Фестиваль «Юные интеллектуалы Среднего Урала»
Муниципальный этап Всероссийской олимпиады по информатике
2010 – 2011 учебный год
8 класс

Время выполнения задач — 4 часа
Ограничение по времени — 2 секунды на тест
Ограничение по памяти — 64 мегабайта

Уважаемый участник!

В настоящее время решения на олимпиадах по информатике проверяются автоматически. Ваша задача — написать программу, которая по заданным входным данным вычисляет и выводит выходные данные. Когда вы сдаёте решение на проверку, проверяющая программа «подсовывает» вашему решению тестовые наборы входных данных, запускает вашу программу и затем анализирует выданный ею результат: если в задаче возможно несколько правильных ответов, то программа может выводить любой из них, если в условии не указано иное.

Ваше решение должно читать входные данные из файла **input.txt** в описанном формате, решать задачу, и выводить результат в файл **output.txt**. Всякая строка в файле **input.txt** завершается переводом строки. Программа должна всегда завершаться с кодом 0 (иначе тестирующая программа считает, что в ходе работы произошла ошибка) — то есть, командой **halt(0);** или просто достижением конца текста, если Вы пишете на Паскале, или **return 0;** для программ на C/C++.

Считывать информацию из текстового файла так же просто, как и с клавиатуры — в Бейсике, Паскале и C/C++ для ввода с клавиатуры и из файла используются или одни и те же операторы или же операторы со сходной структурой вызова. То же касается и вывода в файл. Возникают лишь небольшие отличия, связанные с файловыми переменными (см. памятку по работе с файлами).

Ваша программа не должна ничего выводить на экран, а также ждать какого-либо ввода пользователя. Распространённой ошибкой является ситуация, когда после окончания работы программа ждёт нажатия какой-либо клавиши. Этого быть не должно.

Те, кто программирует на Паскале в среде Borland Pascal, не должны использовать модуль CRT: программа не должна содержать команды **uses crt;**

Программа должна выдавать в выходной файл ту и только ту информацию, которая описана в формате вывода. Более того, вывод программы должен в точности удовлетворять формату, описанному в условии задачи (заглавные/строчные буквы, наличие/отсутствие пробелов, переводы строк). В противном случае, программа считается нерабочей и оценивается в 0 баллов.

Все задачи считаются равноценными и имеют одинаковую максимальную оценку в 100 баллов. Все тесты в рамках каждой задачи также равноценны.

8.1. «Сажаем картошку». Весной Петя Торопыжкин помогал бабушке в деревне сажать картошку. Картофельное поле состоит из n рядов, в каждом из которых выкопано m лунок. В одну лунку нужно посадить одну картофелину. В один мешок входит k картофелин. Сколько мешков понадобится для засева бабушкиного поля? (Мешки можно покупать только целиком!)

Формат ввода: В единственной строке записаны целые числа n, m и k, разделённые пробелами ($1 \leqslant n, m \leqslant 150$; $1 \leqslant k \leqslant 10\,000$).

Формат вывода: Выведите единственное целое число — количество мешков картошки, которое надо купить.

Пример

input.txt:	output.txt:
1 1 2	1

8.2. «Самый большой монитор». На день рождения начинающему программисту Пете Торопыжкину родители решили подарить новый монитор. Они знают, что с точки зрения Пети монитор тем лучше, чем больше на нем пикселов. Напишите программу, которая из заданного набора мониторов (для каждого монитора известны его ширина и высота в пикселах) выберет наилучший.

Формат ввода: В первой строке записано целое число n — количество типов мониторов ($1 \leqslant n \leqslant 150$). В k-й из следующих n строк записаны два целых числа w_k и h_k — ширина и высота монитора k-го типа в пикселах, разделённые пробелом ($1 \leqslant w_k, h_k \leqslant 150$).

Формат вывода: Выведите единственное целое число — номер наилучшего монитора (мониторы занумерованы в порядке, в котором они перечислены в исходном списке). Если есть несколько мониторов с наибольшим количеством пикселов, выведите номер первого из них.

Пример

input.txt:	output.txt:
3 1 1 5 10 12 2	2

8.3. «На уроке английского». В тетради по английскому языку Петя Торопыжкин написал длинное слово, состоящее только из заглавных букв латинского алфавита. Ему стало интересно, сколько различных букв латинского алфавита содержит это слово. Помогите ему — напишите программу, которая по введённой строке будет выдавать эту информацию.

Формат ввода: Единственная строка содержит слово, которое написал Петя. Это слово состоит только из заглавных латинских букв, непусто и имеет длину не более 200 символов. Слово завершается переводом строки.

Формат вывода: Выведите единственное целое число — количество различных букв в слове, которое написал Петя.

Пример

input.txt:	output.txt:
ABCDA	4

8.4. «Сложение дробей». Успешно справившись с домашним заданием по английскому языку и сев за компьютер с новым монитором, Петя Торопыжкин занялся на компьютере математикой. Однако результат первого же действия сильно удивил его: он разделил 10 на 3 и вместо ожидаемого $3\,1/3$ или хотя бы $10/3$ увидел на экране 3.33333333334. Чтобы исправить этот «недостаток» компьютера, Петя решил научить компьютер работать с обыкновенными дробями, а для начала — складывать их. Помогите Пете — напишите программу, которая складывает две положительные обыкновенные дроби и выдаёт результат в виде обыкновенной несократимой дроби (возможно, неправильной). Дроби задаются числителем и знаменателем.

Формат ввода: В первой строке записаны целые числа p_1 и q_1 — числитель и знаменатель первой дроби, во второй — целые числа p_2 и q_2 — числитель и знаменатель второй дроби; $1 \leqslant p_1, q_1, p_2, q_2 \leqslant 10\,000$. Числа разделены пробелами.

Формат вывода: Выведите два целых числа, разделённые пробелом: числитель и знаменатель несократимой дроби, являющейся суммой заданных дробей. Если в результате получилось целое число, следует выдать знаменатель 1.

Пример 1

input.txt:	output.txt:
5 3 1 2	13 6

Пример 2

input.txt:	output.txt:
3 4 2 1	11 4

Пример 3

input.txt:	output.txt:
5 4 3 4	2 1

8.5. «Сортируем числа». Дана последовательность, состоящая из чисел 1, 2 и 3. Длина последовательности не превосходит 200. Можно брать любые два элемента и менять их местами. Требуется за наименьшее число таких перестановок поставить на первые места в последовательности все единички, дальше все двойки и потом все тройки.

Формат ввода: В первой строке записано целое число n ($1 \leqslant n \leqslant 200$). В следующей строке перечислены n чисел, каждое из которых равно 1, 2 или 3. Числа разделены пробелами.

Формат вывода: Выведите неотрицательное целое число — наименьшее количество перестановок, за которое можно отсортировать последовательность.

Пример

input.txt:	output.txt:
5 2 3 3 1 1	3

Фестиваль «Юные интеллектуалы Среднего Урала»
Муниципальный этап Всероссийской олимпиады по информатике
2010 – 2011 учебный год
9 класс

Время выполнения задач — 4 часа
Ограничение по времени — 2 секунды на тест
Ограничение по памяти — 64 мегабайта

Уважаемый участник!

В настоящее время решения на олимпиадах по информатике проверяются автоматически. Ваша задача — написать программу, которая по заданным входным данным вычисляет и выводит выходные данные. Когда вы сдаёте решение на проверку, проверяющая программа «подсовывает» вашему решению тестовые наборы входных данных, запускает вашу программу и затем анализирует выданный ею результат: если в задаче возможно несколько правильных ответов, то программа может выводить любой из них, если в условии не указано иное.

Ваше решение должно читать входные данные из файла **input.txt** в описанном формате, решать задачу, и выводить результат в файл **output.txt**. Всякая строка в файле **input.txt** завершается переводом строки. Программа должна всегда завершаться с кодом 0 (иначе тестирующая программа считает, что в ходе работы произошла ошибка) — то есть, командой **halt(0);** или просто достижением конца текста, если Вы пишете на Паскале, или **return 0;** для программ на C/C++.

Считывать информацию из текстового файла так же просто, как и с клавиатуры — в Бейсике, Паскале и C/C++ для ввода с клавиатуры и из файла используются или одни и те же операторы или же операторы со сходной структурой вызова. То же касается и вывода в файл. Возникают лишь небольшие отличия, связанные с файловыми переменными (см. памятку по работе с файлами).

Ваша программа не должна ничего выводить на экран, а также ждать какого-либо ввода пользователя. Распространённой ошибкой является ситуация, когда после окончания работы программа ждёт нажатия какой-либо клавиши. Этого быть не должно.

Те, кто программирует на Паскале в среде Borland Pascal, не должны использовать модуль CRT: программа не должна содержать команды **uses crt;**

Программа должна выдавать в выходной файл ту и только ту информацию, которая описана в формате вывода. Более того, вывод программы должен в точности удовлетворять формату, описанному в условии задачи (заглавные/строчные буквы, наличие/отсутствие пробелов, переводы строк). В противном случае, программа считается нерабочей и оценивается в 0 баллов.

Все задачи считаются равноценными и имеют одинаковую максимальную оценку в 100 баллов. Все тесты в рамках каждой задачи также равноценны.

9.1. «Кратность делителя». На парте в кабинете математики Петя Торопыжкин обнаружил натуральные числа n и m, причём m большие единицы. Его заинтересовало, является ли число m делителем числа n и, если да, то какой кратности. Помогите ему — напишите программу, отвечающую на этот вопрос.

Формат ввода: В единственной строке записаны целые числа n и m, разделённые пробелами ($1 \leqslant n \leqslant 10^9$; $2 \leqslant m \leqslant 10^9$).

Формат вывода: Выведите единственное неотрицательное целое число — кратность делителя m у числа n. Если m не является делителем n, следует вывести 0.

Пример 1

input.txt:	output.txt:
120 2	3

Пример 2

input.txt:	output.txt:
1034 3	0

9.2. «Кольцо из проводов». У Пети Торопыжкина есть набор проводов, концы которых обжаты в разъёмы двух типов. Существует три типа проводов: 1–1 — их у Пети n штук, 2–2 — их m штук и 1–2 — их k штук (указаны типы разъёмов на концах). При этом два разъёма можно соединять друг с другом только если они одного типа. Напишите программу, которая сможет по заданным n, m и k определить, какое минимальное количество проводов нужно исключить из набора, чтобы из оставшихся можно было собрать кольцо.

Формат ввода: В единственной строке записаны целые числа n, m и k — количество 1–1, 2–2 и 1–2 проводов, соответственно ($0 \leqslant n, m, k \leqslant 10^9$). Числа разделены пробелами.

Формат вывода: Выведите единственное целое число — минимальное количество проводов из набора, которые нужно выкинуть, чтобы из оставшихся можно было составить кольцо. Если можно собрать кольцо, ничего не выкидывая, выведите `0`. Если никакое кольцо составить в принципе невозможно, выведите `-1`.

Пример 1

input.txt:	output.txt:
10 5 2	0

Пример 2

input.txt:	output.txt:
7 0 5	1

9.3. «Сколько приложений?». На смартфоне Пети Торопыжкина установлено много полезных приложений, которыми он часто пользуется. Однако Петя сам никогда не закрывает приложения. Если необходимость в запущенном приложении отпадает, он отправляет его в фоновый режим, не выгружая из памяти. Все приложения устроены таким образом, что позволяют запустить одновременно лишь одну свою копию, так что если Петя запустит приложение, которое уже было запущено ранее, то оно просто выйдет из фонового режима и второго запущенного его экземпляра не возникнет.

Напишите программу, которая по списку названий запущенных Петей приложений (в хронологическом порядке) определит, сколько сейчас приложений в памяти (в активном или фоновом режиме).

Формат ввода: В первой строке записано целое число n — сколько раз Петя запускал приложения ($1 \leqslant n \leqslant 5\,000$). В следующих n строках перечислены названия приложений в том порядке, в котором их запускал Петя. Название приложения — строка, состоящая из заглавных и строчных латинских букв, длиной не более 20 символов. Приложения, названия которых отличаются только регистром букв, следует считать различными.

Формат вывода: Выведите целое число — количество приложений, находящихся в памяти смартфона после всех манипуляций Пети.

Пример 1

input.txt:	output.txt:
1 NotePad	1

Пример 2

input.txt:	output.txt:
7 NotePad Word NotePad CorelDraw Word AdobePhotoshop Word	4

9.4. «Питание насекомых». У Пети Торопыжкина живёт ручной жучок Ипполит. Уходя в школу, Петя оставляет Ипполиту на письменном столе квадратную плитку шоколада размера 2×2. Центр плитки совпадает с началом декартовой системы координат, которую Петя ввёл на столе, а стороны плитки параллельны осям. Однако Ипполит нетерпелив: он падает с потолка куда придётся. Если он не попадает на плитку, то бежит к ближайшей её точке, где и принимается за трапезу. Напишите программу, которая, зная точку падения Ипполита, находит ближайшую к нему точку шоколадной плитки.

Формат ввода: В единственной строке записаны два вещественных числа, по модулю не превышающие 100, с не более чем двумя знаками после десятичной точки — координаты точки, в которую упал Ипполит. Числа разделены пробелом.

Формат вывода: Выведите два вещественных числа, разделённые пробелом — округлённые до сотых координаты точки, в которую следует бежать Ипполиту.

Пример 1

input.txt:	output.txt:
1.10 1.2	1.00 1.00

Пример 2

input.txt:	output.txt:
0.9 1.1	0.90 1.00

9.5. «Математический бильярд». На координатной плоскости в первом квадранте расположен прямоугольный бильярдный стол так, что его стороны парал-

лельны координатным осям, а один из углов находится в начале координат. Размеры стола $a \times b$ (a — вдоль оси Ox, b — вдоль оси Oy). Точечный бильярдный шар расположен в точке (x, y), находящейся на поверхности стола, и движется со скоростью $v = (v_x, v_y)$. Шар движется без трения (сохраняя величину скорости) и отражается от бортов по закону «угол падения равен углу отражения». Если шар попадает точно в угол, считается, что он мгновенно испытывает два отражения и, стало быть, изменяет вектор скорости на противоположный. Все линейные величины заданы в метрах, скорости — в метрах в секунду. Вопрос: в какой точке шар окажется через t секунд?

Формат ввода: В первой строке записаны числа a и b, во второй — числа x и y, в третьей — числа v_x и v_y. Числа разделены пробелом. В четвёртой строке записано единственное число t. Все числа целые, неотрицательные и не превосходят 10 000. a и b строго положительные.

Формат вывода: Выведите два неотрицательных целых числа, разделённые пробелами — координаты шара через t секунд.

Пример

`input.txt`:	`output.txt`:

`10 10`
`1 1`
`1 2`
`7`
	`8 5`

Фестиваль «Юные интеллектуалы Среднего Урала»
Муниципальный этап Всероссийской олимпиады по информатике
2010 – 2011 учебный год
10 класс

Время выполнения задач — 4 часа
Ограничение по времени — 2 секунды на тест
Ограничение по памяти — 64 мегабайта

Уважаемый участник!

В настоящее время решения на олимпиадах по информатике проверяются автоматически. Ваша задача — написать программу, которая по заданным входным данным вычисляет и выводит выходные данные. Когда вы сдаёте решение на проверку, проверяющая программа «подсовывает» вашему решению тестовые наборы входных данных, запускает вашу программу и затем анализирует выданный ею результат: если в задаче возможно несколько правильных ответов, то программа может выводить любой из них, если в условии не указано иное.

Ваше решение должно читать входные данные из файла **input.txt** в описанном формате, решать задачу, и выводить результат в файл **output.txt**. Всякая строка в файле **input.txt** завершается переводом строки. Программа должна всегда завершаться с кодом 0 (иначе тестирующая программа считает, что в ходе работы произошла ошибка) — то есть, командой **halt(0);** или просто достижением конца текста, если Вы пишете на Паскале, или **return 0;** для программ на C/C++.

Считывать информацию из текстового файла так же просто, как и с клавиатуры — в Бейсике, Паскале и C/C++ для ввода с клавиатуры и из файла используются или одни и те же операторы или же операторы со сходной структурой вызова. То же касается и вывода в файл. Возникают лишь небольшие отличия, связанные с файловыми переменными (см. памятку по работе с файлами).

Ваша программа не должна ничего выводить на экран, а также ждать какого-либо ввода пользователя. Распространённой ошибкой является ситуация, когда после окончания работы программа ждёт нажатия какой-либо клавиши. Этого быть не должно.

Те, кто программирует на Паскале в среде Borland Pascal, не должны использовать модуль CRT: программа не должна содержать команды **uses crt;**

Программа должна выдавать в выходной файл ту и только ту информацию, которая описана в формате вывода. Более того, вывод программы должен в точности удовлетворять формату, описанному в условии задачи (заглавные/строчные буквы, наличие/отсутствие пробелов, переводы строк). В противном случае, программа считается нерабочей и оценивается в 0 баллов.

Все задачи считаются равноценными и имеют одинаковую максимальную оценку в 100 баллов. Все тесты в рамках каждой задачи также равноценны.

10.1. «Игра Баше». Петя Торопыжкин играет со своим братом Сашей в игру Баше. В начале игры выкладывается кучка из n камней. На каждом ходу можно взять из неё один, два или три камня. Выигрывает тот, после чьего хода не останется камней. Первым ходит Петя, как старший. Конечно, в этой игре есть выигрышная стратегия, но ни Петя, ни Саша её не знают, поэтому ходят наобум. Запись партии представляет собой последовательность чисел a_1, a_2, ..., a_k — количество камней, взятых на очередном ходу (все числа a_i равны 1, 2 или 3). Проверьте, корректна ли запись партии (все ходы допустимы и последний камень будет взят ровно на k-м ходу), и, если она корректна, выведите имя победителя.

Формат ввода: В первой строке записаны целые числа n и k ($1 \leqslant n, k \leqslant 10\,000$). В i-й из следующих k строк записано число a_i, равное 1, 2 или 3 — запись i-го хода.

Формат вывода: Если запись партии некорректна (в конце остались камни или были попытки взять камней больше, чем их осталось на тот момент), программа должна выдать слово `INCORRECT`. Если запись корректна и выиграл Петя, программа должна выдать `PETYA`, если Саша — `SASHA`.

Пример 1

input.txt:	output.txt:
5 3 1 1 1	INCORRECT

Пример 2

input.txt:	output.txt:
5 3 1 2 2	PETYA

Пример 3

input.txt:	output.txt:
5 4 2 1 1 1	SASHA

10.2. «Хороший монитор». После того, как родители подарили Пете Торопыжкину новый монитор, его младший брат Саша тоже захотел себе новый монитор. Саша подходит к выбору монитора так: он считает, что монитор *хороший*, если его размеры (выраженные в пикселах) взаимно просты (то есть, не имеют общих делителей, кроме единицы). Известно, что на складе лучшей в городе компьютерной фирмы имеются мониторы с любой шириной w из диапазона $w_{\min} \leqslant w \leqslant w_{\max}$ и любой высотой h из диапазона $h_{\min} \leqslant h \leqslant h_{\max}$. Сколько видов *хороших* мониторов имеется на складе компьютерной фирмы?

Формат ввода: В первой строке записаны целые числа $w_{\min}$ и $w_{\max}$, во второй строке — целые числа $h_{\min}$ и $h_{\max}$ ($2 \leqslant w_{\min} \leqslant w_{\max} \leqslant 1\,000$; $2 \leqslant h_{\min} \leqslant h_{\max} \leqslant 1\,000$). Числа разделены пробелом.

Формат вывода: Выведите единственное неотрицательное целое число — количество видов *хороших* мониторов с размерами, лежащими в заданном диапазоне.

Пример

input.txt:	output.txt:
9 10	2
8 9	

10.3. «Купаться!». Летом Петя Торопыжкин отдыхал в деревне у бабушки. В окрестности их домика расположено идеально круглое озеро единичного радиуса с центром в начале деревенской декартовой системы координат. Однажды днём Петю разморило, и он уснул в точке с координатами (x, y). Конечно, Петя спал не в воде. Проснувшись, он побежал к ближайшей точке пруда. Каковы координаты этой точки?

Формат ввода: В единственной строке записаны вещественные числа x и y, по модулю не превышающие 100, с не более чем двумя знаками после десятичной точки — координаты точки, где спал Петя. Числа разделены пробелом.

Формат вывода: Выведите два вещественных числа, разделённые пробелом — округлённые до сотых координаты точки, в которую следует бежать Пете.

Пример

input.txt:	output.txt:
10 10	0.71 0.71

10.4. «Падение метеорита». Петя Торопыжкин с интересом посмотрел телепередачу о том, что Луна попала в метеоритный поток и подверглась метеоритной бомбардировке. Учёные оказались не готовы к этому событию и фиксировали время падения каждого из n метеоритов на отдельном листочке в формате `YYYY-MM-DD hh:mm:ss`, где `ss` — секунды, `mm` — минуты, `hh` — часы, `DD` — день, `MM` — месяц, `YYYY` — год. Никакие два метеорита не упали одновременно. Бумажки смешались, но необходимо срочно определить k-й в хронологическом порядке метеорит, упавший на Луну: именно он является наиболее интересным для изучения.

Формат ввода: В первой строке записаны целые числа n и k, разделённые пробелом — количество метеоритов и номер метеорита, интересующего учёных ($1 \leqslant k \leqslant n \leqslant 10^5$). В i-й из следующих n строк в указанном формате записано время падения i-го метеорита (метеориты перечислены в произвольном порядке). Диапазоны величин: $1600 \leqslant$ `YYYY` $\leqslant 2500$; $1 \leqslant$ `MM` $\leqslant 12$; `DD` $\geqslant 1$ и не больше числа дней в соответствующем месяце (високосность годов не учитывается); $0 \leqslant$ `hh` $\leqslant 23$; $0 \leqslant$ `mm`, `ss` $\leqslant 59$.

Формат вывода: Выведите время падения k-го в хронологическом порядке метеорита в том же формате, в котором оно описано во входных данных.

Пример

input.txt:	output.txt:
3 2 2010-01-31 23:00:21 2009-08-12 01:01:59 2009-10-05 00:59:00	2009-10-05 00:59:00

10.5. «Интересное число». На парте в кабинете математики Петя Торопыжкин увидел длинную строку, состоящую из цифр и вопросительных знаков. Ему стало интересно, можно ли каждый вопросительный знак заменить на цифру так, чтобы получилось число, кратное одиннадцати. В полученном числе не может быть ведущих нулей. Если таких чисел несколько, Петя хотел бы получить наименьшее возможное.

Формат ввода: Единственная строка содержит только цифры и вопросительные знаки и оканчивается переводом строки. Первый символ этой строки отличен от цифры 0. Длина строки — не менее двух и не более ста символов.

Формат вывода: Выведите наименьшее целое число без ведущих нулей, кратное одиннадцати и получаемое из исходной строки заменой вопросительных знаков на цифры. Если такого числа не существует, выведите IMPOSSIBLE.

Пример 1

input.txt:	output.txt:
1??2	1012

Пример 2

input.txt:	output.txt:
?0903	IMPOSSIBLE

Фестиваль «Юные интеллектуалы Среднего Урала»
Муниципальный этап Всероссийской олимпиады по информатике
2010 – 2011 учебный год
11 класс

Время выполнения задач — 4 часа
Ограничение по времени — 2 секунды на тест
Ограничение по памяти — 64 мегабайта

Уважаемый участник!

В настоящее время решения на олимпиадах по информатике проверяются автоматически. Ваша задача — написать программу, которая по заданным входным данным вычисляет и выводит выходные данные. Когда вы сдаёте решение на проверку, проверяющая программа «подсовывает» вашему решению тестовые наборы входных данных, запускает вашу программу и затем анализирует выданный ею результат: если в задаче возможно несколько правильных ответов, то программа может выводить любой из них, если в условии не указано иное.

Ваше решение должно читать входные данные из файла **input.txt** в описанном формате, решать задачу, и выводить результат в файл **output.txt**. Всякая строка в файле **input.txt** завершается переводом строки. Программа должна всегда завершаться с кодом 0 (иначе тестирующая программа считает, что в ходе работы произошла ошибка) — то есть, командой **halt(0);** или просто достижением конца текста, если Вы пишете на Паскале, или **return 0;** для программ на C/C++.

Считывать информацию из текстового файла так же просто, как и с клавиатуры — в Бейсике, Паскале и C/C++ для ввода с клавиатуры и из файла используются или одни и те же операторы или же операторы со сходной структурой вызова. То же касается и вывода в файл. Возникают лишь небольшие отличия, связанные с файловыми переменными (см. памятку по работе с файлами).

Ваша программа не должна ничего выводить на экран, а также ждать какого-либо ввода пользователя. Распространённой ошибкой является ситуация, когда после окончания работы программа ждёт нажатия какой-либо клавиши. Этого быть не должно.

Те, кто программирует на Паскале в среде Borland Pascal, не должны использовать модуль CRT: программа не должна содержать команды **uses crt;**

Программа должна выдавать в выходной файл ту и только ту информацию, которая описана в формате вывода. Более того, вывод программы должен в точности удовлетворять формату, описанному в условии задачи (заглавные/строчные буквы, наличие/отсутствие пробелов, переводы строк). В противном случае, программа считается нерабочей и оценивается в 0 баллов.

Все задачи считаются равноценными и имеют одинаковую максимальную оценку в 100 баллов. Все тесты в рамках каждой задачи также равноценны.

11.1. «Двойной факториал». Помогите Пете Торопыжкину написать программу, которая по введённому целому числу n вычислит $n!!$. Напомним, что

$$n!! = \begin{cases} 1 \cdot 3 \cdot \ldots \cdot n, & \text{если } n \text{ нечётно,} \\ 2 \cdot 4 \cdot \ldots \cdot n, & \text{если } n \text{ чётно.} \end{cases}$$

Формат ввода: В единственной строке записано целое число n ($1 \leqslant n \leqslant 20$).

Формат вывода: Выведите единственное целое число — результат вычисления.

Пример 1

input.txt:	output.txt:
3	3

Пример 2

input.txt:	output.txt:
4	8

11.2. «На уроке английского-2». В тетрадке по английскому языку Петя Торопыжкин написал длинное слово, состоящее из заглавных букв латинского алфавита. Его заинтересовали *подслова* этого слова (то есть, подстроки данной строки). Двухбуквенные подслова он изучил быстро, а с трёхбуквенными вышла заминка. Помогите Пете: напишите программу, которая по заданному слову выдаёт его подслово из трёх букв, максимальное в лексикографическом порядке. Напомним, что *лексикографическим порядком* называется тот порядок, в котором упорядочены слова в словаре: сначала сравниваются первые буквы, в случае их равенства — вторые, а в случае равенства вторых — третьи.

Формат ввода: Единственная строка содержит слово, которое написал Петя. Это слово состоит только из заглавных латинских букв и имеет длину не менее 3 и не более 200 символов. Слово завершается переводом строки.

Формат вывода: Выведите в единственной строке требуемое трёхбуквенное подслово.

Пример

input.txt:	output.txt:
ABCA	BCA

11.3. «Ночной Дозор». Возможно, вам известна игра «Ночной Дозор». Вот один из вариантов правил. Выделяется n локаций на местности. Команда приезжает в первую локацию, где получает первую загадку. Отгадывая её, команда получает указание, в какую локацию ехать дальше, переезжает туда, снова отгадывает загадку, переезжает на новое место и так далее. Игра заканчивается, когда команда достигает одной из локаций, отмеченных организаторами как финальные.

Организаторы очередной игры составили план перемещения по локациям в виде списка чисел k_i. Число k_i указывает, что из i-й локации нужно следовать в локацию с номером k_i. Если i-я локация финальная, то $k_i = -1$.

Сейчас они хотят проверить, завершит ли за разумное время свои поездки команда, приехавшая в начале игры на первую локацию.

Формат ввода: В первой строке записано целое число n — количество локаций ($1 \leqslant n \leqslant 1\,000$). В i-й из следующих n строк записано число k_i ($1 \leqslant k_i \leqslant n$ или $k_i = -1$).

Формат вывода: Выведите единственное целое число — количество переездов, которое придётся совершить команде, приехавшей в начале игры на первую локацию, для того, чтобы достичь какой-либо финальной локации, либо `-1`, если план переездов составлен так, что команда, следуя ему, никогда не достигнет никакой финальной локации.

Пример 1

input.txt:	output.txt:
3	2
2	
3	
-1	

Пример 2

input.txt:	output.txt:
3	-1
2	
1	
-1	

11.4. «Магнитное число». Родители Пети Торопыжкина купили магниты на холодильник, имеющие форму цифр: a_0 нулей, a_1 единиц, ..., a_9 девяток. Когда мама сказала, что пора идти спать, Петя ответил, что хочет сначала составить из магнитов минимально возможное целое положительное число, кратное 15, так, чтобы все магниты были задействованы. Напишите программу, которая по набору магнитов у Пети найдёт это число. Ведущие нули в записи числа не допускаются.

Формат ввода: В единственной строке записаны целые числа a_0, a_1, ..., a_9 ($0 \leqslant a_i \leqslant 1\,000$; $a_0 + a_1 + \ldots + a_9 > 0$). Числа разделены пробелами.

Формат вывода: Выведите единственное целое положительное число без ведущих нулей — минимально возможное число, кратное 15, которое Петя может выложить из магнитов, используя их все. Если Петя не может выложить ни одного положительного числа, кратного 15, выведите `-1`.

Пример 1

input.txt:	output.txt:
1 1 0 0 0 1 0 0 0 0	105

Пример 2

input.txt:	output.txt:
1 0 0 0 0 1 0 0 0 0	-1

11.5. «Гуляя по столице». После того, как Петя Торопыжкин отдохнул в деревне у бабушки, он поехал с родителями в Москву. Благодаря стараниям мэра, столица обзавелась большим количеством кольцевых дорог. Эти дороги представляют собой окружности с центром в начале координат и радиусами, выражаемыми всевозможными целыми числами от 1 до 100 000.

Родители Пети наметили культурную программу — осмотр n достопримечательностей, расположенных в точках с координатами (x_i, y_i). Они хотят добраться к первой достопримечательности на такси и затем перемещаться от одной достопримечательности к другой пешком по прямой. Им интересно, пересекает ли очередной отрезок их пути хотя бы одну кольцевую дорогу (касание также считается пересечением). Помогите родителям Пети — напишите программу, которая ответит на этот вопрос.

Формат ввода: В первой строке записано целое число n — количество точек маршрута ($2 \leqslant n \leqslant 100$). В каждой из n следующих строк записаны вещественные числа x_i, y_i, по модулю не превышающие $10\,000$ и заданные с точностью не более двух знаков после десятичной точки — координаты очередной точки маршрута. Никакие две соседние точки маршрута не совпадают. Числа разделены пробелами.

Формат вывода: Выведите $n-1$ строку, в i-й из которых будет записано `RISKY`, если переход между i-й и $(i+1)$-й точками пересекает или касается хотя бы одной из кольцевых дорог, и `SAFE` в противном случае.

Пример

`input.txt`:	`output.txt`:
`4`	`RISKY`
`2.2 1.91`	`RISKY`
`1 1`	`SAFE`
`1 -1`	
`1 -1.1`	

Фестиваль
«Юные интеллектуалы Среднего Урала»

Муниципальный этап
Всероссийской олимпиады
по информатике

2010 – 2011 учебный год

Разборы решений и идеи тестов

Фестиваль «Юные интеллектуалы Среднего Урала»
Муниципальный этап Всероссийской олимпиады по информатике
2010 – 2011 учебный год
8 класс

Время выполнения задач — 4 часа
Ограничение по времени — 2 секунды на тест
Ограничение по памяти — 64 мегабайта

8.1. «Сажаем картошку». Весной Петя Торопыжкин помогал бабушке в деревне сажать картошку. Картофельное поле состоит из n рядов, в каждом из которых выкопано m лунок. В одну лунку нужно посадить одну картофелину. В один мешок входит k картофелин. Сколько мешков понадобится для засева бабушкиного поля? (Мешки можно покупать только целиком!)

Формат ввода: В единственной строке записаны целые числа n, m и k, разделённые пробелами ($1 \leqslant n, m \leqslant 150$; $1 \leqslant k \leqslant 10\,000$).

Формат вывода: Выведите единственное целое число — количество мешков картошки, которое надо купить.

Пример

input.txt:	output.txt:
1 1 2	1

8.2. «Самый большой монитор». На день рождения начинающему программисту Пете Торопыжкину родители решили подарить новый монитор. Они знают, что с точки зрения Пети монитор тем лучше, чем больше на нем пикселов. Напишите программу, которая из заданного набора мониторов (для каждого монитора известны его ширина и высота в пикселах) выберет наилучший.

Формат ввода: В первой строке записано целое число n — количество типов мониторов ($1 \leqslant n \leqslant 150$). В k-й из следующих n строк записаны два целых числа w_k и h_k — ширина и высота монитора k-го типа в пикселах, разделённые пробелом ($1 \leqslant w_k, h_k \leqslant 150$).

Формат вывода: Выведите единственное целое число — номер наилучшего монитора (мониторы занумерованы в порядке, в котором они перечислены в исходном списке). Если есть несколько мониторов с наибольшим количеством пикселов, выведите номер первого из них.

Пример

input.txt:	output.txt:
3 1 1 5 10 12 2	2

8.3. «На уроке английского». В тетради по английскому языку Петя Торопыжкин написал длинное слово, состоящее только из заглавных букв латинского алфавита. Ему стало интересно, сколько различных букв латинского алфавита содержит это слово. Помогите ему — напишите программу, которая по введённой строке будет выдавать эту информацию.

Формат ввода: Единственная строка содержит слово, которое написал Петя. Это слово состоит только из заглавных латинских букв, непусто и имеет длину не более 200 символов. Слово завершается переводом строки.

Формат вывода: Выведите единственное целое число — количество различных букв в слове, которое написал Петя.

Пример

input.txt:	output.txt:
ABCDA	4

8.4. «Сложение дробей». Успешно справившись с домашним заданием по английскому языку и сев за компьютер с новым монитором, Петя Торопыжкин занялся на компьютере математикой. Однако результат первого же действия сильно удивил его: он разделил 10 на 3 и вместо ожидаемого $3\,1/3$ или хотя бы $10/3$ увидел на экране 3.33333333334. Чтобы исправить этот «недостаток» компьютера, Петя решил научить компьютер работать с обыкновенными дробями, а для начала — складывать их. Помогите Пете — напишите программу, которая складывает две положительные обыкновенные дроби и выдаёт результат в виде обыкновенной несократимой дроби (возможно, неправильной). Дроби задаются числителем и знаменателем.

Формат ввода: В первой строке записаны целые числа p_1 и q_1 — числитель и знаменатель первой дроби, во второй — целые числа p_2 и q_2 — числитель и знаменатель второй дроби; $1 \leqslant p_1, q_1, p_2, q_2 \leqslant 10\,000$. Числа разделены пробелами.

Формат вывода: Выведите два целых числа, разделённые пробелом: числитель и знаменатель несократимой дроби, являющейся суммой заданных дробей. Если в результате получилось целое число, следует выдать знаменатель 1.

Пример 1

input.txt:	output.txt:
5 3 1 2	13 6

Пример 2

input.txt:	output.txt:
3 4 2 1	11 4

Пример 3

input.txt:	output.txt:
5 4 3 4	2 1

8.5. «Сортируем числа». Дана последовательность, состоящая из чисел 1, 2 и 3. Длина последовательности не превосходит 200. Можно брать любые два элемента и менять их местами. Требуется за наименьшее число таких перестановок поставить на первые места в последовательности все единички, дальше все двойки и потом все тройки.

Формат ввода: В первой строке записано целое число n ($1 \leqslant n \leqslant 200$). В следующей строке перечислены n чисел, каждое из которых равно 1, 2 или 3. Числа разделены пробелами.

Формат вывода: Выведите неотрицательное целое число — наименьшее количество перестановок, за которое можно отсортировать последовательность.

Пример

input.txt:	output.txt:
5	3
2 3 3 1 1	

Фестиваль «Юные интеллектуалы Среднего Урала»
Муниципальный этап Всероссийской олимпиады по информатике
2010 – 2011 учебный год
9 класс

Время выполнения задач — 4 часа
Ограничение по времени — 2 секунды на тест
Ограничение по памяти — 64 мегабайта

9.1. «Кратность делителя». На парте в кабинете математики Петя Торопыжкин обнаружил натуральные числа n и m, причём m большие единицы. Его заинтересовало, является ли число m делителем числа n и, если да, то какой кратности. Помогите ему — напишите программу, отвечающую на этот вопрос.

Формат ввода: В единственной строке записаны целые числа n и m, разделённые пробелами ($1 \leqslant n \leqslant 10^9$; $2 \leqslant m \leqslant 10^9$).

Формат вывода: Выведите единственное неотрицательное целое число — кратность делителя m у числа n. Если m не является делителем n, следует вывести 0.

Пример 1

input.txt:	output.txt:
120 2	3

Пример 2

input.txt:	output.txt:
1034 3	0

9.2. «Кольцо из проводов». У Пети Торопыжкина есть набор проводов, концы которых обжаты в разъёмы двух типов. Существует три типа проводов: 1–1 — их у Пети n штук, 2–2 — их m штук и 1–2 — их k штук (указаны типы разъёмов на концах). При этом два разъёма можно соединять друг с другом только если они одного типа. Напишите программу, которая сможет по заданным n, m и k определить, какое минимальное количество проводов нужно исключить из набора, чтобы из оставшихся можно было собрать кольцо.

Формат ввода: В единственной строке записаны целые числа n, m и k — количество 1–1, 2–2 и 1–2 проводов, соответственно ($0 \leqslant n, m, k \leqslant 10^9$). Числа разделены пробелами.

Формат вывода: Выведите единственное целое число — минимальное количество проводов из набора, которые нужно выкинуть, чтобы из оставшихся можно было составить кольцо. Если можно собрать кольцо, ничего не выкидывая, выведите `0`. Если никакое кольцо составить в принципе невозможно, выведите `-1`.

Пример 1

input.txt:	output.txt:
10 5 2	0

Пример 2

input.txt:	output.txt:
7 0 5	1

9.3. «Сколько приложений?». На смартфоне Пети Торопыжкина установлено много полезных приложений, которыми он часто пользуется. Однако Петя сам никогда не закрывает приложения. Если необходимость в запущенном приложении отпадает, он отправляет его в фоновый режим, не выгружая из памяти. Все приложения устроены таким образом, что позволяют запустить одновременно лишь одну свою копию, так что если Петя запустит приложение, которое уже было запущено ранее, то оно просто выйдет из фонового режима и второго запущенного его экземпляра не возникнет.

Напишите программу, которая по списку названий запущенных Петей приложений (в хронологическом порядке) определит, сколько сейчас приложений в памяти (в активном или фоновом режиме).

Формат ввода: В первой строке записано целое число n — сколько раз Петя запускал приложения ($1 \leqslant n \leqslant 5\,000$). В следующих n строках перечислены названия приложений в том порядке, в котором их запускал Петя. Название приложения — строка, состоящая из заглавных и строчных латинских букв, длиной не более 20 символов. Приложения, названия которых отличаются только регистром букв, следует считать различными.

Формат вывода: Выведите целое число — количество приложений, находящихся в памяти смартфона после всех манипуляций Пети.

Пример 1

input.txt:	output.txt:
1 NotePad	1

Пример 2

input.txt:	output.txt:
7 NotePad Word NotePad CorelDraw Word AdobePhotoshop Word	4

9.4. «Питание насекомых». У Пети Торопыжкина живёт ручной жучок Ипполит. Уходя в школу, Петя оставляет Ипполиту на письменном столе квадратную плитку шоколада размера 2×2. Центр плитки совпадает с началом декартовой системы координат, которую Петя ввёл на столе, а стороны плитки параллельны осям. Однако Ипполит нетерпелив: он падает с потолка куда придётся. Если он не попадает на плитку, то бежит к ближайшей её точке, где и принимается за трапезу. Напишите программу, которая, зная точку падения Ипполита, находит ближайшую к нему точку шоколадной плитки.

Формат ввода: В единственной строке записаны два вещественных числа, по модулю не превышающие 100, с не более чем двумя знаками после десятичной точки — координаты точки, в которую упал Ипполит. Числа разделены пробелом.

Формат вывода: Выведите два вещественных числа, разделённые пробелом — округлённые до сотых координаты точки, в которую следует бежать Ипполиту.

Пример 1

input.txt:	output.txt:
1.10 1.2	1.00 1.00

Пример 2

input.txt:	output.txt:
0.9 1.1	0.90 1.00

9.5. «Математический бильярд». На координатной плоскости в первом квадранте расположен прямоугольный бильярдный стол так, что его стороны параллельны координатным осям, а один из углов находится в начале координат. Размеры стола $a \times b$ (a — вдоль оси Ox, b — вдоль оси Oy). Точечный бильярдный шар расположен в точке (x, y), находящейся на поверхности стола, и движется со скоростью $v = (v_x, v_y)$. Шар движется без трения (сохраняя величину скорости) и отражается от бортов по закону «угол падения равен углу отражения». Если шар попадает точно в угол, считается, что он мгновенно испытывает два отражения и, стало быть, изменяет вектор скорости на противоположный. Все линейные величины заданы в метрах, скорости — в метрах в секунду. Вопрос: в какой точке шар окажется через t секунд?

Формат ввода: В первой строке записаны числа a и b, во второй — числа x и y, в третьей — числа v_x и v_y. Числа разделены пробелом. В четвёртой строке записано единственное число t. Все числа целые, неотрицательные и не превосходят 10 000. a и b строго положительные.

Формат вывода: Выведите два неотрицательных целых числа, разделённые пробелами — координаты шара через t секунд.

Пример

input.txt:	output.txt:
10 10 1 1 1 2 7	8 5

Фестиваль «Юные интеллектуалы Среднего Урала»
Муниципальный этап Всероссийской олимпиады по информатике
2010 – 2011 учебный год
10 класс

Время выполнения задач — 4 часа
Ограничение по времени — 2 секунды на тест
Ограничение по памяти — 64 мегабайта

10.1. «Игра Баше». Петя Торопыжкин играет со своим братом Сашей в игру Баше. В начале игры выкладывается кучка из n камней. На каждом ходу можно взять из неё один, два или три камня. Выигрывает тот, после чьего хода не останется камней. Первым ходит Петя, как старший. Конечно, в этой игре есть выигрышная стратегия, но ни Петя, ни Саша её не знают, поэтому ходят наобум. Запись партии представляет собой последовательность чисел a_1, a_2, ..., a_k — количество камней, взятых на очередном ходу (все числа a_i равны 1, 2 или 3). Проверьте, корректна ли запись партии (все ходы допустимы и последний камень будет взят ровно на k-м ходу), и, если она корректна, выведите имя победителя.

Формат ввода: В первой строке записаны целые числа n и k ($1 \leqslant n, k \leqslant 10\,000$). В i-й из следующих k строк записано число a_i, равное 1, 2 или 3 — запись i-го хода.

Формат вывода: Если запись партии некорректна (в конце остались камни или были попытки взять камней больше, чем их осталось на тот момент), программа должна выдать слово `INCORRECT`. Если запись корректна и выиграл Петя, программа должна выдать `PETYA`, если Саша — `SASHA`.

Пример 1

input.txt:	output.txt:
5 3 1 1 1	INCORRECT

Пример 2

input.txt:	output.txt:
5 3 1 2 2	PETYA

Пример 3

input.txt:	output.txt:
5 4 2 1 1 1	SASHA

10.2. «Хороший монитор». После того, как родители подарили Пете Торопыжкину новый монитор, его младший брат Саша тоже захотел себе новый монитор. Саша подходит к выбору монитора так: он считает, что монитор *хороший*, если его размеры (выраженные в пикселах) взаимно просты (то есть, не имеют общих делителей, кроме единицы). Известно, что на складе лучшей в городе компьютерной фирмы имеются мониторы с любой шириной w из диапазона $w_{\min} \leqslant w \leqslant w_{\max}$ и любой высотой h из диапазона $h_{\min} \leqslant h \leqslant h_{\max}$. Сколько видов *хороших* мониторов имеется на складе компьютерной фирмы?

Формат ввода: В первой строке записаны целые числа $w_{\min}$ и $w_{\max}$, во второй строке — целые числа $h_{\min}$ и $h_{\max}$ ($2 \leqslant w_{\min} \leqslant w_{\max} \leqslant 1\,000$; $2 \leqslant h_{\min} \leqslant h_{\max} \leqslant 1\,000$). Числа разделены пробелом.

Формат вывода: Выведите единственное неотрицательное целое число — количество видов *хороших* мониторов с размерами, лежащими в заданном диапазоне.

Пример

input.txt:	output.txt:
9 10	2
8 9	

10.3. «Купаться!». Летом Петя Торопыжкин отдыхал в деревне у бабушки. В окрестности их домика расположено идеально круглое озеро единичного радиуса с центром в начале деревенской декартовой системы координат. Однажды днём Петю разморило, и он уснул в точке с координатами (x, y). Конечно, Петя спал не в воде. Проснувшись, он побежал к ближайшей точке пруда. Каковы координаты этой точки?

Формат ввода: В единственной строке записаны вещественные числа x и y, по модулю не превышающие 100, с не более чем двумя знаками после десятичной точки — координаты точки, где спал Петя. Числа разделены пробелом.

Формат вывода: Выведите два вещественных числа, разделённые пробелом — округлённые до сотых координаты точки, в которую следует бежать Пете.

Пример

input.txt:	output.txt:
10 10	0.71 0.71

10.4. «Падение метеорита». Петя Торопыжкин с интересом посмотрел телепередачу о том, что Луна попала в метеоритный поток и подверглась метеоритной бомбардировке. Учёные оказались не готовы к этому событию и фиксировали время падения каждого из n метеоритов на отдельном листочке в формате `YYYY-MM-DD hh:mm:ss`, где `ss` — секунды, `mm` — минуты, `hh` — часы, `DD` — день, `MM` — месяц, `YYYY` — год. Никакие два метеорита не упали одновременно. Бумажки смешались, но необходимо срочно определить k-й в хронологическом порядке метеорит, упавший на Луну: именно он является наиболее интересным для изучения.

Формат ввода: В первой строке записаны целые числа n и k, разделённые пробелом — количество метеоритов и номер метеорита, интересующего учёных ($1 \leqslant k \leqslant n \leqslant 10^5$). В i-й из следующих n строк в указанном формате записано время падения i-го метеорита (метеориты перечислены в произвольном порядке). Диапазоны величин: $1600 \leqslant$ `YYYY` $\leqslant 2500$; $1 \leqslant$ `MM` $\leqslant 12$; `DD` $\geqslant 1$ и не больше числа дней в соответствующем месяце (високосность годов не учитывается); $0 \leqslant$ `hh` $\leqslant 23$; $0 \leqslant$ `mm`, `ss` $\leqslant 59$.

Формат вывода: Выведите время падения k-го в хронологическом порядке метеорита в том же формате, в котором оно описано во входных данных.

Пример

input.txt:	output.txt:
3 2 2010-01-31 23:00:21 2009-08-12 01:01:59 2009-10-05 00:59:00	2009-10-05 00:59:00

10.5. «Интересное число». На парте в кабинете математики Петя Торопыжкин увидел длинную строку, состоящую из цифр и вопросительных знаков. Ему стало интересно, можно ли каждый вопросительный знак заменить на цифру так, чтобы получилось число, кратное одиннадцати. В полученном числе не может быть ведущих нулей. Если таких чисел несколько, Петя хотел бы получить наименьшее возможное.

Формат ввода: Единственная строка содержит только цифры и вопросительные знаки и оканчивается переводом строки. Первый символ этой строки отличен от цифры 0. Длина строки — не менее двух и не более ста символов.

Формат вывода: Выведите наименьшее целое число без ведущих нулей, кратное одиннадцати и получаемое из исходной строки заменой вопросительных знаков на цифры. Если такого числа не существует, выведите IMPOSSIBLE.

Пример 1

input.txt:	output.txt:
1??2	1012

Пример 2

input.txt:	output.txt:
?0903	IMPOSSIBLE

Фестиваль «Юные интеллектуалы Среднего Урала»
Муниципальный этап Всероссийской олимпиады по информатике
2010 – 2011 учебный год
11 класс

Время выполнения задач — 4 часа
Ограничение по времени — 2 секунды на тест
Ограничение по памяти — 64 мегабайта

11.1. «Двойной факториал». Помогите Пете Торопыжкину написать программу, которая по введённому целому числу n вычислит $n!!$. Напомним, что

$$n!! = \begin{cases} 1 \cdot 3 \cdot \ldots \cdot n, & \text{если } n \text{ нечётно,} \\ 2 \cdot 4 \cdot \ldots \cdot n, & \text{если } n \text{ чётно.} \end{cases}$$

Формат ввода: В единственной строке записано целое число n ($1 \leqslant n \leqslant 20$).

Формат вывода: Выведите единственное целое число — результат вычисления.

Пример 1

input.txt:	output.txt:
3	3

Пример 2

input.txt:	output.txt:
4	8

11.2. «На уроке английского-2». В тетрадке по английскому языку Петя Торопыжкин написал длинное слово, состоящее из заглавных букв латинского алфавита. Его заинтересовали *подслова* этого слова (то есть, подстроки данной строки). Двухбуквенные подслова он изучил быстро, а с трёхбуквенными вышла заминка. Помогите Пете: напишите программу, которая по заданному слову выдаёт его подслово из трёх букв, максимальное в лексикографическом порядке. Напомним, что *лексикографическим порядком* называется тот порядок, в котором упорядочены слова в словаре: сначала сравниваются первые буквы, в случае их равенства — вторые, а в случае равенства вторых — третьи.

Формат ввода: Единственная строка содержит слово, которое написал Петя. Это слово состоит только из заглавных латинских букв и имеет длину не менее 3 и не более 200 символов. Слово завершается переводом строки.

Формат вывода: Выведите в единственной строке требуемое трёхбуквенное подслово.

Пример

input.txt:	output.txt:
ABCA	BCA

11.3. «Ночной Дозор». Возможно, вам известна игра «Ночной Дозор». Вот один из вариантов правил. Выделяется n локаций на местности. Команда приезжает в

первую локацию, где получает первую загадку. Отгадывая её, команда получает указание, в какую локацию ехать дальше, переезжает туда, снова отгадывает загадку, переезжает на новое место и так далее. Игра заканчивается, когда команда достигает одной из локаций, отмеченных организаторами как финальные.

Организаторы очередной игры составили план перемещения по локациям в виде списка чисел k_i. Число k_i указывает, что из i-й локации нужно следовать в локацию с номером k_i. Если i-я локация финальная, то $k_i = -1$.

Сейчас они хотят проверить, завершит ли за разумное время свои поездки команда, приехавшая в начале игры на первую локацию.

Формат ввода: В первой строке записано целое число n — количество локаций ($1 \leqslant n \leqslant 1\,000$). В i-й из следующих n строк записано число k_i ($1 \leqslant k_i \leqslant n$ или $k_i = -1$).

Формат вывода: Выведите единственное целое число — количество переездов, которое придётся совершить команде, приехавшей в начале игры на первую локацию, для того, чтобы достичь какой-либо финальной локации, либо `-1`, если план переездов составлен так, что команда, следуя ему, никогда не достигнет никакой финальной локации.

Пример 1

input.txt:	output.txt:
3	2
2	
3	
-1	

Пример 2

input.txt:	output.txt:
3	-1
2	
1	
-1	

11.4. «Магнитное число». Родители Пети Торопыжкина купили магниты на холодильник, имеющие форму цифр: a_0 нулей, a_1 единиц, ..., a_9 девяток. Когда мама сказала, что пора идти спать, Петя ответил, что хочет сначала составить из магнитов минимально возможное целое положительное число, кратное 15, так, чтобы все магниты были задействованы. Напишите программу, которая по набору магнитов у Пети найдёт это число. Ведущие нули в записи числа не допускаются.

Формат ввода: В единственной строке записаны целые числа a_0, a_1, ..., a_9 ($0 \leqslant a_i \leqslant 1\,000$; $a_0 + a_1 + \ldots + a_9 > 0$). Числа разделены пробелами.

Формат вывода: Выведите единственное целое положительное число без ведущих нулей — минимально возможное число, кратное 15, которое Петя может выложить из магнитов, используя их все. Если Петя не может выложить ни одного положительного числа, кратного 15, выведите `-1`.

Пример 1

input.txt:	output.txt:
1 1 0 0 0 1 0 0 0 0	105

Пример 2

input.txt:	output.txt:
1 0 0 0 0 1 0 0 0 0	-1

11.5. «Гуляя по столице». После того, как Петя Торопыжкин отдохнул в деревне у бабушки, он поехал с родителями в Москву. Благодаря стараниям мэра, столица обзавелась большим количеством кольцевых дорог. Эти дороги представляют собой окружности с центром в начале координат и радиусами, выражаемыми всевозможными целыми числами от 1 до 100 000.

Родители Пети наметили культурную программу — осмотр n достопримечательностей, расположенных в точках с координатами (x_i, y_i). Они хотят добраться к первой достопримечательности на такси и затем перемещаться от одной достопримечательности к другой пешком по прямой. Им интересно, пересекает ли очередной отрезок их пути хотя бы одну кольцевую дорогу (касание также считается пересечением). Помогите родителям Пети — напишите программу, которая ответит на этот вопрос.

Формат ввода: В первой строке записано целое число n — количество точек маршрута ($2 \leqslant n \leqslant 100$). В каждой из n следующих строк записаны вещественные числа x_i, y_i, по модулю не превышающие 10 000 и заданные с точностью не более двух знаков после десятичной точки — координаты очередной точки маршрута. Никакие две соседние точки маршрута не совпадают. Числа разделены пробелами.

Формат вывода: Выведите $n-1$ строку, в i-й из которых будет записано `RISKY`, если переход между i-й и $(i+1)$-й точками пересекает или касается хотя бы одной из кольцевых дорог, и `SAFE` в противном случае.

Пример

input.txt:	output.txt:
4	RISKY
2.2 1.91	RISKY
1 1	SAFE
1 -1	
1 -1.1	

Фестиваль «Юные интеллектуалы Среднего Урала»
Муниципальный этап Всероссийской олимпиады по информатике
2010 – 2011 учебный год
8 класс. Разбор решений и идеи тестов

Далее в разборах используются обозначения типов: short int — 2-байтовый целый тип (integer в TurboPascal/BorlandPascal и int в BorlandC++), int — 4-байтовый целый тип (longint в TurboPascal/BorlandPascal/FreePascal, long int в BorlandC++, integer в Delphi/PascalABC и int в VisualC++/C++ Builder), int64 — 8-байтный целый (отсутствует в BorlandPascal и BorlandC++, но присутствует в системах программирования под Windows).

Задачи оцениваются по 100 баллов за каждую; стало быть, полная стоимость пакета — 500 баллов. В рамках одной задачи все тесты считать равноценными.

8.1. «Сажаем картошку». *Весной Петя Торопыжкин помогал бабушке в деревне сажать картошку. Картофельное поле состоит из n рядов, в каждом из которых выкопано m лунок. В одну лунку нужно посадить одну картофелину. В один мешок входит k картофелин. Сколько мешков понадобится для засева бабушкиного поля? (Мешки можно покупать только целиком!)*

Программный комитет оценивает эту задачу как очень простую. Алгоритм решения очевиден: количество мешком есть округлённое вверх отношение общего требуемого числа картофелин к объёму мешка: $\lceil nm/k \rceil$. Здесь $\lceil \cdot \rceil$ — округление вверх. Если в языке нет операции округления вверх (Бейсик, Паскаль), то его можно вычислить как $(nm + k - 1)/k$, где / — операция деления нацело.

Идеи тестов:

1–2. Количество картофелин делится на размер мешка.
3. Нужно $150 \cdot 150 = 22500$ мешков.
4. Размер одного мешка больше, чем требуется картошки.
5. Размер мешка равен требуемому количеству картошки.
6–9. Различные варианты частичного использования последнего мешка:

$nm \equiv 1 (\mathrm{mod}\ k)$;

$nm \equiv (k-1) (\mathrm{mod}\ k)$;

$nm \equiv k/2 (\mathrm{mod}\ k)$;

некоторый случайная часть использования последнего мешка.
10. Максимальный тест: все числа максимальные из допустимых диапазонов.

8.2. «Самый большой монитор». *На день рождения начинающему программисту Пете Торопыжкину родители решили подарить новый монитор. Они знают, что с точки зрения Пети монитор тем лучше, чем больше на нем пикселов. Напишите программу, которая из заданного набора мониторов (для*

каждого монитора известны его ширина и высота в пикселах) выберет наилучший.

Жюри считает данную задачу достаточно простой: необходимо реализовать классический алгоритм поиска максимального элемента в последовательности. Отличие от классического алгоритма в том, что элементы последовательности заданы как произведения пар чисел.

Заметим, что нет необходимости загружать все числа в память: обработка очередного элемента происходит при его считывании.

Идеи тестов:

1. Почти пример.
2. 5 чисел, не превосходящих 5.
3. 5 чисел, не превосходящих 15.
4. 10 чисел, не превосходящих 50.
5. 50 чисел, не превосходящих 50.
6. 100 чисел, не превосходящих 100.
7. 150 чисел, не превосходящих 150.
8. 10 чисел, не превосходящих 10 в порядке возрастания.
9. 150 чисел, не превосходящих 150 в порядке возрастания.
10. 150 чисел, не превосходящих 150 в порядке убывания.

8.3. «На уроке английского». *В тетради по английскому языку Петя Торопыжкин написал длинное слово, состоящее только из заглавных букв латинского алфавита. Ему стало интересно, сколько различных букв латинского алфавита содержит это слово. Помогите ему — напишите программу, которая по введённой строке будет выдавать эту информацию.*

Данная задача имеет несколько возможных разумных алгоритмов решения. Сложность её заключается в том, что восьмиклассники могут неуверенно работать со строками.

Один из возможных алгоритмов решения: заводим логический массив из 26 элементов и инициализируем его false'ами. Затем перебираем символы строки, и для каждого символа выставляем в true элемент массива, соответствующий этой букве. Индекс элемента, соответствующего i-ому символу строки `str`, может быть вычислен как `ord(str[i]) - ord('A') + 1` в Паскале и `str[i] - 'A' + 1` в Си. После того, как все символы строки обработаны, считаем количество ячеек нашего массива, выставленных в true.

Идеи тестов:

Тесты 1–4: мало различных букв.
Тесты 5, 10, 18: только одна буква.
6, 9, 11, 16, 17: много различных букв в случайном порядке.

Тесты 12–15: среднее количество различных букв (около 10).
Тесты 7, 19: много различных букв в порядке возрастания.
Тесты 8, 20: много различных букв в порядке убывания.

Тесты в одной категории отличаются длиной строки.

8.4. «Сложение дробей». *Успешно справившись с домашним заданием по английскому языку и сев за компьютер с новым монитором, Петя Торопыжкин занялся на компьютере математикой. Однако результат первого же действия сильно удивил его: он разделил 10 на 3 и вместо ожидаемого $3\frac{1}{3}$ или хотя бы $\frac{10}{3}$ увидел на экране* `3.33333333334`*. Чтобы исправить этот «недостаток» компьютера, Петя решил научить компьютер работать с обыкновенными дробями, а для начала — складывать их. Помогите Пете — напишите программу, которая складывает две положительные обыкновенные дроби и выдаёт результат в виде обыкновенной несократимой дроби (возможно, неправильной). Дроби задаются числителем и знаменателем.*

Программный комитет считает данную задачу средней сложности. Для решения требуется реализовать алгоритм сложения двух дробей, причём не в самом экономном варианте (не с приведением к наименьшему общему знаменателю, а просто к тривиальному общему знаменателю, равному произведению знаменателей слагаемых):

$$\frac{p_1}{q_1} + \frac{p_2}{q_2} = \frac{p_1 q_2 + p_2 q_1}{q_1 q_2} = \frac{p}{q}.$$

Затем нужно сократить полученную дробь p/q. Здесь от школьников требуется реализовать тот или иной вариант алгоритма Евклида.

Идеи тестов:

1–3. Нет сокращения.
4. Есть сокращение, в результате — дробь.
5. Есть сокращение, в результате — целое число.
6–7. Одно из слагаемых целое.
8. Сумма двух целых.
9. Слагаемые около 100.
10–12. Крайние случаи.
13–20. Случайные тесты.

8.5. «Сортируем числа». *Дана последовательность, состоящая из чисел 1, 2 и 3. Длина последовательности не превосходит 200. Можно брать любые два элемента и менять их местами. Требуется за наименьшее число таких перестановок поставить на первые места в последовательности все единички, дальше все двойки и потом все тройки.*

Программный комитет считает эту задачу сложной. Важным моментом является то, что многие школьники начнут считать перестановки в известном им методе сортировки, скорее всего — в методе «пузырька», что, конечно же, неправильно: переставлять можно не только соседние элементы.

Идея решения следующая. После того, как все числа считаны в память, посчитаем q_1 — количество единиц, q_2 — количество двоек, q_3 — количество троек в последовательности. Зная эту информацию, мы можем определить «зону единиц», «зону двоек» и «зону троек» — части массива, где в отсортированном виде стоят единицы, двойки и тройки. Собственно, зона единиц — это элементы с первого по q_1-й, зона двоек — элементы с (q_1+1)-го по (q_1+q_2)-й, зона троек — элементы с (q_1+q_2+1)-го по $(q_1+q_2+q_3)$-й.

Затем посчитаем количество a_{11} единиц, a_{21} двоек и a_{31} троек, стоящих в зоне единиц. Аналогично, посчитаем a_{12}, a_{22}, a_{32} — количества единиц, двоек, троек в зоне двоек и a_{13}, a_{23}, a_{33} — количества единиц, двоек, троек в зоне троек. Заметим, что единицы в зоне единиц, двойки в зоне двоек и тройки в зоне троек уже стоят на своих местах, их не надо никуда передвигать.

Произведём обмены между зоной единиц и двоек, которые обменивают местами единицы и двойки, ставя их на места. Таких обменов, будет $\min(a_{12}, a_{21})$. Аналогично, обменов между зоной единиц и троек будет $\min(a_{13}, a_{31})$, а между зоной двоек и троек — $\min(a_{32}, a_{23})$. После этого, либо будет достигнуто упорядоченное состояние последовательности, либо в зоне единиц кроме самих единиц останутся либо только двойки, либо только тройки. Действительно, пусть там остались и двойки, и тройки. При этом не все единицы на своих местах. Значит, единицы есть либо в зоне двоек, либо в зоне троек, и можно провести хотя бы ещё один обмен, что противоречит тому, что все «прямые» обмены уже произведены.

Пусть в зоне единиц остались двойки (случай троек рассматривается аналогично). Тогда в зоне двоек кроме двоек только тройки, а в зоне троек кроме троек — только единицы. При этом количество «чужих» цифр в каждой зоне одинаково и равно $a_{21} - a_{12}$. Каждая тройка чисел, стоящая не в своей зоне упорядочивается за два обмена: единица из зоны троек меняется с двойкой, а потом эта двойка — с тройкой из зоны двоек. Таким образом, количество дополнительных обменов равно $2(a_{21} - a_{12})$.

В случае, когда в зоне единиц остались кроме единиц только тройки, количество дополнительных обменов равно $2(a_{31} - a_{13})$.

Чтобы получить общую формулу для обоих случаев, надо взять максимум из этих двух чисел: в общем случае количество дополнительных обменов равно $\max\big(2(a_{21} - a_{12}), 2(a_{31} - a_{13})\big)$. И в итоге имеем, что общее число обменов равно

$$\min(a_{12}, a_{21}) + \min(a_{13}, a_{31}) + \min(a_{32}, a_{23}) + \max\big(2(a_{21} - a_{12}), 2(a_{31} - a_{13})\big).$$

Заметим, что ровно одно из выражений внутри максимума положительно.

Впрочем, поскольку все величины a_{ij} связаны между собой и длиной последовательности, можно придумать и другие выражения для минимально необходи-

мого числа обменов.

Меньшим количеством обменов обойтись нельзя. Действительно, каждый обмен может поставить на своё место два, одно или ноль чисел. В предложенном алгоритме каждый обмен ставит на своё место или два, или одно число, притом обменов, ставящих на место одно число, минимально возможное количество.

Идеи тестов:

1–3. Последовательность состоит из одинаковых цифр.
4–6. Последовательность состоит из цифр только двух видов.
7. В последовательности есть все три вида чисел, но уже упорядочена.
8–10. В последовательности есть все три вида чисел, но она неупорядочена; различные длины, меньшие 200.
11–20. Тесты с теми же идеями, что и в тестах 1–10, но большого размера.
21–25. Случайные тесты разной длины.

Фестиваль «Юные интеллектуалы Среднего Урала»
Муниципальный этап Всероссийской олимпиады по информатике
2010 – 2011 учебный год
9 класс. Разбор решений и идеи тестов

Далее в разборах используются обозначения типов: short int — 2-байтовый целый тип (integer в TurboPascal/BorlandPascal и int в BorlandC++), int — 4-байтовый целый тип (longint в TurboPascal/BorlandPascal/FreePascal, long int в BorlandC++, integer в Delphi/PascalABC и int в VisualC++/C++ Builder), int64 — 8-байтный целый (отсутствует в BorlandPascal и BorlandC++, но присутствует в системах программирования под Windows).

Задачи оцениваются по 100 баллов за каждую; стало быть, полная стоимость пакета — 500 баллов. В рамках одной задачи все тесты считать равноценными.

9.1. «Кратность делителя». *На парте в кабинете математики Петя Торопыжкин обнаружил натуральные числа n и m, причём m большие единицы. Его заинтересовало, является ли число m делителем числа n и, если да, то какой кратности. Помогите ему — напишите программу, отвечающую на этот вопрос.*

Программный комитет считает эту задачу предельно простой. Алгоритм решения: положить текущее значение переменной равным n; положить счётчик равным 0; пока остаток текущего значения переменной на m равен 0 (то есть, текущее значение делится на m), делим текущее значение на m и увеличиваем счётчик на один. Результирующее значение счётчика и есть результат. Иными словами, нужно делить n на m, пока делится нацело, и считать количество успешных делений.

В силу указанных ограничений важно использовать тип int.

Идеи тестов:

1–2. $n = 1$.
3–7. случайные тесты с небольшими n и m.
8–10. $n = m$, числа разных величин.
11–13. n делится на степень двойки.
14–18. n делится на степень тройки.
19–20. n делится на степень 10.

9.2. «Кольцо из проводов». *У Пети Торопыжкина есть набор проводов, концы которых обжаты в разъёмы двух типов. Существует три типа проводов: 1–1 — их у Пети n штук, 2–2 — их m штук и 1–2 — их k штук (указаны типы разъёмов на концах). При этом два разъёма можно соединять друг с другом только если они одного типа. Напишите программу, которая сможет по заданным n, m и k определить, какое минимальное количество проводов нужно исключить из набора, чтобы из оставшихся можно было собрать кольцо.*

Программный комитет считает эту задачу простой.

Наблюдения:

1° Любое количество проводов типа 1–1 можно соединить в один длинный провод типа 1–1. Аналогично, любое количество проводов типа 2–2 можно соединить в один длинный провод типа 2–2.

2° Провода типа 1–2 можно использовать только парами: если у нас есть один провод такого типа, то ни провода 1–1, ни 2–2 по отдельности или в совокупности не смогут замкнуть кольца.

3° Любое чётное число проводов типа 1–2 можно собрать в один длинный провод типа 1–1 или 2–2.

4° Если имеется хотя бы два провода типа 1–2, то можно использовать все провода типов 1–1 и 2–2: они соединяются между собой, после чего длинный провод типа 1–1 подсоединяется к 1-концам пары проводов типа 1–2, а длинный провод типа 2–2 подсоединяется к 2-концам той же пары.

Выводы:

1) если имеется 1 или 0 проводов типа 1–2, то нельзя соединить вместе провода типа 1–1 и 2–2. Стало быть, надо выкинуть провод типа 1–2 и те из однотипных проводов, которых меньше. Соответственно, если других проводов нет, то кольцо не собрать.

2) если проводов типа 1–2 имеется чётное число, большее или равное 2, то можно использовать все имеющиеся провода, выкидывать ничего не надо.

3) если проводов типа 1–2 имеется нечётное число, большее или равное 3, то надо выкинуть один провод этого типа и соединить в кольцо все оставшиеся провода.

Идеи тестов:

1. Кольцо не собрать. Нет проводов.
2. Кольцо не собрать. Один провод третьего типа.
3. Два провода типа 1–2. Ничего убирать не надо.

4–5. Кольцо из одного провода (типа 1–1 или 2–2). Ничего убирать не надо.

6–7. Кольцо из одного провода (типа 1–1 или 2–2). Один провод нужно убрать (другого типа).

8. Только провода типа 1–2. Ничего убирать не надо.
9. Только провода типа 1–2. Нужно убрать один.

10–11. Только провода типа 1–1, проводов типа 1–2 мало (ноль или один). Убираем все провода типа 2–2.

12–13. Только провода типа 2–2, проводов типа 1–2 мало (ноль или один). Убираем все провода типа 1–1.

14–15. Нужно убрать провода одного из типов 1–1 или 2–2 и единственный имеющийся проводов типа 1–2.

16. Только провода 1–1 и 2–2. Нет разницы какие убирать.
17. Только провода 1–1 и 1–2. Нужно убрать один.

18. Только провода 1–1 и 1–2. Ничего убирать не надо.
19. Только провода 2–2 и 1–2. Нужно убрать один.
20. Только провода 2–2 и 1–2. Ничего убирать не надо.
21–22. Провода всех типов. Нужно убрать один провод типа 2–2.
23–25. Провода всех типов. Ничего убирать не нужно.

9.3. «Сколько приложений?». *На смартфоне Пети Торопыжкина установлено много полезных приложений, которыми он часто пользуется. Однако Петя сам никогда не закрывает приложения. Если необходимость в запущенном приложении отпадает, он отправляет его в фоновый режим, не выгружая из памяти. Все приложения устроены таким образом, что позволяют запустить одновременно лишь одну свою копию, так что если Петя запустит приложение, которое уже было запущено ранее, то оно просто выйдет из фонового режима и второго запущенного его экземпляра не возникнет.*

Напишите программу, которая по списку названий запущенных Петей приложений (в хронологическом порядке) определит, сколько сейчас приложений в памяти (в активном или фоновом режиме).

Программный комитет оценивает эту задачу как среднюю.

Фактически в задаче спрашивается, сколько различных строк имеется в заданном списке. Один из вариантов решения: считать список строк и отсортировать его, после чего просто посчитать количество различных блоков одинаковых строк.

Другой вариант: при считывании каждую новую строку сравнивать со всеми строками, запомненными к этому моменту. Если такая строка уже запомнена, то с новой строкой ничего не делать. Если же новая строка не найдена в хранилище, то запоминаем её. Размер хранилища в конце — количество уникальных строк. Хотя это решение, вообще говоря, более медленное и в неоптимальной реализации при указанных ограничениях на входные данные может не укладываться в ограничения по времени.

Еще одна сложность заключается в том, что данные сами по себе могут занимать до 100 килобайт, что делает сложным размещение этих данных в программах, написанных для MS DOS.

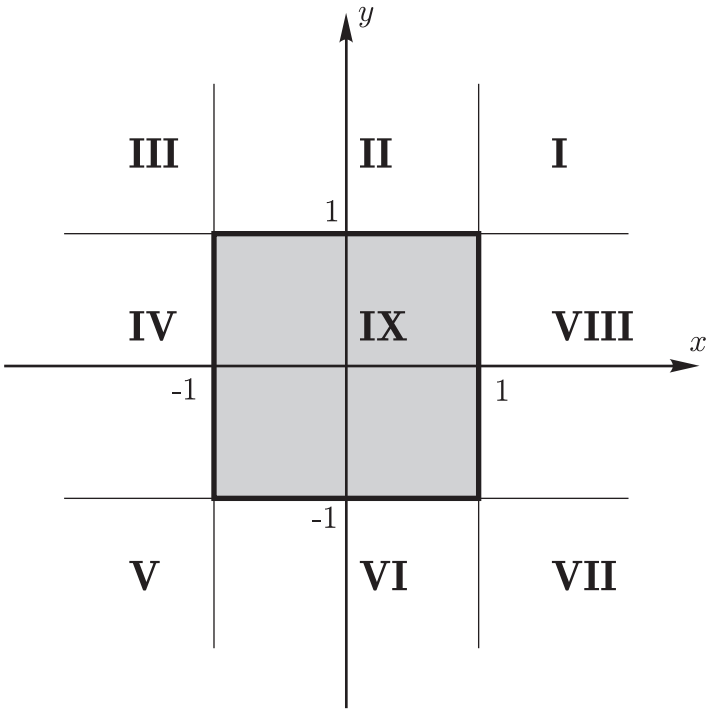
Идеи тестов:

1. Первая программа отличается от следующих.
2. Все программы различные.
3. Все программы одинаковые.
4. Последняя программа отличается от предыдущих.
5. Две разные программы.
6–7. Случайные тесты.
8–10. Большие случайные тесты.

9.4. «Питание насекомых». *У Пети Торопыжкина живёт ручной жучок Ипполит. Уходя в школу, Петя оставляет Ипполиту на письменном столе квадратную плитку шоколада размера 2×2. Центр плитки совпадает с началом декартовой системы координат, которую Петя ввёл на столе, а стороны плитки параллельны осям. Однако Ипполит нетерпелив: он падает с потолка куда придётся. Если он не попадает на плитку, то бежит к ближайшей её точке, где и принимается за трапезу. Напишите программу, которая, зная точку падения Ипполита, находит ближайшую к нему точку шоколадной плитки.*

Программный комитет оценивает сложность этой задачи как «выше среднего». Сама по себе задача несложна, но она связана с вычислительной геометрией, которая для школьников традиционно является непростым разделом информатики.

Рассмотрим рисунок:



Из рисунка видно, что прямые, проведённые через стороны квадрата, делят плоскость на 9 областей, обозначенных римскими цифрами. Каждая область задаётся неравенствами относительно координат точки и имеет явное выражение ближайшей точки A квадрата:

I: $x > 1$, $y > 1$, $A(1, 1)$;
II: $|x| \leqslant 1$, $y > 1$, $A(x, 1)$;
III: $x < -1$, $y > 1$, $A(-1, 1)$;
IV: $x < -1$, $|y| \leqslant 1$, $A(-1, y)$;
V: $x < -1$, $y < -1$, $A(-1, -1)$;
VI: $|x| \leqslant 1$, $y < -1$, $A(x, -1)$;
VII: $X > 1$, $y < -1$, $A(1, -1)$;
VIII: $x > 1$, $|y| \leqslant 1$, $A(1, y)$;
IX: $|x| \leqslant 1$, $|y| \leqslant 1$, $A(x, y)$.

Соответственно, по этим формулам и находится ближайшая точка — никаких дополнительных вычислений не требуется, точность результирующих координат совпадает с точностью координат исходной точки.

Идеи тестов:

1–6. Точка внутри квадрата.
7. Ближайшая точка $(1, 1)$.
8–10. Ближайшая точка $(1, a)$.
11. Ближайшая точка $(1, -1)$.
12–14. Ближайшая точка $(a, -1)$.
15. Ближайшая точка $(-1, -1)$.
16–18. Ближайшая точка $(-1, a)$.
19. Ближайшая точка $(-1, 1)$.
20–22. Ближайшая точка $(a, 1)$.
23–27. Точка внутри квадрата на оси.
28–35. Точка внутри квадрата.
36–50. Случайные тесты.

9.5. «Математический бильярд». *На координатной плоскости в первом квадранте расположен прямоугольный бильярдный стол так, что его стороны параллельны координатным осям, а один из углов находится в начале координат. Размеры стола $a \times b$ (a — вдоль оси Ox, b — вдоль оси Oy). Точечный бильярдный шар расположен в точке (x, y), находящейся на поверхности стола, и движется со скоростью $v = (v_x, v_y)$. Шар движется без трения (сохраняя величину скорости) и отражается от бортов по закону «угол падения равен углу отражения». Если шар попадает точно в угол, считается, что он мгновенно испытывает два отражения и, стало быть, изменяет вектор скорости на противоположный. Все линейные величины заданы в метрах, скорости — в метрах в секунду. Вопрос: в какой точке шар окажется через t секунд?*

Данная задача программным комитетом оценивается как сложная. Она тоже может быть отнесена к геометрическим задачам, поскольку наиболее разумный способ её решения связан с координатами.

По входным данным вычислим следующие координаты $\bar{x} = x + v_x * t$, $\bar{y} = y + v_y * t$ (конечные координаты точки, если бы она двигалась по плоскости без ограничений).

Предположим, что стенки у стола имеются только снизу и слева, то есть, что стол ограничен только координатными осями. Тогда понятно, что конечные координаты точки в этом случае будут $|\bar{x}|$, $|\bar{y}|$: точка может лишь по разу отразиться от каждой из двух стенок, поменяв знак координаты.

Наоборот, представим, что стол ограничен сверху прямой $y = b$. Тогда конечные координаты точки будут: абсцисса равна $\bar{x}$; ордината равна $\bar{y}$, если $\bar{y} \leqslant b$, и $2b - \bar{y}$, если $\bar{y} > b$.

Аналогично, если стол ограничен справа прямой $x = a$, то ордината конечной точки будет равна $\bar{y}$, а абсцисса — либо $\bar{x}$, если $\bar{x} \leqslant a$, либо $2a - \bar{x}$, если $\bar{x} > a$.

Таким образом, мы установили, как меняются координаты конечной точки при единичном отражении от той или иной стенки стола.

Возможный алгоритм полного решения выглядит следующим образом:

1. $\bar{x} = x + v_x * t$, $\bar{y} = y + v_y * t$.
2. пока $\bar{x} < 0$ или $\bar{x} > a$ делать

   если $\bar{x} < 0$ то $\bar{x} \leftarrow -\bar{x}$ иначе $\bar{x} \leftarrow 2a - \bar{x}$.
3. пока $\bar{y} < 0$ или $\bar{y} > b$ делать

   если $\bar{y} < 0$ то $\bar{y} \leftarrow -\bar{y}$ иначе $\bar{y} \leftarrow 2b - \bar{x}$.
4. вывести $\bar{x}$, $\bar{y}$.

Идеи тестов:

1. $t = 0$.
2. $t = 0$, шар в правом верхнем углу.
3. $t = 0$, шар в правом нижнем углу.
4. $t = 0$, шар в левом верхнем углу.
5. $t = 0$, шар в левом нижнем углу.
6. $v = (0, 0)$.
7. $v = (0, 0)$, шар в правом верхнем углу.
8. $v = (0, 0)$, шар в правом нижнем углу.
9. $v = (0, 0)$, шар в левом верхнем углу.
10. $v = (0, 0)$, шар в левом нижнем углу.
11. шар долетает до правого верхнего угла и там останавливается.
12. шар долетает до правого нижнего угла и там останавливается.
13. шар долетает до левого верхнего угла и там останавливается.
14. шар долетает до левого нижнего угла и там останавливается.
15. шар на диагонали прямоугольника.

16–25. случайные тесты.

Фестиваль «Юные интеллектуалы Среднего Урала»
Муниципальный этап Всероссийской олимпиады по информатике
2010 – 2011 учебный год
10 класс. Разбор решений и идеи тестов

Далее в разборах используются обозначения типов: short int — 2-байтовый целый тип (integer в TurboPascal/BorlandPascal и int в BorlandC++), int — 4-байтовый целый тип (longint в TurboPascal/BorlandPascal/FreePascal, long int в BorlandC++, integer в Delphi/PascalABC и int в VisualC++/C++ Builder), int64 — 8-байтный целый (отсутствует в BorlandPascal и BorlandC++, но присутствует в системах программирования под Windows).

Задачи оцениваются по 100 баллов за каждую; стало быть, полная стоимость пакета — 500 баллов. В рамках одной задачи все тесты считать равноценными.

10.1. «Игра Баше». *Петя Торопыжкин играет со своим братом Сашей в игру Баше. В начале игры выкладывается кучка из n камней. На каждом ходу можно взять из неё один, два или три камня. Выигрывает тот, после чьего хода не останется камней. Первым ходит Петя, как старший. Конечно, в этой игре есть выигрышная стратегия, но ни Петя, ни Саша её не знают, поэтому ходят наобум. Запись партии представляет собой последовательность чисел a_1, a_2, ..., a_k — количество камней, взятых на очередном ходу (все числа a_i равны 1, 2 или 3). Проверьте, корректна ли запись партии (все ходы допустимы и последний камень будет взят ровно на k-м ходу), и, если она корректна, выведите имя победителя.*

Задача оценивается программным комитетом как простая, поскольку носит несложный технический характер.

Единственная проверка заключается в том, что сумма взятых камней (сумма всех чисел-ходов) должна равняться начальному числу камней n. Если это не так, то запись неверна. Если так, то чётность числа ходов k укажет победителя: если число ходов нечётно, то выиграл Петя. Если число ходов чётно, то выиграл Саша.

Идеи тестов:

1–3. Корректные записи.
4. Сумма чисел больше n.
5. Сумма чисел меньше n
6–10. Те же тесты, только для n порядка 10000.

10.2. «Хороший монитор». *После того, как родители подарили Пете Торопыжкину новый монитор, его младший брат Саша тоже захотел себе новый монитор. Саша подходит к выбору монитора так: он считает, что монитор хороший, если его размеры (выраженные в пикселах) взаимно просты (то есть, не имеют общих делителей, кроме единицы). Известно, что на складе лучшей в*

городе компьютерной фирмы имеются мониторы с любой шириной w из диапазона $w_{min} \leqslant w \leqslant w_{max}$ и любой высотой h из диапазона $h_{min} \leqslant h \leqslant h_{max}$. Сколько видов хороших мониторов имеется на складе компьютерной фирмы?

Данная задача идейно несложна, однако привлекает для своей реализации алгоритм Евклида. Поэтому программный комитет оценивает сложность как «ниже средней».

Подразумевается лобовое решение: в двойном цикле `for` перебор всех возможных пар $\{(w,h) : w_{\min} \leqslant w \leqslant w_{\max},\ h_{\min} \leqslant h \leqslant h_{\max}\}$, проверка каждой пары на взаимную простоту и подсчёт количества взаимно простых пар.

Идеи тестов:

1–4. Маленькие тесты.
5–8. Средние тесты (максимальное значение до 500).
9–13. Разные случайные тесты.
14–16. Тесты, близкие к максимальным ограничениям.
17–20. Минимальные и максимальные тесты.

10.3. «Купаться!». *Летом Петя Торопыжкин отдыхал в деревне у бабушки. В окрестности их домика расположено идеально круглое озеро единичного радиуса с центром в начале деревенской декартовой системы координат. Однажды днём Петю разморило, и он уснул в точке с координатами (x, y). Конечно, Петя спал не в воде. Проснувшись, он побежал к ближайшей точке пруда. Каковы координаты этой точки?*

Идея решения тривиальна: берём начальный вектор (x, y) (он отличен от нулевого по условию) и нормализуем, то есть, делим на длину. Результат выводим соответствующим образом. Сложность задачи оценивается программным комитетом как простая.

Идеи тестов:

10 случайных тестов.

10.4. «Падение метеорита». *Петя Торопыжкин с интересом посмотрел телепередачу о том, что Луна попала в метеоритный поток и подверглась метеоритной бомбардировке. Учёные оказались не готовы к этому событию и фиксировали время падения каждого из n метеоритов на отдельном листочке в формате `YYYY-MM-DD hh:mm:ss`, где `ss` — секунды, `mm` — минуты, `hh` — часы, `DD` — день, `MM` — месяц, `YYYY` — год. Никакие два метеорита не упали одновременно. Бумажки смешались, но необходимо срочно определить k-й в хронологическом порядке метеорит, упавший на Луну: именно он является наиболее интересным для изучения.*

Решение данной задачи подразумевает реализацию «хорошей» (быстрой) сортировки и выбор указанного элемента из упорядоченной последовательности. Из-за

необходимости реализовать быструю сортировку программный комитет оценивает эту задачу как среднюю. Те, кто реализую квадратичную сортировку («пузырёк» или сортировку вставкой), получат за эту задачу 60-70% от максимального балла.

Также при написании программы надо аккуратно реализовать сравнение моментов времени, записанных в указанном формате. Впрочем, формат записи времени подобран так, что хронологический порядок совпадает с лексикографическим порядком, так что можно использовать простое сравнение строк.

Идеи тестов:

1–3. Случайный порядок, годы разные.
4–5. Случайный порядок, годы одинаковые.
6–7. Обратный порядок, годы разные.
8. Последовательность упорядочена, годы разные.
9. Последовательность упорядочена, годы одинаковые.
10. Обратный порядок, годы одинаковые.
11–20. Те же тесты, но большого размера.

10.5. «Интересное число». *На парте в кабинете математики Петя Торопыжкин увидел длинную строку, состоящую из цифр и вопросительных знаков. Ему стало интересно, можно ли каждый вопросительный знак заменить на цифру так, чтобы получилось число, кратное одиннадцати. В полученном числе не может быть ведущих нулей. Если таких чисел несколько, Петя хотел бы получить наименьшее возможное.*

Программный комитет оценивает эту задачу как очень сложную, поскольку при решении используются весьма нетривиальные идеи.

Напомним, что признак делимости на 11 звучит следующим образом: число делится на 11 тогда и только тогда, когда делится на 11 разность сумм цифр, стоящих на чётных и нечётных позициях. В дальнейшем первой позицией будем считать позицию старшего разряда числа.

При считывании числа вычисляются S_1 и q_1 — сумма цифр, стоящих на нечётных позициях, и количество вопросительных знаков, стоящих на нечётных позициях. Также вычисляются величины S_2 и q_2, связанные с чётными позициями. Кроме того, запоминается, является ли символ в старшем разряде вопросительным знаком (поскольку он обрабатывается особо: вместо него нельзя подставить ноль).

Тогда суммы цифр, стоящих на нечётных и чётных позициях могут меняться в диапазонах $I_1 = [S_1, S_1 + 9q_1]$ и $I_2 = [S_2, S_2 + 9q_2]$, соответственно. Нижняя граница первого диапазона должна быть увеличена на 1, если первый символ в последовательности — вопросительный знак.

Соответственно встаёт вопрос: есть ли в этих диапазонах два числа, разность которых кратна 11, и, если да, то какая пара порождает минимальное число?

Общая идея решения может быть изложена следующим образом: рассмотрим самые правые вопросики на чётных и нечётных позициях (если таковые есть). Перебираем пары цифр, выставляя их на эти места; перебор ведётся либо по полному набору от (0,0) до (9,9) или же (как это описано ниже) по набору, сокращённому, исходя из смысла задачи. Остальные вопросы заменяем минимально возможными цифрами, отслеживая отсутствие ведущих нулей. Из получаемых чисел находи минимальное.

Изложим теперь идею решения формальным образом.

Заметим, что для всякого числа $a_1 \in I_1$ несложно выделяется набор чисел $A_2(a_1) = \{a_{2,k}\} \subset I_2$ таких, что разность $a_1 - a_{2,k}$ кратна 11. Несложно понять, что для получения минимального результирующего числа достаточно рассматривать число $\bar{a}_2$, минимальное в наборе $A_2(a_1)$. Кроме того, если диапазон I_1 имеет длину, бóльшую 11, то достаточно рассмотреть только первые 11 чисел из этого диапазона.

Таким образом, в целом, процедура решения выглядит следующим образом:

1) Считываем последовательность, строим диапазоны I_1 и I_2.

2) Корректируем диапазон I_1: увеличиваем нижнюю границу на 1 при необходимости. После этого, если его длина больше 11, уменьшаем его верхнюю границу так, чтобы длина стала равной 11. Обозначим новый диапазон через $\tilde{I}_1$.

3) Для каждого числа $a_1 \in \tilde{I}_1$ находим минимальное число $a_2 \in I_2$ такое, что разность $a_1 - a_2$ кратна 11. Понятно, что эти минимальные числа a_2 исчерпывают начальную часть диапазона I_2. Заметим также, что для каких-то или для всех a_1 может не найтись соответствующих a_2. Если ни для одного a_1 число a_2 не нашлось, то останавливаем работу и выдаём отрицательное заключение.

4) Для каждой полученной пары a_1, a_2 восстанавливаем минимальное число, дающую соответствующие суммы цифр. Для этого находим величины $b_i = a_i - S_i$ — недостатки сумм и распределяем их по разрядам. Соответственно, величину b_1 распределяем по нечётным позициям, начиная с самой правой, а величину b_2 — по чётным позициям. Распределение заключается в том, что мы заменяем вопросительный знак на b_i, если $b_i < 10$, или на девятку, уменьшая при этом b_i на 9 и переходя к следующей позиции. Если же в старшем разряде числа стоит вопросительный знак, то в случае, когда a_1 исчерпалось раньше позиции старшего разряда, то отщепляем от его финального значения единичку, которую помещаем в старший разряд (чтобы не было ведущих нулей).

5) Перебирая пары a_1, a_2 и получая минимальные числа, дающие соответствующие суммы цифр, выбираем из этих чисел минимальное. Оно и будет ответом.

В принципе, при небольшом количестве вопросительных знаков в исходной последовательности можно применять «лобовой» метод — полный перебор всех возможных комбинаций цифр, подставляемых вместо вопросительных знаков. Однако величина перебора очень быстро нарастает с количеством вопросительных знаков, и если в числе больше десяти-двенадцати вопросительных знаков, такая

программа не уложится ограничения по времени.

Идеи тестов:

1. Нет вопросиков, не делится на 11.
2. Нет вопросиков, делится на 11.
3. Есть вопросики, не в начале, решения нет.

4–5. Есть вопросики не в начале, решение есть.

6. Есть вопросики, в том числе и в начале, решения нет: нельзя расставить цифры так, чтобы разность между суммами была кратна 11.
7. Есть вопросики, в том числе и в начале, решения нет: расставить цифры так, чтобы разность между суммами была кратна 11, можно только в случае возникновения ведущих нулей.

8–10. Есть вопросики, в том числе и в начале, решение есть.

11. Только вопросики.
12. Единственный вопросик.
13. Два вопросика.
14. Вопросик в разряде единиц, решения нет

15–25. Тесты с теми же идеями, что и в первой половине, но с максимальной длиной числа.

Фестиваль «Юные интеллектуалы Среднего Урала»
Муниципальный этап Всероссийской олимпиады по информатике
2010 – 2011 учебный год
11 класс. Разбор решений и идеи тестов

Далее в разборах используются обозначения типов: short int — 2-байтовый целый тип (integer в TurboPascal/BorlandPascal и int в BorlandC++), int — 4-байтовый целый тип (longint в TurboPascal/BorlandPascal/FreePascal, long int в BorlandC++, integer в Delphi/PascalABC и int в VisualC++/C++ Builder), int64 — 8-байтный целый (отсутствует в BorlandPascal и BorlandC++, но присутствует в системах программирования под Windows).

Задачи оцениваются по 100 баллов за каждую; стало быть, полная стоимость пакета — 500 баллов. В рамках одной задачи все тесты считать равноценными.

11.1. «Двойной факториал». *Помогите Пете Торопыжкину написать программу, которая по введённому целому числу n вычислит $n!!$. Напомним, что*

$$n!! = \begin{cases} 1 \cdot 3 \cdot \ldots \cdot n, & \text{если } n \text{ нечётно,} \\ 2 \cdot 4 \cdot \ldots \cdot n, & \text{если } n \text{ чётно.} \end{cases}$$

Задача чисто техническая, не привлекает никаких особых знаний, поэтому оценивается программным комитетом как очень простая. Однако в своём решении эта задача использует беззнаковый тип int, так что те, кто программируют под DOS и не используют unsigned long int, не получат полного балла.

Идеи тестов:

1–4. Маленькие тесты, ответы влезают в short.
5–10. Большие тесты.

Тесты с чётными номерами содержат чётное число n, с нечётными — нечётное.

11.2. «На уроке английского-2». *В тетрадке по английскому языку Петя Торопыжкин написал длинное слово, состоящее из заглавных букв латинского алфавита. Его заинтересовали подслова этого слова (то есть, подстроки данной строки). Двухбуквенные подслова он изучил быстро, а с трёхбуквенными вышла заминка. Помогите Пете: напишите программу, которая по заданному слову выдаёт его подслово из трёх букв, максимальное в лексикографическом порядке. Напомним, что лексикографическим порядком называется тот порядок, в котором упорядочены слова в словаре: сначала сравниваются первые буквы, в случае их равенства — вторые, а в случае равенства вторых — третьи.*

Программным комитетом подразумевается лобовое решение данной задачи, привлекающее алгоритм поиска максимального элемента в наборе. Сложность оценивается как «ниже средней». Требуется умение работы со строками: вычисление длины, выделение подстроки, сравнение строк.

Идеи тестов:

1. Длина слова равна 3.
2. Длина слова равна 4. Нужно выбрать последние три символа.
3. Длина слова равна 4. Нужно выбрать первые три символа.
4. Длинная строка. Нужно выбрать последние три символа.
5. Длинная строка. Нужно выбрать первые три символа.
6. Все буквы в слове одинаковые.
7. Все буквы в слове кроме одной одинаковые. Эта буква больше остальных.
8. Все буквы в слове кроме одной одинаковые. Эта буква меньше остальных.

9–10. Из нескольких слов, совпадающих по первым двум буквам, нужно выбрать лучшее.

11. Тест против поиска подпоследовательности из трёх символов.
12. Строка заканчивается на ZZ, но брать нужно другой кусок.

13–20. Случайные тесты.

11.3. «Ночной Дозор». *Возможно, вам известна игра «Ночной Дозор». Вот один из вариантов правил. Выделяется n локаций на местности. Команда приезжает в первую локацию, где получает первую загадку. Отгадывая её, команда получает указание, в какую локацию ехать дальше, переезжает туда, снова отгадывает загадку, переезжает на новое место и так далее. Игра заканчивается, когда команда достигает одной из локаций, отмеченных организаторами как финальные.*

Организаторы очередной игры составили план перемещения по локациям в виде списка чисел k_i. Число k_i указывает, что из i-й локации нужно следовать в локацию с номером k_i. Если i-я локация финальная, то $k_i = -1$.

Сейчас они хотят проверить, завершит ли за разумное время свои поездки команда, приехавшая в начале игры на первую локацию.

Решение задачи несложно идейно, но требует определённой технической подготовки. Программный комитет оценивает эту задачу как среднюю.

Если формализовать постановку, то имеется следующая задача. Задан ориентированный граф, из каждой вершины которого выходит либо одна дуга, либо ни одной. Те вершины, из которых не выходит дуга, являются финальными. Требуется определить, попадём ли мы, двигаясь из первой вершины, на цикл или же достигнем одной из финальных вершин. Решение состоит в прямом моделировании: вначале устанавливаем указатель на первую вершину и отмечаем её как посещённую. В дальнейшем передвигаем указатель в соответствии с информацией о выходящих дугах, помечая посещённые вершины и считая количество переходов. Если в какой-то момент попадаем на финальную вершину, ответ положительный. Если же попадаем на вершину, помеченную как посещённая, то следует вывести `-1`.

Идеи тестов:

1. Первая локация является конечной.
2. Длина искомого пути равна 1.

3–4. Для достижения конечной локации нужно пройти все остальные.

5. Для достижения конечной локации нужно пройти все остальные. Максимальное количество вершин.

6–19. Случайные тесты с существующим путём.

20–23. Случайные тесты с несуществующим путём.

24. Тест с петлёй.

25. Тест с петлёй в первой локации.

11.4. «Магнитное число». *Родители Пети Торопыжкина купили магниты на холодильник, имеющие форму цифр: a_0 нулей, a_1 единиц, ..., a_9 девяток. Когда мама сказала, что пора идти спать, Петя ответил, что хочет сначала составить из магнитов минимально возможное целое положительное число, кратное 15, так, чтобы все магниты были задействованы. Напишите программу, которая по набору магнитов у Пети найдёт это число. Ведущие нули в записи числа не допускаются.*

В целом, решение данной задачи несложно, однако требуется аккуратная реализация большого числа вариантов. Поэтому программный комитет оценивает сложность этой задачи как «выше средней».

Сначала проверяем, можно ли вообще составить число, кратное 15: в наборе должен быть хотя бы один ноль или одна пятёрка и сумма всех цифр должна быть кратна 3. Исключение из правила: набор состоит только из нулей.

Теперь нужно научиться выкладывать наименьшее число, кратное 15. Для этого в последний разряд (разряд единиц) ставим ноль или пятёрку. Заметим, что при наличии в наборе и пятёрки, и нуля, не всегда нужно ставить в разряд единиц пятёрку. Например, в случае набора цифр 0, 5, 7, минимально возможное число, кратное 15 — это 570. В нём пятёрка стоит не в разряде единиц. Выбор можно осуществить, попробовав оба варианта или воспользовавшись следующим правилом: если кроме этой пятёрки, нет ненулевых цифр, не превосходящих 5, то в разряд единиц надо ставить нуль; иначе — эту пятёрку.

Затем в старший разряд ставим минимальную ненулевую цифру из оставшихся. Остальные цифры размещаем в порядке возрастания между старшим и младшим разрядами.

Идеи тестов:

1. Одна цифра ноль.

2–4. Сумма цифр числа не делится на 3.

5–6. Сумма цифр делится на 3, но нет ни 0, ни 5.

7–8. Число существует, в наборе нет нулей.

9–10. Число существует, в наборе нет пятёрок.

11. Набор состоит из большого числа нулей.
12. В наборе одна пятёрка, нет цифр 1–4.
13. В наборе две пятёрки, нет цифр 1–4.

14–17. Случайные тесты.

18–20. Максимальные тесты

11.5. «Гуляя по столице». *После того, как Петя Торопыжкин отдохнул в деревне у бабушки, он поехал с родителями в Москву. Благодаря стараниям мэра, столица обзавелась большим количеством кольцевых дорог. Эти дороги представляют собой окружности с центром в начале координат и радиусами, выражаемыми всевозможными целыми числами от 1 до* 100 000*.*

Родители Пети наметили культурную программу — осмотр n достопримечательностей, расположенных в точках с координатами (x_i, y_i). Они хотят добраться к первой достопримечательности на такси и затем перемещаться от одной достопримечательности к другой пешком по прямой. Им интересно, пересекает ли очередной отрезок их пути хотя бы одну кольцевую дорогу (касание также считается пересечением). Помогите родителям Пети — напишите программу, которая ответит на этот вопрос.

Программный комитет оценивает эту задачу как очень сложную, поскольку она относится к непростой области вычислительной геометрии, а кроме того привлекает идеологически непростую процедуру пересечения отрезка с окружностью.

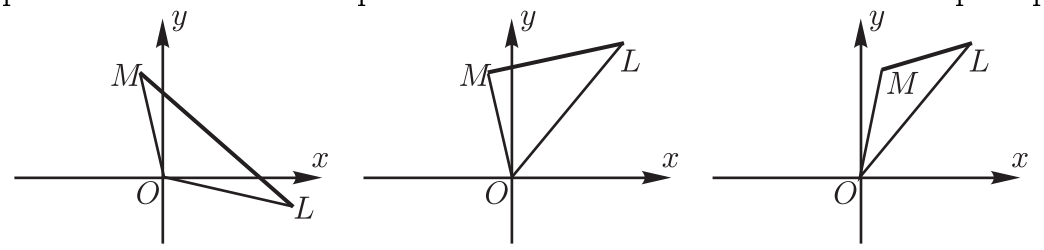
Заметим, что каждый факт пересечения или непересечения отрезка и окружности имеет результатом неразвёрнутый ответ. Если тест подразумевает проверку единичного пересечения, то жюри рискует столкнуться «наивными» программами, которые не решают задачу, а игнорируют входные данные и всегда отвечают `SAFE`, или всегда отвечают `RISKY`, или отвечают случайным образом. Для борьбы с подобными «решениями» применяется так называемый «мультитест»: в рамках каждого теста требуется ответить на несколько вопросов, и тест считается пройденным только если ответы на все вопросы правильные. Иногда мультитест вводится искусственно, иногда (как в рассматриваемой задаче) — естественно.

Другое замечание состоит в том, что можно отказаться от приближённых вычислений с плавающей точкой и перейти к вычислениям в целых числах, если умножить все координаты и радиусы на 100. Правда, в этом случае для работы потребуется тип int64.

Рассмотрим процедуру пересечения отрезка и окружности. Всякий раз, когда речь идёт о сравнении каких-либо длин, подразумевается сравнение их квадратов, которые есть целые числа (в случае целочисленности координат концов отрезков и радиусов).

Отрезок пересекается с окружностью с центром в начале координат тогда и только тогда, когда минимальное расстояние от начала координат до точек отрезка меньше или равно радиуса, а максимальное — больше или равно. Макси-

мальное расстояние ищется как максимум расстояний от начала координат до концов отрезка. Минимальное расстояние найти сложнее. Рассмотрим рисунок:



Из рисунка видно, что когда один из углов $\angle M$ и $\angle L$ треугольника OML прямой или тупой, то минимум расстояния достигается на одном из концов. Но когда оба эти угла острые, то минимум расстояния достигается на внутренней точке отрезка.

Острота, прямота или тупизна угла треугольника устанавливается через выражение, участвующее в теореме Пифагора. Например, $\angle M$ острый, если $OM^2 + ML^2 > OL^2$, прямой, если $OM^2 + ML^2 = OL^2$, и тупой, если $OM^2 + ML^2 < OL^2$. Проверку надо проводить дважды, соответственно меняя используемые отрезки, чтобы установить тип каждого из углов $\angle M$ и $\angle L$.

Если минимальное расстояние от точки $S(x_0, y_0)$ до отрезка достигается на внутренней точке отрезка, то оно равно расстоянию от этой точки до прямой l, содержащей отрезок. А это расстояние может быть найдено по формуле

$$d(S, l) = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}},$$

где $Ax + By + C = 0$ — уравнение прямой l. Поскольку измеряется расстояние от начала координат — точки $O(0, 0)$, то

$$d(O, l) = \frac{|C|}{\sqrt{A^2 + B^2}}, \text{ или } d^2(O, l) = \frac{C^2}{A^2 + B^2}.$$

Чтобы найти уравнение прямой, проходящей через несовпадающие точки с координатами (x_1, y_1) и (x_2, y_2), достаточно взять $A = y_1 - y_2$, $B = x_2 - x_1$, $C = -(Ax_1 + Bx_2)$.

Таким образом, определена процедура поиска расстояния от точки до отрезка, являющаяся центральной в рассматриваемом алгоритме. Повторимся, что возможно использование этой процедуры как в приближённом виде с применением вычислений в числах с плавающей точкой, так и в точных целочисленных построения, использующих длинный целый тип int64.

Процедура проверки пересечения/касания отрезка с одной из окружностей из нашего семейства организуется следующим образом. Ищем минимальное и максимальное расстояние до отрезка от начала координат. В случае работы с вещественными числами, пересечение имеется, если эти расстояния имеют разные целые части. При работе с целыми числами проверка пересечения чуть более сложна: нужно проверить, что квадраты расстояний принадлежат различным промежуткам вида $(n^2 \cdot 10^4, (n+1)^2 \cdot 10^4)$. Если наименьшее расстояние целое (при работе

с вещественными числами) или квадрат его равен $n^2 \cdot 10^4$ при каком-то натуральном n (при работе с целыми числами), то имеет место касание. В противном случае нет ни пересечения, ни касания.

Идеи тестов:

1. Один отрезок, пересечения есть с несколькими окружностями.
2. Два отрезка, один пересекается, другой нет.
3. Несколько отрезков, нет пересечения.
4. Несколько отрезков, есть пересечение.
5. Несколько отрезков, есть касание.
6. Несколько отрезков, треугольник прямоугольный, нет пересечения.
7. Несколько отрезков, треугольник прямоугольный, есть касание.
8. Есть точка, лежащая на окружности.
9. Все точки внутри самой маленькой окружности с радиусом 1.
10. Есть точка в начале координат.