

Задача A. Secret Code

Имя входного файла: `code.in`
Имя выходного файла: `code.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Разработчики системы NoPlan выяснили, что обработка обычных паролей занимает слишком много времени. Поэтому в системе используется более продвинутая система идентификации. Участник получает два натуральных числа s и k — сумму цифр и количество знаков в цифровом пароле, который ему требуется ввести. После чего он генерирует пароль (число, записанное без ведущих нулей) и сообщает системе. После неоднократных зависаний системы, по мнению автора, вызванных несанкционированным доступом, было решено усложнить систему защиты. Теперь от участника требуется ввести два числа: минимальное и максимальное из чисел, удовлетворяющих определяемым k и n требованиям.

Для тестирования системы аутентификации Вам поручено написать программу, по заданным n и k выводящую требуемые числа.

Формат входного файла

Первая строка ввода содержит число M — количество тестов. Далее, в последующих M строках, записаны по два разделённых пробелами натуральных числа — сумма цифр S и количество цифр $K < 10$. Гарантируется, что хотя бы одно решение всегда существует.

Формат выходного файла

Каждому тесту в выходном файле должна соответствовать строка с двумя числами, разделёнными пробелом — соответственно максимальным и минимальным числом с заданными количеством и суммой цифр.

Пример

<code>code.in</code>	<code>code.out</code>
3	100 100
1 3	200 101
2 3	3000 1002
3 4	

Задача В. Dance

Имя входного файла: `dance.in`
Имя выходного файла: `dance.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

На праздничном мероприятии, посвящённом исправлению юбилейного, 2008-го бага в системе NoPlan, выступил ансамбль современного танца с композицией «Отладка NoPlan-a». Танцоры двигаются цепочкой, затем выходят из цепочки по одному. Первый артист становится на левом краю сцены, второй уходит в конец исходной цепочки, третий — на левый край сцены (справа от первого), четвёртый — в конец исходной цепочки и т.д., пока все артисты не выстроятся на краю сцены. По замыслу постановщика, это должно символизировать отладку системы NoPlan.

При этом каждый из артистов держит табличку синего или красного цвета — как сказал постановщик, это символизирует обычные или фатальные ошибки. Помогите постановщику танца определить, каким должно быть исходное расположение артистов, если по завершении выступления на краю сцены они должны быть выстроены так, что их таблички чередуются по цвету (слева направо): синяя, красная, синяя, красная и т.д.?

Формат входного файла

Первая строка входных данных содержит число $N \leq 20$ — количество тестов. Далее следуют N строк, в каждой записано одно натуральное число K ($K \leq 1000$) — количество артистов.

Формат выходного файла

Для каждого теста в выходной файл выводится строка, содержащая цепочку символов, состоящую из заглавных букв B и R, соответствующих цветам табличек — синему и красному соответственно.

Пример

<code>dance.in</code>	<code>dance.out</code>
3	BR
2	BBR
3	BBRR
4	

Задача C. Digits

Имя входного файла: `digits.in`
Имя выходного файла: `digits.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В системе NoPlan для хранения больших чисел применяется следующая хеш-функция: каждому числу ставится в соответствие знакопеременная сумма его цифр, при этом цифра в наибольшем разряде всегда идёт в сумму со знаком «+», далее с каждой следующей цифрой знак меняется. Например, для числа 1234 значение функции равно $1 - 2 + 3 - 4$, то есть -2 .

В связи с недостаточным быстродействием системы было принято решение уменьшить все внутренние константы на один десятичный разряд, просто удалив одну цифру из их десятичной записи. При этом удаляемая цифра выбирается так, чтобы значение упомянутой выше знакопеременной суммы для получившегося числа было максимальным.

Вам поручено написать программу, по заданному числу (возможно, с ведущими нулями) определяющему, каким будет максимальное значение хеш-функции в случае, если из данного числа удалить какую-то одну цифру.

Формат входного файла

Первая строка входных данных содержит число $K \leq 20$ — количество тестов. Далее следуют K строк, в каждой из которых записано одно N -значное число ($1 \leq N \leq 50$) для каждого теста.

Формат выходного файла

:

Для каждого теста в выходной файл выводится одно число — значение наибольшей знакопеременной суммы для чисел, получающихся из заданного вычёркиванием одной цифры.

Пример

<code>digits.in</code>	<code>digits.out</code>
1 6274861	14

Задача D. Submits

Имя входного файла: `submits.in`
Имя выходного файла: `submits.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

К сожалению, экспериментальный запуск системы NoPlan оказался безуспешным, и организаторы соревнований приняли решение оценивать задачи аналогично тому, как оцениваются выступления участников в соревнованиях по художественной гимнастике: оценки решениям дают судьи вручную по методике В.А. Каймина.

Каждый submit описывается двумя числами: рейтинг участника (чем больше, тем участник сильнее), и номер сданной задачи (задачи в системе NoPlan нумеруются, начиная с самых простых). Судья сравнивает каждый submit со всеми, посланными ранее. Если в результате сравнения получается, что участник с большим рейтингом сдал более простую задачу, чем участник с меньшим рейтингом, судья снижает «балл за качество» на единицу, а также получает 1 тугрик компенсации за потерю веры в человечество.

Следует учитывать, что один submit может привести к снижению балла несколько раз. Один участник может сделать несколько submit-ов, так же, как и одна задача может быть решена несколькими участниками, в последнем случае это не может являться причиной снижения балла (то есть требуется строго более простая задача). Считается, что участники сдают все задачи с первой попытки и не отправляют сданные задачи повторно. Требуется подсчитать сумму в тугриках, которые оргкомитет должен будет выплатить судье по окончании соревнования.

Формат входного файла

Первая строка входных данных каждого теста содержит количество участников n ($0 < n \leq 250$), количество задач m ($0 < m \leq 250$), количество submit-ов p . Следующие p строк содержат по два числа, описывающие submit — номер участника в рейтинге, номер задачи. Далее начинается следующий тест. Окончание входных данных сигнализируется тестом, где $n = m = p = 0$, и этот тест обрабатывать не следует. Считать, что рейтинг участника остаётся неизменным всё время контеcта.

Формат выходного файла

Для каждого теста вывести одно число в строке — сумму в тугриках, которую необходимо выплатить судье по окончании соревнований.

Пример

submits.in	submits.out
3 3 4	3
3 1	0
1 2	
1 3	
3 2	
2 2 2	
1 1	
2 2	
0 0 0	

Задача E. Radio

Имя входного файла: `radio.in`
Имя выходного файла: `radio.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Перед проведением первого турнира с использованием системы NoPlan всем ответственным, сидящим в классах, выдали небольшие рации. На рации есть кнопка «Висим: НоПлан кончился!», подающая особый сигнал, а также ретранслятор, работающий так: в случае, если с какой-то из раций, находящихся в зоне приёма, поступает вышеупомянутый сигнал, то рация работает как ретранслятор, повторяя этот сигнал. Все рации однотипны, так что зона приёма-передачи для них одинакова и представляет круг радиуса R .

Требование организаторов заключается в том, чтобы во время турнира, проходящего на одном этаже, сигнал о зависании системы, поданный с произвольной рации, был принят всеми остальными рациями. При каком минимальном радиусе зоны приёма-передачи рации это возможно?

Формат входного файла

В первой строке задано количество тестов $K \leq 20$. Каждый тест задаётся несколькими строками. В первой содержится целое число $1 < N < 1000$ — количество ответственных, снабжённых рациями, далее идут N пар вещественных чисел x_i и y_i , по модулю не превосходящих 10000 — координаты каждого из N ответственных. Считается, что ответственные во время турнира должны находиться на месте и не должны перемещаться.

Формат выходного файла

Для каждого теста выведите в отдельной строке минимальный радиус R , при котором требования организаторов будут выполнены.

Пример

radio.in	radio.out
2	2.24
4	2.83
0 0	
2 0	
0 2	
2 3	
3	
2 0	
0 2	
4 2	

Задача F. Alloc

Имя входного файла: `ship.in`
Имя выходного файла: `ship.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В системе NoPlan выделение динамической памяти реализовано по аналогии с тем, как участник игры в «морской бой» размещает на игровом поле свои корабли, с тем уточнением, что «корабли» могут быть только прямоугольниками любых размеров, не могут пересекаться и иметь общих граничных точек.

В данный момент система обрабатывает особо сложный запрос обновления страницы результатов, так что требуется выделить максимальный блок памяти. В описанной выше реализации это значит, что при k уже размещённых кораблях следует разместить максимальный (по количеству занимаемых клеток) корабль.

Формат входного файла

Первая строка входа содержит количество тестов. Первая строка каждого теста содержит число n ($1 \leq n \leq 100$) — количество клеток в игровом поле по вертикали. Вторая строка теста — число m ($1 \leq m \leq 100$) — количество клеток в игровом поле по горизонтали. Третья строка теста — число k ($1 \leq k \leq 10$) — число кораблей, которые уже размещены. Следующие k строк теста содержат координаты k размещённых кораблей — 4 числа в каждой строке. 1-е и 2-е число — вертикальная и горизонтальная координаты левой верхней угловой клетки корабля, 3-е и 4-е число — вертикальная и горизонтальная координаты правой нижней угловой клетки корабля. Клетки поля нумеруются сверху вниз (от 1 до n) и слева направо (от 1 до m).

Формат выходного файла

Для каждого теста в выходной файл выводится количество клеток в последнем корабле.

Пример

ship.in	ship.out
1	21
8	
7	
3	
1 1 2 2	
3 5 3 7	
4 2 4 3	