hello"**, **"hell"**, **"ello"**, **"hel"**, **"ell"**, **"llo"**, **"he"**, **"el"**, **"ll"**, **"lo"**, **"h"**, **"e"**, **"l"**, **"l"**, **"o"**.

Среди этих слов 8 раз встречается буква **"e"**, 5 раз — буква **"h"**, 17 раз — буква **"l"** и 5 раз буква **"o"**.

Задача D. Диадема Клеопатры

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Аня — страстный любитель ювелирных изделий. Ее коллекция насчитывает множество бриллиантов, изумрудов и алмазов.

...Срочная новость! Бесценный змеиный рубин Клеопатры был украден!

Три дня назад мир потрясло сенсационное известие: исследовательская экспедиция обнаружила в одном из храмов, построенных во времена великой египетской императрицы Клеопатры, потайную комнату. В ней кроме бронзовой статуи императрицы обнаружилась поразительной красоты диадема, ранее считавшаяся бесследно утерянной! Ученые сообщили, что диадема увенчана алым а-каратным рубином в форме змеиной головы. Однако буквально пару часов назад поступила новость, что бесценное украшение было изувечено: кто-то пробрался в камеру хранения диадемы и вырезал из нее рубин! Полиция устанавливает круг подозреваемых...

Услышав о краже рубина, Аня сразу бросилась исследовать информацию на черных рынках. В течение дня она обнаружила N объявлений о продаже рубина в форме змеиной головы, утверждающих что это именно украденная древняя ценность. Аня не простит себе, если она упустит такой бесценный экспонат для своей коллекции, поэтому она приказала своему помощнику Глебу срочно купить все эти камни, надеясь приобрести среди них настоящую реликвию.

Купив все N камней, Глеб тут же провел несколько пробных измерений, взвесив некоторые наборы из них, и отправил результаты Ане по электронной почте. Тем временем она проконсультировалась с известным исследователем старины Андрэ Шесто-Мерта по поводу украденной драгоценности и узнала, что по всем имеющимся историческим источникам рубин весил не a карат, как утверждали журналисты, а b карат!

Зная результаты взвешиваний Глеба, и учитывая, что все поддельные камни весят a карат, и только настоящий змеиный рубин может весить b карат, определите, какие из купленных камней могут на самом деле являться потерянной реликвией великой императрицы прошлого.

Формат входного файла

В первой строке находятся четыре целых числа N , a , b и K ($1 \leq N \leq 200$, $1 \leq a, b \leq 1\,000\,000$, $a \neq b$, $1 \leq K \leq 1\,000$).

Далее идут K строк, описывающих взвешивания, проведенные Глебом.

Первое число в i -ом описании — w_i ($1 \leq w_i \leq 200\,000\,000$), суммарный вес группы камней, участвовавших в i -ом взвешивании.

Второе число — m_i ($1 \leq m_i \leq N$) — количество камней, участвовавших в i -ом взвешивании.

Далее следуют m_i целых чисел, упорядоченных по возрастанию, — номера камней, участвовавших в i -ом взвешивании.

Формат выходного файла

Если среди купленных Глебом камней змеиного рубина точно нет, выведите строку "Fail" (без кавычек).

Если украденный рубин может присутствовать среди камней, то выведите в первой строке количество всех возможных кандидатур на роль древней реликвии, а второй строке — номера возможных вариантов. Выводить номера можно в произвольном порядке.

Если же Глеб в некоторый момент ошибся в расчетах, и присланная им информация о взвешиваниях не может соответствовать действительности, выведите строку "Impossible" (без кавычек).

Примеры

Входные данные	Выходные данные
4 15 17 2 30 2 1 3 47 3 2 3 4	2 2 4
3 15 17 3 30 2 1 2 30 2 2 3 47 3 1 2 3	Impossible
2 1 2 2 1 1 2 1 1 1	Fail

Примечание

В первом тесте из первого взвешивания мы делаем вывод, что первый и третий камни гарантированно поддельные. С другой стороны, среди второго, третьего и четвёртого камня точно есть настоящий. Значит настоящим может оказаться второй или четвёртый камень.

Во втором тесте Глеб ошибся в измерениях, потому что из первых двух измерений следует, что все камни фальшивые, а из последнего — что настоящий камень, тем не менее, среди них присутствует.

В третьем тесте из результатов явно следует, что оба приобретенных камня фальшивые.

Задача Е. Кружки в Маховниках

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Маленький Петя очень любит компьютеры и хочет научиться программировать. В небольшом городке Маховники, где он живёт, работает сеть кружков по программированию самой разной тематики. Когда Петя пошёл записываться, он увидел большой список, состоящий из N кружков. Петя хочет быть всесторонне развитой личностью, поэтому он собрался отучиться во всех этих кружках. Но когда он собрался записаться на все занятия сразу, обнаружилось, что не всё так просто. Во-первых, в один момент времени разрешается учиться только в одном из этих N кружков. Во-вторых, некоторые преподаватели выдвигают входные требования к знаниям учеников, заключающиеся в знании курсов каких-то других кружков!

Петя хочет стать великим программистом, поэтому подобные мелочи его не останавливают. Действительно, ему достаточно всего-лишь составить правильный порядок посещения кружков, чтобы удовлетворить всем входным требованиям — это совсем простая задача, доступная даже совсем неопытному программисту.

Перед тем как сесть составлять порядок посещения кружков, Петя внимательно перечитал условия обучения и обнаружил ещё один важный пункт. Оказывается, для привлечения школьников, во всех кружках действует система поощрения учеников конфетами. Это означает, что по окончании очередного кружка ученику выдают несколько коробок конфет, всё больше и больше с каждым пройденным кружком. С другой стороны, в каждом кружке количество конфет в коробке своё, зависящее от сложности курса. Более конкретно — за прохождение i -го по счёту кружка, если этот кружок идёт в общем списке под номером j , ученику выдают аж $N^{i-1} \cdot j$ конфет — такие щедрые люди программисты. Петя решил совместить полезное с приятным — теперь он хочет выбрать такой порядок посещения кружков, чтобы при этом получить как можно больше конфет, однако эта задача ему уже не под силу. Помогите будущему великому человеку отыскать такой порядок.

Формат входного файла

В первой строке входного файла содержится целое число N ($1 \leq N \leq 100\,000$) — количество кружков в Маховниках. В последующих N строках идут описания входных требований кружков, в порядке их следования в общем списке. В i -ой строке сначала записано целое число k_i ($0 \leq k_i \leq N - 1$) — количество кружков, в которых нужно отучиться перед записью в i -й кружок, а потом k_i номеров этих кружков. Сумма k_i не превосходит 200 000.

Гарантируется, что возможно посетить все эти кружки в некотором порядке, не нарушая условия посещения.

Формат выходного файла

Выведите N номеров, разделённых пробелами — порядок, в котором Пете надо посещать кружки, чтобы съесть как можно больше конфет.

Примеры

Входные данные	Выходные данные
6	2 1 3 5 4 6
1 2	
0	
1 2	
3 1 2 5	
1 2	
4 1 3 4 5	

Примечание

Пояснение к примеру. Посещая кружки в указанном порядке, Петя получит $6^0 \cdot 2 + 6^1 \cdot 1 + 6^2 \cdot 3 + 6^3 \cdot 5 + 6^4 \cdot 4 + 6^5 \cdot 6 = 2 + 6 + 108 + 1080 + 5184 + 46656 = 53036$ конфет.

Задача А. Гонорар

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Инопланетяне с планеты Пандора уже много лет исследуют землян. Для этого на Землю введено множество агентов-пандорианцев. Так как среднестатистический пандорианец примерно в 20 раз меньше среднестатистического землянина, для перемещения по планете они используют специальных роботов, внешне неотличимых от жителей Земли. Внутри этих роботов с комфортом располагается команда агентов-исследователей.

Сегодня команда, управляющая роботом «Геннадий», получает свой гонорар. Команда состоит из двух пандорианцев, Джейка и Джейка, которые в данный момент исследуют жителей России. Гонорар исследователи получают один на двоих наличными в обыкновенном российском банке.

Джейк и Джейк должны получить сегодня K рублей. Смогут ли они разделить полученные монеты и купюры на двоих так, чтобы гонорар оказался поделен поровну, вне зависимости от того, какими именно монетами и купюрами им решат выдать K рублей в банке?

Напишите программу, которая поможет Джейку и Джейку ответить на этот вопрос.

Формат входного файла

На вход подается число K — сумма, которую получают Джейк и Джейк ($1 \leq K \leq 100\,000$).

Формат выходного файла

Выведите «YES», если вне зависимости от того, какими именно монетами и купюрами будет выдана нужная сумма, их можно будет поделить поровну, и «NO» — в противном случае. Действие происходит в России, поэтому для выдачи нужной суммы могут быть использованы купюры и монеты следующих номиналов: 1, 2, 5, 10, 50, 100, 500, 1000 и 5000 рублей.

Примеры

Входные данные	Выходные данные
7	NO
24	YES
10	NO

Примечания и система оценки

Тесты к этой задаче состоят из шести групп. За каждую группу баллы начисляются только при прохождении всех тестов группы.

- Тесты 1–3. Тесты из условия, оцениваемые в ноль баллов.
- Тесты 4–9. В тестах этой группы $K < 10$. Эта группа оценивается в 20 баллов,
- Тесты 10–18. В тестах этой группы $10 \leq K < 100$. Эта группа оценивается в 20 баллов.
- Тесты 19–29. В тестах этой группы $100 \leq K < 1000$. Эта группа оценивается в 20 баллов.
- Тесты 30–42. В тестах этой группы $1000 \leq K < 10\,000$. Эта группа оценивается в 20 баллов.
- Тесты 43–51. В тестах этой группы $10\,000 \leq K \leq 100\,000$. Эта группа оценивается в 20 баллов. Решение будет тестироваться на тестах этой группы **offline**, т. е. после окончания тура.

Тестирование на тестах каждой группы производится вне зависимости от прохождения всех тестов из предыдущих групп.

Задача В. Воды слонам!

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Кроме Земли, пандорианцы уже много тысячелетий исследуют и другие планеты. Большой интерес для них в прошлом представляла планета Арракис. К сожалению, с началом исследований на Земле финансирование исследований на Арракисе было существенно урезано, и местным агентам-исследователям пришлось искать дополнительные источники дохода.

К счастью, пандорианцы очень хорошо разбираются в финансовых вопросах. Им не составило труда проанализировать политические, экономические и психологические тенденции, а также некоторые другие факторы, не имеющих названий на земных языках и на основе этих данных точно предсказать изменение стоимости воды на Арракисе на ближайший год. Как известно, вода на этой планете является главной ценностью после золота, на которое эту воду можно купить.

Изначально пандорианцы обладают некоторым запасом золота. Они решили в один из дней года купить на все это золото воды, а в какой-то последующий день продать всю купленную воду и получить прибыль за счет разницы стоимости. Конечно же, пандорианцы хотят максимизировать свой доход в результате этих махинаций. Помогите им выбрать оптимальные дни для покупки и продажи воды!

Формат входного файла

В первой строке задано целое число N — количество дней в году на планете Арракис.

Во второй строке заданы N целых положительных чисел a_i ($1 \leq i \leq N, 1 \leq a_i \leq 5000$), задающих стоимость воды на Арракисе в день i .

Формат выходного файла

Выведите два целых числа: первое число — номер дня, в который стоит купить воду, второе число — номер дня, в который следует воду продать. Дни нумеруются с единицы. Если оптимальных пар дней для покупки/продажи несколько, то выведите любую из них.

Выведите два нуля, если покупка и продажа воды по указанной схеме не принесет пандорианцам прибыли.

Примеры

Входные данные	Выходные данные
6 10 3 5 3 11 9	2 5
4 5 5 5 5	0 0

Примечания и система оценки

Тесты к этой задаче состоят из трёх групп.

- Тесты 1–2. Тесты из условия, оцениваются в ноль баллов.
- Тесты 3–7. В тестах этой группы $2 \leq N \leq 1000$. Тесты в этой группе оцениваются **независимо**. Каждый успешно пройденный тест оценивается в 10 баллов.
- Тесты 8–12. В тестах этой группы $2 \leq N \leq 100\,000$. Решение будет тестироваться на тестах этой группы **offline**, т. е. после окончания тура. Тесты в этой группе оцениваются **независимо**. Каждый успешно пройденный тест оценивается в 10 баллов.

Задача С. Будьте осторожны при купании!

Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Инопланетяне с планеты Пандора продолжают изучать жителей Земли. В этот раз они хотят проверить, влияет ли цвет глаз землянина на его зрение. Запланированное исследование состоит из серии экспериментов, для каждого из которых требуется двое землян с глазами одинакового цвета.

В поисках материала для исследования пандорианцы на своей летающей научной базе прибыли на обыкновенный курортный пляж, до отказа забитый греющимися на солнце туристами. Туристы занимают все доступное место на пляже, образуя, таким образом, прямоугольник размером $N \times M$ человек. По пляжу они не перемещаются. То есть, каждый турист однозначно задается своими координатами в этом прямоугольнике: номером ряда, в котором он лежит, и номером в этом ряду.

Научная база пандорианцев оборудована двумя специальными антигравитационными лучами, предназначенными для подъема грузов на базу и для их спуска на поверхность близлежащей планеты. Чтобы база не потеряла равновесия, оба луча следует использовать одновременно. Таким образом, пандорианцы могут либо поднять двух произвольных туристов с пляжа на базу, либо опустить двух туристов обратно на пляж.

Исследование должно проходить следующим образом. Оператор антигравитационного луча выбирает на пляже двоих туристов и поднимает их на базу. После этого специальная аппаратура определяет цвет глаз каждого из них. Если полученные цвета совпадают, ученые уводят эту пару туристов для проведения очередного эксперимента, и на пляж они больше не возвращаются. Если же они не совпадают, оператор опускает этих туристов обратно на их места на пляже, после чего процесс повторяется.

Обычно такие исследования не вызывали никаких трудностей, но в этот раз ученые заявили оператору, что после завершения исследования на пляже не должно остаться ни одного землянина. Помогите оператору выбирать туристов в таком порядке, чтобы это требование оказалось выполненным. Гарантируется, что это возможно, то есть туристов можно разбить на пары так, чтобы цвет глаз в каждой паре совпадал.

Это интерактивная задача. В процессе тестирования программа-решение будет взаимодействовать с использованием стандартных потоков ввода/вывода с программой-интерактором, сообщаемой в ответ на координаты двух очередных выбранных туристов цвета их глаз.

Формат входного файла

В первой строке заданы два числа: N и M — размеры прямоугольника, который туристы образуют своими телами, где N — это количество рядов, а M — количество человек в одном ряду пляжа. Гарантируется, что туристов на пляже четное число.

Затем программа-решение начинает взаимодействие с программой-интерактором в соответствии со следующим протоколом:

- Программа-решение выводит в стандартный поток вывода одну строчку, описывающую, каких туристов оператор поднимает с пляжа в этот раз. Строчка должна содержать четыре целых числа x_1, y_1, x_2, y_2 , где x_1 — номер ряда, в котором лежит первый турист, y_1 — его номер в этом ряду, а x_2 и y_2 — координаты второго туриста, заданные аналогично. Ряды нумеруются сверху вниз, начиная с 1, туристы в них — слева направо, начиная с 1. **Вывод должен завершаться переводом строки и сбросом буфера потока вывода.** Для последнего используйте
 - `flush(output)` в паскале или Delphi;
 - `fflush(stdout)` или `cout.flush()` в C/C++;
 - `sys.stdout.flush()` в Python;
 - `Console.out.flush()` в Visual Basic.
- После этого программа должна считать из стандартного потока ввода ответ программы-интерактора. Ответ состоит из двух натуральных чисел, не превышающих 200 000 — цвета

глаз первого и второго туриста соответственно. Отправка туристов обратно на пляж или на исследование в соответствии с этими данными происходит автоматически. Если их забирают ученые, больше эти земляне на пляж не возвращаются. Их места на пляже остаются пустыми до конца исследования.

- Программа-решение должна завершить работу, когда на пляже не останется ни одного туриста. Гарантируется, что организовать выбор туристов так, чтобы это условие оказалось выполненным, всегда возможно. Количество запросов не должно превышать 262 144.

Примеры

Входные данные	Выходные данные
2 3	1 1 1 2
1 20	1 3 2 1
3 1	2 1 1 1
1 1	2 2 2 3
20 3	2 2 1 2
20 20	2 3 1 3
3 3	

Примечания и система оценки

Тесты к этой задаче состоят из пяти групп.

- Тест 1. Тест из условия, оценивается в ноль баллов.
- Тесты 2–7. В тестах этой группы глаза у всех туристов одного цвета, $1 \leq N, M \leq 50$. Эта группа оценивается в 30 баллов, баллы начисляются только при прохождении всех тестов группы.
- Тесты 8–13. В тестах этой группы $1 \leq N, M \leq 50$. Эта группа оценивается в 30 баллов, баллы начисляются только при прохождении всех тестов группы.
- Тесты 14-17. В тестах этой группы $1 \leq N, M \leq 200$. Эта группа оценивается в 20 баллов. Решение будет тестироваться на тестах этой группы **offline**, т. е. после окончания тура. Тесты в этой группе оцениваются **независимо**.
- Тесты 18-21. В тестах этой группы $1 \leq N, M \leq 500$. Эта группа оценивается в 20 баллов. Решение будет тестироваться на тестах этой группы **offline**, т. е. после окончания тура. Тесты в этой группе оцениваются **независимо**.

Тестирование на тестах каждой группы производится только в случае прохождения всех тестов из **всех предыдущих** групп.

Задача D. Паровозик из Морошково

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

После своего первого контакта с землянами обитатели планеты Пандора решили всё-таки перенять часть земных технологий. В частности, они предприняли попытку приспособить к особенностям своей планеты систему железнодорожного сообщения.

Построенная пандорианцами железная дорога представляет собой прямой отрезок, проходящий в непосредственной близости от N красивейших водопадов. Паровозик, который планируется запустить по этой дороге, будет без остановок проезжать весь маршрут, преодолевая участки между водопадами за строго определённое время.

Так как жители Пандоры очень трепетно относятся к красотам своей природы, они не хотят упускать ни малейшей возможности полюбоваться водопадами, искрящимися в дневном свете. Грамотно составлять расписания пандорианцы ещё не научились, и теперь они обратились за помощью к вам, участникам олимпиады по программированию.

Помогите обитателям Пандоры определить, в какое время дня паровозик должен проехать рядом с первым водопадом, чтобы пассажиры смогли увидеть все водопады на пути в дневное время, при этом не обязательно в один и тот же день. Водопад виден лишь в момент проезда паровозика мимо него, но, как известно, пандорианские водопады настолько впечатляющие, что и за мгновение можно сполна насладиться любым из них.

Формат входного файла

В первой строке через пробел вводятся два натуральных числа: количество часов в одних сутках (H) и минут в одном часу (M) на Пандоре ($1 \leq H, M \leq 500$).

Следующая строка содержит четыре целых числа, описывающих время начала (H_s, M_s) и конца (H_f, M_f) светового дня ($0 \leq H_s, H_f < H; 0 \leq M_s, M_f < M$). При этом либо $H_s < H_f$, либо $H_s = H_f$ и $M_s < M_f$ (гарантируется, что день начинается раньше, чем заканчивается). Если паровозик проезжает мимо водопада ровно в H_s часов M_s минут или ровно в H_f часов M_f минут, то считается, что он проехал мимо водопада днём.

Третья строка содержит одно натуральное число N — количество водопадов, рядом с которыми проезжает паровозик ($1 \leq N \leq 100\,000$).

В следующих $N - 1$ строках вводятся по 2 целых числа H_i и M_i , описывающих продолжительность временных интервалов для проезда между соседними водопадами: H_1, M_1 — время в пути между первым и вторым водопадами, H_2, M_2 — между вторым и третьим и так далее. Гарантируется, что время, затрачиваемое на дорогу между любыми двумя соседними водопадами, строго положительно, не превосходит одних пандорианских суток и записано корректно: $0 \leq H_i \leq H$, $0 \leq M_i < M$.

Формат выходного файла

Если составить подходящее расписание невозможно, то в качестве ответа выведите одно слово «Impossible» (без кавычек). Иначе выведите два числа H_0 и M_0 , разделённые пробелом, описывающие любое подходящее время проезда паровозика рядом с первым водопадом.

Примеры

Входные данные	Выходные данные
24 60 8 0 22 0 6 6 0 21 0 19 0 12 0 10 0	12 0
24 60 8 17 20 10 2 11 59	Impossible

Примечания и система оценки

Тесты к этой задаче состоят из четырех групп.

- Тесты 1–2. Тесты из условия, оцениваемые в ноль баллов.
- Тесты 3–17. В тестах этой группы $H = 24$, $M = 60$ и $N \leq 1000$. Эта группа оценивается в 30 баллов, баллы начисляются только при прохождении всех тестов группы.
- Тесты 18–38. В тестах этой группы $H \leq 80$, $M \leq 100$, $N \leq 100000$. Эта группа оценивается в 30 баллов, баллы начисляются только при прохождении всех тестов группы.
- В тестах этой группы дополнительные ограничения отсутствуют. Эта группа оценивается в 40 баллов. Решение будет тестироваться на тестах этой группы **offline**, т. е. после окончания тура. Тесты в этой группе оцениваются **независимо**.

Тестирование на тестах каждой группы производится только в случае прохождения всех тестов из **всех предыдущих** групп.

Задача Е. Макемакеанские пирамиды

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Как известно, знаменитые египетские пирамиды были построены инопланетянами. Именно они послужили толчком к развитию цивилизации на Земле. Но мало кто знает, что этими инопланетянами был пандорианцы. Теперь они хотят повторить свой успех на планете Макемаке, одной из карликовых планет Солнечной системы.

Для постройки пирамид на Макемаке были завезены и расставлены в ряд N каменных блоков различных типов. Всего существует 9 типов блоков. Тип блока определяется его размером: самые большие блоки имеют тип 9, а самые маленькие — 1. Правильная пирамида должна состоять из поставленных друг на друга блоков, причем сверху обязательно должен быть блок типа 1, а каждый блок должен стоять на блоке следующего по величине типа.

Конечно, пирамиды строят не сами пандорианцы, а местное население Макемаке. Пандорианцы лишь руководят строительным процессом, указывая, какой блок нужно двигать. Особенности анатомии макемакеанцев позволяют им поднять один блок и поставить его на первый встреченный **справа** блок или стопку блоков. Как только очередная пирамида оказывается достроенной (то есть на ней сверху оказывается блок типа 1), она вывозится из ряда блоков и устанавливается на специально подготовленную для нее площадку.

Разработайте стратегию постройки пирамид, при которой неиспользованных блоков останется как можно меньше.

Формат входного файла

В первой строке задано целое число N — количество завезенных блоков ($1 \leq N \leq 100\,000$).

Во второй строке даны N целых чисел от 1 до 9 — типы блоков в том порядке, в котором они стоят в ряду, перечисленные слева направо.

Формат выходного файла

Выведите минимально возможное число неиспользованных блоков.

Примеры

Входные данные	Выходные данные
3 3 1 2	1
4 2 1 3 1	2
7 1 2 3 1 2 4 5	0

Примечания и система оценки

Во втором примере можно построить только две пирамиды высотой 1.

В третьем примере можно из стоящих в середине блоков 1 2 построить пирамиду высотой 2, а из оставшихся блоков — пирамиду высотой 5.

Тесты к этой задаче состоят из трёх групп.

- Тесты 1–3. Тесты из условия, оцениваются в ноль баллов.
- Тесты 4–18. В тестах этой группы $1 \leq N \leq 2000$. Тесты в этой группе оцениваются **независимо**. Каждый успешно пройденный тест оценивается в 4 балла.
- Тесты 19–28. В тестах этой группы $1 \leq N \leq 100\,000$. Решение будет тестироваться на тестах этой группы **offline**, т. е. после окончания тура. Тесты в этой группе оцениваются **независимо**. Каждый успешно пройденный тест оценивается в 4 балла.