

Миронов Илья Лазаревич

Х ВСЕРОССИЙСКАЯ ОЛИМПИАДА ШКОЛЬНИКОВ ПО ИНФОРМАТИКЕ

В десятый раз прошла Всероссийская олимпиада школьников по информатике. Во второй раз ее хозяином был Санкт-Петербург. 116 школьников из большей части субъектов Российской Федерации бились в двух турах, борясь за каждый балл и за каждый тест. И как итог насыщенной и увлекательной недели остались два листочка с таблицей результатов, где каждому участнику указано его место в общем зачете. Эти результаты нам предстоит еще оценить, чтобы приобретенный опыт и знания дошли до школьников и учителей всей страны.

Олимпиада проходила в два тура, разделенных днем культурной программы для школьников. На каждом туре было предложено три задачи разной сложности и на разные темы. В условии каждой задачи указывалась ее максимальная оценка в баллах, поэтому участники могли распределять свои усилия, исходя из той оценки, которую они рассчитывали получить за задачу. Кроме этого, в условии приводится обычно и максимальное время тестирования задачи на каждом тесте, которое позволяет судить об эффективности ожидаемого алгоритма. Во Всероссийской олимпиаде принимают участие как опытные участники подобных соревнований,

так и новички. Так как чтение условия задачи – целое искусство, которое можно постигать только практикой, то и реакция участников на выданные условия оказалась различной.

Для обеспечения равных возможностей и наиболее справедливых критериев оценки жюри олимпиады требует от решений участников скрупулезного следования условиям задач, а особенно форматам входных и выходных данных.

Для тех участников, которым это требование знакомо, следовать ему не составляет труда. К сожалению, всегда находятся школьники, которые по невнимательности или от незнания отклоняются от соблюдения точного условия задачи. Конечно, бывает обидно получать ноль баллов за правильный ответ, который выводится на экран вместо оговоренного в условии текстового файла. Но, на взгляд жюри, единственный способ обеспечить одинаковый подход ко всем участникам состоит в том, чтобы проверять их решения на точное соответствие условию задачи. Тем более, что обязательное соблюдение требований заказчика есть непреложный закон программирования в производственных структурах, а умением тщательно соблюдать спецификации должен обладать любой профессиональный программист.



В первой задаче первого тура олимпиады предлагалось построить хеш-функцию, дающую минимальное число коллизий на подмножествах данного множества слов русского языка. Хотя это и не оговаривалось в условии, но на “хорошую” хеш-функцию налагались и дополнительные условия, такие как ее качественная работа на словах, стоящих близко в алфавитном порядке, или на словах с одинаковыми окончаниями. Эти требования можно было вывести из того, что решение должно было тестироваться на некотором наборе тестов, представляющих собой подмножества данного списка слов. Среди этих подмножеств обязательно должны были бы встретиться случайные выборки, а также выборки, подчиняющиеся определенным закономерностям. Несколько наиболее естественных из них и сформировали тестовый набор.

Решение, в котором была запрограммирована какая-нибудь разумная хеш-функция, получало за эту задачу, по меньшей мере, половину баллов. Но для того, чтобы гарантировать хорошие результаты на специальных тестах, а также на большом случайном тесте требовалось учитывать заданный набор слов. Например, можно было бы разбить данное множество из 10 тысяч слов на подмножества по 40 слов. Если выбирать в качестве подмножеств слова, идущие подряд по алфавиту, то программа может показать очень плохой результат на тесте с последовательной выборкой из файла, где слова расположены в алфавитном порядке. Поэтому можно предварительно применить к словам какую-нибудь “перепутывающую” функцию, запомнить 254 слова, разделяющие область значений этой функции на 255 интервалов, и присваивать словам, попавшим в один такой интервал, одинаковые значения хеш-функции. Такой подход обеспечивает хорошее поведение получившейся функции на естественных тестах.

Хотя эта задача и оказалась непривычной для большинства участников, было сдано много решений, получивших за нее высокие баллы.

Вторая задача первого тура алгоритмически не очень сложна, но требует аккуратного программирования и внимательного разбора крайних случаев. Правильное решение можно сформулировать так: необходимо продолжить в обе стороны все засеки (делая это последовательно, а не одновременно) до их пересечения с границей прямоугольника или с другими засеками. Получившееся разбиение будет удовлетворять условию задачи.

Большое количество неполных или частичных решений этой задачи вскрыло другую беду участников олимпиады. Запрограммировав правильный алгоритм, школьник считает, что программа готова, и приступает к решению следующей задачи. К несчастью, обычно это оказывается далеко не так. Любую программу следует тщательно протестировать, особенно внимательно проанализировав ее работу в граничных и крайних случаях. Если это не сделать во время тура, то этим с удовольствием займется жюри, фиксируя ошибки и снимая баллы. До школьников необходимо довести известный факт: в серьезных проектах до 40% времени (и уж никак не меньше 25%) отводится на тестирование программного продукта. По нашим наблюдениям, неспособность тестировать собственные программы есть столь же значимая причина, что и неумение читать условия задач, по которой школьники показывают результаты хуже тех, на которые они могли бы рассчитывать.

Третья задача первого тура предполагала наличие определенной математической культуры у школьников. Построение возможного решения начинается с произвольного распределения дорог между компаниями. Если это решение не удовлетворяет условию задачи, то это означает, что существует город, из которого выходят три дороги, принадлежащие одной компании. Отдадим одну из этих дорог другой компании и рассмотрим город, в который ведет эта дорога. Если теперь в нем нарушается условие антимонопольного комитета, то выберем одну из дорог, выходящих из него (но не ту, по

которой только что прошли), и сменим ее владельца. Будем поступать так до тех пор, пока мы не попадем в город, из которого выходят дороги, принадлежащие разным компаниям. Легко показать, что процесс остановится, а значит, искомое решение будет построено.

Как и первый тур, второй тур начинался с нестандартной для всероссийских олимпиад задачи. В ней школьникам предлагалось написать программу, играющую в игру с программой жюри. Игровая программа жюри во время тура участникам не выдавалась, зато в их распоряжение предоставлялся тренажер, позволяющий играть человеку или программе с другим человеком или программой.

Так как у жюри олимпиады было существенно больше времени для написания и отладки своей программы, то неудивительно, что она выигрывала на случайных полях почти у всех программ участников олимпиады. Стратегия, которую она применяла, разбивается на следующие шаги:

- сначала программа прорывается по кратчайшему пути на половину противника (если быть более точным, то на диагональ ниже главной диагонали);
- затем программа стремится к боковым сторонам, противоположным ее углу;
- на завершающем этапе игры программа жюри применяет перебор на четыре хода в глубину для определения наиболее выгодного очередного хода.

Это решение весьма трудоемкое, использующее разные алгоритмы на разных стадиях игры, действительно обычно обыгрывало решения участников. Тем не менее, десятиклассник из Орловской области сумел получить за эту задачу 48 баллов из 50 возможных.

Пятая задача олимпиады допускала два возможных решения. Аналитическое решение, хотя и не очень простое в реализации, гарантирует точный результат. Решение, находящее искомый промежуток между отправлением поездов дихотомией, делением интервала пополам, приводит только к приближительному ответу. Один

из тестов был выбран так, чтобы программа, использующая метод половинного деления, не проходила по времени.

Задача про бал позиционировалась как утешительная задача, с изящным решением и легкая в программировании. Действительно, дама, стоящая в углу, заменяет двух дам, располагающихся около стен. Следовательно, одна дама в углу как бы увеличивает количество дам на одну. Если бы все дамы стояли около стен, то их количество делилось бы на количество стен. Отсюда легко вычислить количество дам, стоящих в углах. Оно должно дополнять N до числа, делящегося на M . Например, в углах может стоять 0 дам, если N делится на M , и $M - (N \bmod M)$ в противном случае. Решение отсутствует, если $2 * N < M$.

Несколько слов необходимо сказать о проверке решений участников. Если прикинуть объем необходимых работ, то становится очевидно, что без автоматизированной системы не обойтись. На самом деле проверка решений могла бы происходить вообще без участия человека, если бы не непредвиденные случаи, которыми всегда столь богата олимпиада. Решение запускалось на тестирование, после чего проверяющая система автоматически подавала ему на вход тесты, прекращала исполнение в случае превышения времени тестирования, проверяла выходные файлы и записывала результаты тестирования в особый каталог. Человеческий фактор был полностью исключен, а вероятность ошибки сведена к минимуму. Стоит отметить, что подобная система проверки функционирует и на международных олимпиадах школьников.

Благодаря такой организации проверки, апелляция, традиционно самое нервное мероприятие олимпиады, проходит быстро и сводится лишь к повторному тестированию некоторых решений и к голосованию жюри по спорным случаям.

Результаты юбилейной всероссийской олимпиады оказались по-хорошему уникальными. Впервые в Российской Федерации появился двукратный чемпион

среди школьников по информатике – десятиклассник из Нижнего Новгорода Владимир Мартьянов (он же – абсолютный чемпион мира 1997 года). Во-вторых, первый раз за всю историю всероссийских олимпиад ни один участник не набрал нуля баллов. Приятно отметить и высокий итоговый балл победителей, набранный в нелегкой борьбе на сложных задачах, и успех невыпускных классов, характеризующийся тем, что в первой десятке одиннадцатиклассников оказалось меньше по-

ловины. Призы, которые предоставил генеральный спонсор олимпиады Инкомбанк и фонд “Открытое общество”, оказались вполне достойными высокого уровня знаний и подготовки победителей олимпиады. Нам осталось лишь пожелать успеха сборной России на Международной олимпиаде школьников и надеяться на то, что результат предыдущего года будет, по крайней мере, повторен: первое командное место и первое место в общем зачете.

Задача 1. Хеш-функция

имя входного файла: hash.in
имя выходного файла: hash.out
время на один ход: 20 секунд
максимальный балл: 34
автор: И. Миронов

Для быстрого распознавания идентификаторов в современных трансляторах используются *хеш-функции*. Хеш-функция сопоставляет строке целое число из некоторого промежутка. Критерием оценки хеш-функции является количество *коллизий*, то есть ситуаций, когда различным встретившимся строкам сопоставляется одно и то же число.

Вы должны построить хеш-функцию для заданного множества слов русского языка и реализовать ее в программе, которая будет вычислять хеш-значения из интервала от 1 до 255 для всех слов из входного файла. Ваша функция будет оцениваться в зависимости от максимального количества слов, хеш-значения которых оказались одинаковыми: чем это количество меньше, тем лучше функция.

На время тура вам предоставляется файл words.txt, содержащий перечисленные в алфавитном порядке 10 000 слов. Слова из этого файла будут использоваться при тестировании вашего решения. Это означает, что во всех тестах слова будут выбираться только из этого множества. Во время тестирования файл words.txt будет вашей программе недоступен.

Для одного и того же слова хеш-значение, формируемое вашей программой, должно быть одинаковым для всех тестов. В противном случае задача считается нерешенной.

Размер исходного текста вашей программы не должен превышать 10 килобайт. Программы большего размера оцениваться не будут.

Входные данные

Входной файл содержит список неповторяющихся слов по одному в строке, выбранных из файла words.txt. Регистр букв в словах будет тем же, что и в файле words.txt.

Выходные данные

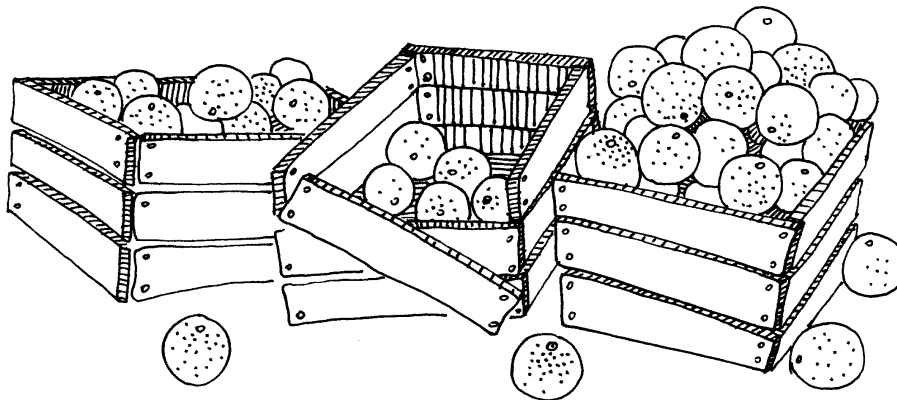
Для каждого слова из заданного списка в выходной файл требуется вывести соответствующее хеш-значение. Числа должны быть записаны по одному в строке.

Пример входного файла hash.in

мертвые
души
кому
на
Руси
жить
хорошо
герой
нашего
времени

Пример выходного файла hash.out

8
5
5
3
5
5
7
6
7
8



Примечание

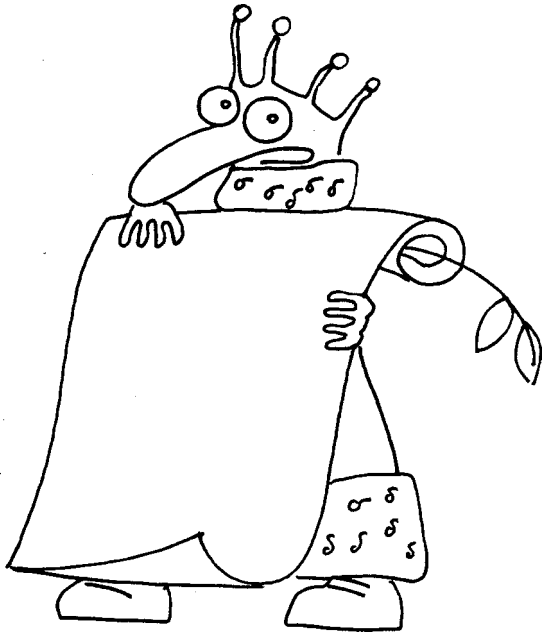
Оценка хеш-функции в приведенном примере равна четырем.

Задача 2. Разделяй и властвуй

имя входного файла: Divide.in
имя выходного файла: Divide.out
время на один ход: 20 секунд
максимальный балл: 33
автор: Д. Хмелев

В тридесятом государстве ради сбора мытных денег (таможенных пошлин) с торговых людей царь Салтан указал сделать засеки – заградительные полосы, которые тянутся с севера на юг или с запада на восток и преграждают путь всякому пешему и конному.

Тридесятое государство имеет форму прямоугольника, стороны которого тянутся с запада на восток и с севера на юг. Картограф изобразил карту государства в виде прямоугольника на системе координат с центром O , совпадающим с юго-западным углом страны и осями, параллельными осям координат. Засеки на карте изображены в виде отрезков различной длины, параллельных осям координат. При этом оказалось, что никакие два отрезка не пересекаются и не лежат на одной прямой.



Для удобства управления своей страной царь Салтан повелел разделить ее на непересекающиеся прямоугольные области с соблюдением следующих условий:

- каждая засека входит полностью в границы областей
- число областей должно быть наименьшим, дабы уменьшить расходы на приказный люд.

Напишите программу, помогающую картографическому приказу построить искомое разбиение.

Входные данные

В первой строке входного файла находятся координаты северо-восточного угла страны, Во второй строке указано количество засек N ($N=1000$). Каждая из следующих N строк содержит описание

одной из засек в виде четверки чисел x_1, y_1, x_2, y_2 , что соответствует отрезку с концевыми точками (x_1, y_1) и (x_2, y_2) . Все координаты являются неотрицательными целыми и не превосходят 10000.

Выходные данные

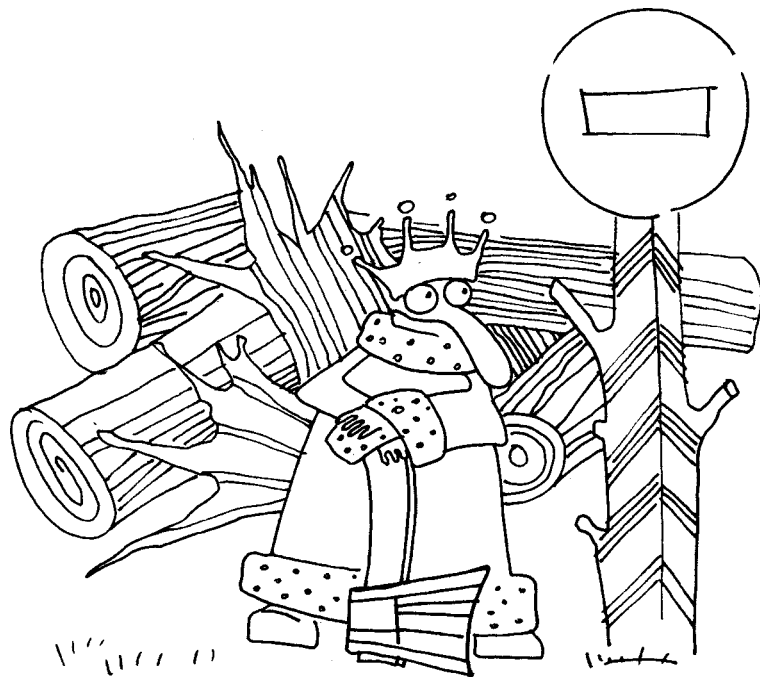
В первую строку выходного файла выведите количество областей M в найденном разбиении. В последующих M строках должны находиться целочисленные координаты юго-западного и северо-восточного углов для каждой из областей разбиения. Числа внутри строки разделять пробелами.

Пример входного файла divide.in

```
5 5
3
4 1 2 1
2 2 2 3
2 4 0 4
```

Пример выходного файла divide.out

```
4
2 0 5 1
2 1 5 5
0 4 2 5
0 0 2 4
```



Задача 3. Приватизация

имя входного файла rail.in
имя выходного файла: rail.out
время тестирования 20 секунд
максимальный балл: 33
автор: С.Волченков

Железнодорожная сеть состоит из отдельных участков, соединяющих города попарно. Этими участками каждый из N имеющихся городов соединен с тремя другими городами. Две компании собираются приватизировать все эти участки, но антимонопольный комитет настаивает на том, чтобы из каждого города выходили дороги разных компаний.

Напишите программу, которая распределяет все участки между компаниями с соблюдением вышеназванного требования или сообщает, что это сделать невозможно.

Входные данные

В первой строке входного файла содержится натуральное число N — количество городов ($N=5000$). Далее для каждого города i ($i=1, \dots, N$) записана тройка различных чисел — номера тех городов, с которыми он связан. Соединение городов i и j , если оно существует, отражено в соответствующих строках. Между любыми двумя городами может проходить не более одной дороги.

Выходные данные

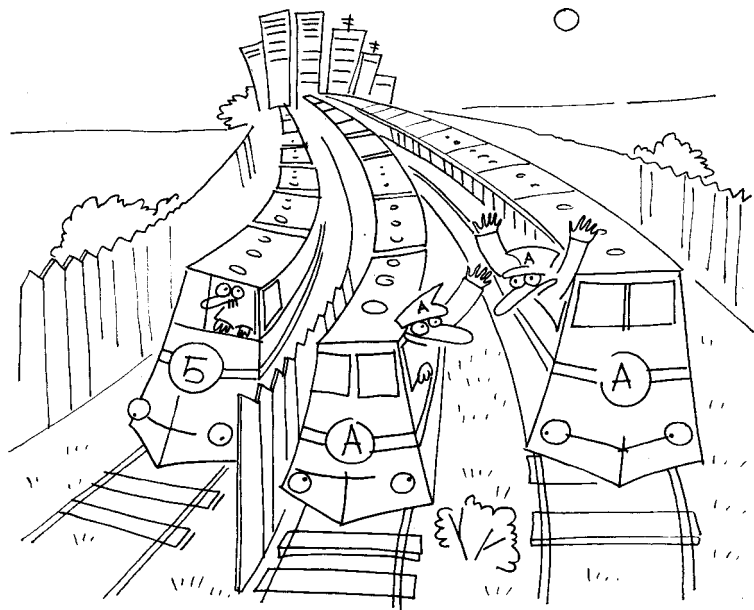
В выходной файл требуется вывести список участков, предназначенных для приватизации одной из компаний (все остальные участки приватизируются другой компанией). Каждый участок задается в отдельной строке двумя числами — номерами тех городов, которые он соединяет. Числа должны разделяться пробелами. Если решения не существует, выходной файл должен содержать единственную строку "NO SOLUTION".

Пример входного файла rail.in

```
6
6 5 3
6 4 3
4 1 2
5 3 2
6 4 1
2 1 5
```

Пример выходного файла rail.out

```
1 6
2 4
3 4
5 6
```



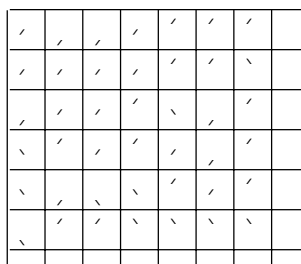
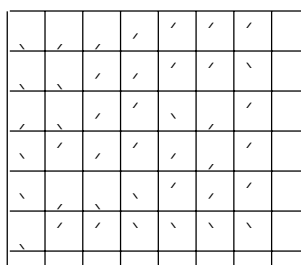
Задача 4. Игра “Просто Филя”

имя входного файла: `fiya.in`
имя выходного файла: `fiya.out`
время на один ход: 5 секунд
максимальный балл: 50

Дано прямоугольное клеточное поле 41×45 клеток. Каждая клетка покрашена в один из шести цветов. Цвета пронумерованы от 1 до 6. Левая верхняя и правая нижняя клетки поля имеют различный цвет. В результате поле разбивается на некоторое количество одноцветных областей. Две клетки одного и того же цвета, имеющие общую сторону, принадлежат одной и той же области.

Правила игры

Играют два игрока. За первым игроком закреплена область, включающая левую верхнюю клетку, за вторым – правую нижнюю. Игроки ходят по очереди. Делая ход, игрок перекрашивает свою область в любой из шести цветов по своему выбору, за исключением цвета своей области и цвета области противника. В результате хода к области игрока присоединяются все прилегающие к ней области выбранного цвета, если такие имеются (на рисунке изображена раскраска левого верхнего угла поля до хода первого игрока и после него).



Цель игры – включить в свою область как можно больше клеток. Игра заканчивается, когда все поле разобьется на две области или в течение 4 ходов ни одна область не увеличилась.

Задание

Написать программу, которая по текущему состоянию поля делает один ход за первого игрока. Программа будет участвовать в тестовой игре следующим образом:

по данным входного файла ваша программа выбирает номер цвета, записывает его в выходной файл и заканчивает свою работу;

тестирующая система по выбранному цвету перекрашивает область первого игрока, делает ответный ход и вновь запускает вашу программу, предоставляя ей в качестве начальных данных измененное поле;

ваша программа будет запускаться до тех пор, пока тестовая игра не завершится; в случае некорректного хода вашей программы тестовая игра завершается досрочно.

Критерии оценки

Качество вашей программы будет оцениваться в зависимости от количества клеток в вашей области после завершения тестовой игры.

Входные данные

Входной файл `filya.in` содержит 41 строку по 45 цифр в каждой без пробелов. Первая цифра файла соответствует цвету левой верхней клетки игрового поля. Вам предоставляется пример входного файла.

Выходные данные

Выходной файл `filya.out` содержит номер цвета хода.

Пример выходного файла

5

Примечание

Для отладки своей программы вы можете воспользоваться программой `filya.exe`, правила работы с которой описаны в файле `read.me`.

Задача 5. Поезда

имя входного файла

train.in

имя выходного файла:

train.out

время тестирования

20 секунд

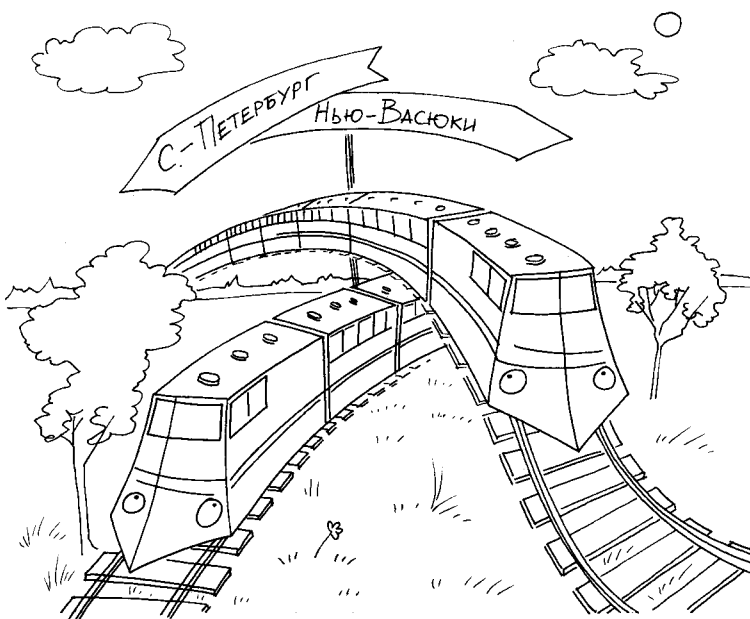
максимальный балл:

33

автор:

В.Матюхин

В связи с увеличившимся числом аварий на железнодорожной трассе “Нью-Васюки–Петербург” руководство железнодорожной компании решило изменить график движения поездов. Тщательный анализ состояния полотна установил, что оптимальным является следующий график движения: сначала T_1 минут поезд идет со скоростью V_1 метров в минуту, затем T_2 минут со скоростью V_2 м/мин, ..., наконец, T_N минут со скоростью V_N м/мин.



В течение интервала T_i ($1 \leq i \leq N$) поезд может стоять.

По действующей инструкции обеспечения безопасности движения расстояние между началами следующих друг за другом поездов никогда не должно быть менее L метров. Напишите программу, определяющую минимально допустимый интервал в минутах между отправлениями поездов, позволяющий им двигаться по установленному графику без опасного сближения.

Входные данные

В первых двух строках входного файла содержатся натуральные числа, задающие минимально допустимое расстояние L и количество участков пути N ($100 \leq L \leq 10000$, $1 \leq N \leq 1000$). Далее следуют N пар целых чисел $T_1, V_1, \dots, T_N, V_N$, описывающих график движения ($1 \leq T_i \leq 1000$, $0 \leq V_i \leq 1000$).

Выходные данные

В выходной файл требуется вывести искомый интервал между отправлениями поездов в минутах, не менее чем с тремя верными знаками после десятичной точки.

Пример входного файла

```
1000
4
10 0
30 80
15 0
20 100
```

Пример выходного файла

```
27.5
```

Задача 6. Бал

имя входного файла	bal.in
имя выходного файла:	bal.out
время тестирования	20 секунд
максимальный балл:	17
авторы:	В. и Т. Овсянниковы

*Увы, на разные забавы
Я много жизни погубил!
Но если б не страдали нравы,
Я балы б до сих пор любил.
А.С.Пушкин*

Во время бала в зале, имеющем форму M -угольника $A_1A_2\dots A_M$, этикетом предписано размещаться N придворным дамам вдоль стен и в углах так, чтобы у всех стен стояло равное число дам. Если дама находится в углу зала, то она считается стоящей у обеих стен этого угла. Вдоль стены может размещаться любое количество дам, а в углу не больше одной.

Напишите программу, находящую требуемое расположение дам.

Входные данные

Во входном файле содержатся два натуральных числа M и N ($3 \leq M \leq 1000$, $1 \leq N \leq 10^9$).

Выходные данные

В выходной файл для каждого угла требуется вывести число дам, стоящих в этом углу (0 или 1), а для каждой стены — количество дам, стоящих вдоль нее (не считая тех, что стоят в углах). Таким образом, в файле должно быть $2M$ чисел в соответствии со следующим порядком: сторона $A_M A_1$, угол A_1 , сторона $A_1 A_2$, угол A_2 , ..., сторона $A_{M-1} A_M$, угол A_M .

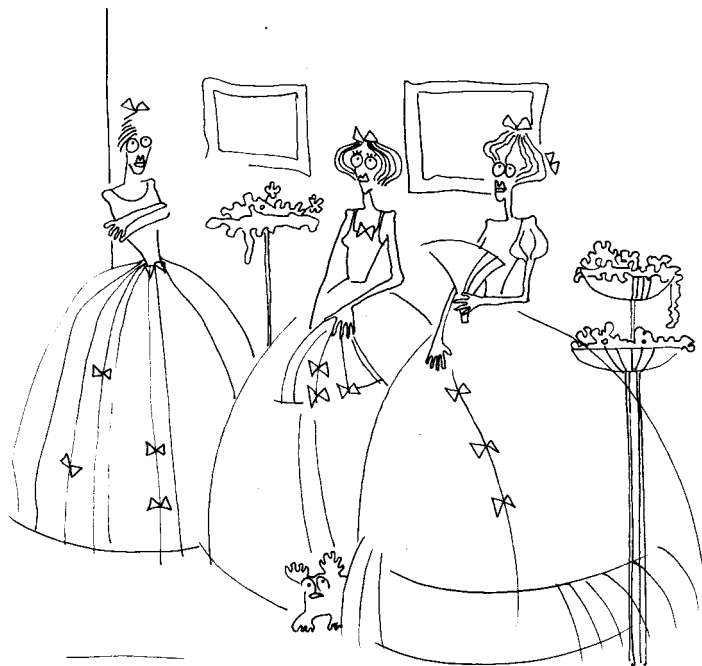
Числа в файле разделяются пробелами и/или переводами строки. В случае, если требуемым образом дамам разместиться невозможно, выходной файл должен содержать единственную строку NO.

Пример входного файла

3 10

Пример выходного файла

3 1
2 1
3 0



*Миронов Илья Лазаревич,
председатель научного комитета
X Всероссийской олимпиады
школьников по информатике,
студент математико-механического
факультета СПбГУ.*

НАШИ АВТОРЫ