

Задача А. Монеты

Имя входного файла: coins.in
Имя выходного файла: coins.out
Ограничение по памяти: 64 мегабайта

— Мистер Холмс, к вам какой-то офицер, — сообщила миссис Хадсон.

— Пусть войдёт, — ответил Холмс.

Хотя посетитель был в штатском, отставной военный в нём чувствовался сразу. Сэр Моррис служил в колониальных войсках в Африке, потом вышел в отставку. Приличное наследство плюс некоторый вывезенный из Африки капитал позволяли полковнику жить на широкую ногу.

— И как ваши успехи в игре, полковник? — спросил Холмс.

— А вы откуда... — начал Моррис.

— О, это элементарно. То, как Вы достали визитку, показывает, что Вы привыкли иметь дело с игральными картами.

— Понятно. Что же, я рад, что рассказы о Ваших способностях не являются слухами. Собственно, я пришёл, чтобы Вы помогли мне выплатить долг.

Оказалось, что в одном из карточных клубов появилась новая игра, для которой не требуются карты — только монета в один шиллинг. Вначале каждый из двух игроков выбирает последовательность «орлов» или «решек» произвольной длины и делает ставки. Затем игроки бросают монету, пока в последовательности результатов не встретится одна из двух выбранных последовательностей. Игрок, чья последовательность встретилась, выигрывает. В клубе также существовало правило выбирать примерно равновероятные последовательности, но реально оно держалось «на честном слове».

Наиболее успешным игроком в клубе был некий нотариус Джон Доу. Моррис проиграл ему некоторую сумму, но когда на следующий день пришёл, чтобы её отдать, Доу исчез. В его конторе тоже не имели понятия, куда он делся.

— Я привык отдавать свои долги, это дело чести! — завершил рассказ Моррис.

— Полковник, Вы можете привести примеры последовательностей, на которые ставили Вы и на которые ставил мистер Доу?

— Конечно, — полковник продиктовал несколько последовательностей.

— Мне кажется, что Вас обыграли с нарушением правил: последовательность, названная Доу, имеет куда большие шансы на выигрыш, чем это разрешено обычаями вашего клуба. А раз так, то и долга нет.

Полковник рассыпался в благодарностях, выдал Холмсу щедрый гонорар и ушёл, пообещав сообщить новость правлению клуба.

— А всё-таки... мне кажется, что с исчезновением Доу что-то нечисто. Вряд ли он бы просто так оставил такой надёжный источник дохода, как простак из этого клуба, — задумчиво заметил Холмс после ухода полковника. — Посмотрите, с какой вероятностью выигрывала его ставка!

Формат входного файла

Входной файл содержит несколько тестов. Каждый тест расположен на одной строке и содержит две непустые последовательности нулей и единиц, разделённые пробелами: сначала выигрывающая последовательность A для первого игрока (Доу), затем выигрывающая последовательность B для второго игрока (полковника Морриса). Длина каждой последовательности не превышает 50. После последнего теста вход содержит строку с единственным словом END. Гарантируется, что ни в одном тесте ни одна последовательность A или B не

содержит другую последовательность (например, 0100 и 10), что вероятность выигрыша каждого игрока положительна и что ничья невозможна.

Формат выходного файла

Для каждого теста выводится строка с номером теста (в формате, указанном в примере к задаче; при этом все пробелы одиночные) и вероятность победы первого игрока с 6 цифрами после запятой.

Пример

coins.in	coins.out
01 10	Case 1: 0.500000
00 1	Case 2: 0.250000
001 011	Case 3: 0.666667
END	

Задача В. Карточный клуб

Имя входного файла: club.in
Имя выходного файла: club.out
Ограничение по памяти: 64 мегабайта

— Что вы собираетесь предпринять, Холмс? — спросил я.

— Нанести визит в карточный клуб, затем в контору Джона Доу, а затем, если понадобится — к хозяину дома, в котором квартировал нотариус. Как минимум где-то, я предполагаю, должна отыскаться зацепка.

В карточном клубе про Доу ничего не было слышно уже неделю. Завсегдатаи рассказывали, что необычные игры в клуб приносил именно он. В последний раз его видели с блокнотом, в котором он делал какие-то записи, явно не относящиеся к игре, при этом впервые за долгое время Доу почти не играл «по-крупному».

— Кстати, если хотите посмотреть — вот одна из принесённых им игр, — предложил хозяин клуба.

Мы прошли в зал. Около стола с фишками сидел крупье и один из посетителей.

Игра шла по следующим правилам. Сначала у игрока есть N столбиков по p_i фишек в i -м столбике. Нижняя фишка имеет зелёный цвет, остальные — или красный, или синий, при этом в одном столбике не могут одновременно присутствовать красные и синие фишки.

Игрок может или добавить L красных фишек к столбику, в котором нет ни одной синей фишки, или изъять L синих фишек из столбика, в котором нет ни одной красной фишки и не менее L синих, или — в случае, когда в столбике ровно $K \leq L - 1$ фишек синего цвета — заменить K синих фишек на $L - K$ красных. Также в случае, когда два столбика идентичны, игрок может снять со стола оба эти столбика (включая и зелёные фишки). Задача — оставить на столе как можно меньше зелёных фишек.

— Заметьте, Ватсон, это ещё одна игра, которая при нужном навыке элементарно просчитывается в уме. Доу имел здесь неплохой доход: по начальному раскладу фишек вполне можно определить наилучший результат. Кстати, попробуйте это сделать и Вы.

Формат входного файла

В первой строке входного файла содержится количество тестов. Далее для каждого теста отводится по две строки. Первая строка содержит натуральные числа L и N ($L \leq 500000$, $N \leq 10^6$). Во второй строке записаны N целых чисел p_1, p_2, \dots, p_N (в пределах от -2147483648 до 2147483647), где p_i — количество красных (если число положительное) или синих (если число отрицательное) фишек в i -м столбике.

Формат выходного файла

Для каждого теста в отдельной строке должно располагаться единственное значение — минимальное количество зелёных фишек, которые могут остаться в данной игре.

Пример

club.in	club.out
2	1
6 5	0
35 16 11 10 22	
1 8	
2 -4 2 6 0 4 -3 7	

Задача С. Премии

Имя входного файла: `bonus.in`
Имя выходного файла: `bonus.out`
Ограничение по памяти: 64 мегабайта

В фирме Джона Доу существовала следующая традиция. В конце каждой недели Джон выплачивал своим сотрудникам премии в соответствии с их стажем работы на Джона в днях: из двух сотрудников с различным стажем тот, кто обладает большим стажем, должен получить как минимум на K фунтов больше, чем тот, кто обладает меньшим стажем. Всего в фирме Джона N работников (включая самого Джона, который сам себе премию не выплачивает). По традиции, все премии исчисляются целым количеством фунтов, каждый работник получает как минимум 1 фунт премиальных, при этом общий фонд премиальных не должен превышать m фунтов.

Холмс выяснил, что помощник Джона по кадрам был крайне педантичен, так что принять на работу более одного человека в день не мог. Тем самым стало понятно, что у всех работников фирмы стаж различный.

Вариантов распределения премий было достаточно много, и перед принятием решения Джон заносил все варианты в блокноты из P страниц, по одному варианту на страницу (начинал с пустого блокнота, когда очередной блокнот заканчивался, переходил на следующий). Все блокноты, кроме последнего, были обнаружены в офисе. И судя по всему, именно этот, последний, блокнот и содержал на оставшихся страницах текст загадочного послания...

— Знаете, Ватсон, если бы мы подсчитали, сколькими способами Доу мог распределить премии, мы бы вычислили, сколько страниц занимает интересующее нас послание.

— Но, Холмс, это число может быть слишком велико!

— Возможно. Но количество оставшихся после расчётов страниц, как несложно заметить, в любом случае не превышает $P - 1$. Так что за дело!

Формат входного файла

Первая строка содержит целое число $1 \leq T \leq 200$ — количество тестов. Каждая из следующих T строк состоит из четырех целых чисел, разделённых пробелами — $2 \leq N \leq 1000$, $1 \leq m \leq 1000$, $1 \leq K \leq 1000$, $1 \leq P \leq 10^9$, описывающих один тестовый случай.

Формат выходного файла

Для каждого тестового случая в отдельной строке выведите одно целое число $0 \leq R \leq P - 1$ — количество оставшихся после расчётов премий страниц в последнем блокноте Джона Доу.

Пример

<code>bonus.in</code>	<code>bonus.out</code>
1 3 5 1 7	4

Задача D. Нумерация

Имя входного файла: `street.in`
Имя выходного файла: `street.out`
Ограничение по памяти: 64 мегабайта

В поисках дома, где жил Доу, мы оказались в одном из старых районов Лондона. Пытаясь сориентироваться в номерах домов, я окончательно запутался.

— Что за странная нумерация, Холмс? Она не подчиняется никакой логике! Более того, числа даже не идут подряд!

— Элементарно, Ватсон. В этом районе Лондона раньше жили каббалисты, народ весьма умелый в счёте. И чтобы оградить себя от желающих с ними расправиться, они стали нумеровать дома хитрым образом. Для заданного множества K простых чисел $S = \{p_1, p_2, \dots, p_K\}$ они брали множество Q всех чисел, для которых множество простых делителей является подмножеством S . Это множество Q содержит, среди других чисел, например, p_1 , p_1p_2 , p_1p_1 и $p_1p_2p_3$. Затем они упорядочивали элементы этого множества в порядке возрастания и нумеровали дома. Так что здесь номеру N по обычной нумерации соответствует N -е число в Q . А так как помощник Доу не знал этой тонкости, то сейчас нам предстоит по заданному N вычислить нужный номер дома.

Формат входного файла

Первая строка входа содержит количество тестов T . Первая строка каждого теста содержит два разделённых пробелами целых числа K ($1 \leq K \leq 100$) и N ($1 \leq N \leq 100000$). Вторая строка содержит K разделённых пробелами целых положительных чисел, из которых состоит множество S .

Формат выходного файла

Для каждого теста необходимо вывести в отдельной строке N -е число множества Q , построенного в соответствии с условиями задачи.

Пример

<code>street.in</code>	<code>street.out</code>
2	27
4 19	45
2 3 5 7	
2 7	
3 5	

Задача E. Лестница

Имя входного файла: `stairs.in`
Имя выходного файла: `stairs.out`
Ограничение по памяти: 64 мегабайта

Найдя дом, в котором Доу квартировал, мы долго поднимались и спускались по лестницам различной длины, прежде чем нашли хозяина. Он оказался весьма словоохотливым. По его словам, нотариус был достаточно замкнут и имел странную привычку иногда задаваться самыми неожиданными вопросами.

— Один раз, мистер, я подхожу к лестнице и вижу, как Джон что-то считает. На мой вопрос он ответил, что ему интересно, каково общее количество возможных путей для поднятия по лестнице, если человек может подниматься по лестнице, делая шаг на следующую ступеньку или перепрыгивая через одну или две последовательные ступеньки, — разглагольствовал хозяин.

Мы зашли в кабинет Джона. Судя по всему, Доу собирался в суматохе, часть его бумаг лежала на столе. Но Холмса привлёк листок бумаги, выброшенный в мусорную корзину. На нём было несколько чисел. При этом числа, большие или равные миллиону, были записаны не полностью — три первых цифры, многоточие, три последних. Одно из чисел на листке было обведено.

— Вот-вот, на этом листке он и считал, — раздался голос хозяина.

— Интересно. А не кажется ли вам, Ватсон, что таким способом Доу обозначал лестницы — лестнице из N ступеней соответствует количество возможных способов подняться по ней. И что обведённое число указывает на лестницу, под которой, например, устроен тайник? Я предлагаю построить аналогичные записи для всех лестниц в доме — возможно, для какой-то из них оно совпадёт с обведённой на листке...

Формат входного файла

Входной файл содержит несколько тестовых примеров. Первая строка содержит одно целое число $T < 30$, которое является количеством примеров. Затем следуют T строк, каждая из которых описывает один тестовый пример и содержит целое число $N (1 \leq N \leq 2 * 10^9)$, равное количеству ступенек на лестнице, для которой нужно подсчитать количество способов подъёма.

Формат выходного файла

Выходной файл должен содержать T строк — по одной для каждой совокупности тестовых данных. Каждая строка должна содержать одно число — количество различных путей, которыми может воспользоваться человек для поднятия по лестнице, описанной в соответствующем тесте. Если число содержит более 6 десятичных знаков, то данная строка должна содержать три первых десятичных знака, затем три точки и три последних десятичных знака числа.

Пример

<code>stairs.in</code>	<code>stairs.out</code>
3	1
1	2
2	138...537
24	

Задача F. Словарь

Имя входного файла: `grammar.in`
Имя выходного файла: `grammar.out`
Ограничение по памяти: 64 мегабайта

Найденный под лестницей тайник содержал исчезнувший блокнот Джона Доу, а также карту обширного поместья баронетов Флойд, на которую был нанесён знаменитый лабиринт.

Записи в конце блокнота, после перечисления всех вариантов премий, были написаны какими-то непонятными словами.

— Хм... «Первая Троица на самом дне в круге девятом...», дальше надо смотреть по статье...

— Что это за тайнопись, Холмс?

— Это не тайнопись, а язык одного из африканских племён. Недавно в заметках Географического Общества сэра Абрахам Флойд описал своё годичное пребывание в одном африканском племени, в языке которого местная мифология переплелась со следами рассказов миссионеров. Скорее всего, это фрагмент завещания Флойда, который был большим оригиналом.

— Кажется, мне понятно, почему исчез Доу — как нотариус, он узнал про завещание и понял, что обладание этой самой информацией сулит ему немалую выгоду, — я, кажется, начал что-то понимать.

— Именно. Кстати, в той же статье приведено формальное описание грамматики этого языка. И для того, чтобы убедиться, что мы на верном пути, нам просто надо проверить, принадлежит ли каждое слово этого текста данному языку.

Так как племя, язык которого использовал Флойд, находится на невысоком уровне развития, то язык этого племени можно задать грамматикой очень простого вида. Каждое правило грамматики $G = (N, \Sigma, P, S)$, (N — конечное множество нетерминальных символов, Σ — не пересекающееся с N конечное множество терминальных символов, P — конечное множество правил, S — выделенный из N начальный символ) имеет вид либо $W \rightarrow XY$, либо $T \rightarrow t$, где W, X, Y, T — символы из множества нетерминалов N , а t — символ из множества терминальных символов Σ .

Формат входного файла

Первая строка входного файла содержит количество тестовых случаев T ($1 \leq T \leq 20$). Каждый тестовый случай состоит из нескольких строк. Первая строка каждого тестового случая содержит все нетерминальные символы некоторой грамматики G , за исключением начального символа 'S'. Символ 'S' служит начальным для грамматик из всех тестовых случаев и не включается в строку нетерминальных символов. Нетерминальным символом может быть любая заглавная буква латинского алфавита. Строка, задающая нетерминальные символы, не пустая и не содержит пробелов. Следующая строка содержит все терминальные символы грамматики G . Терминальным символом может быть любая маленькая буква латинского алфавита. Строка, задающая терминальные символы, не пустая и не содержит пробелов. Следующая строка содержит единственное целое положительное число K ($1 \leq K \leq 20$) — количество продукций грамматики G . Далее следуют K строк, каждая из которых содержит одно правило. Каждое правило представляет собой строку, не содержащую пробелов, вида $W \rightarrow XY$ или $T \rightarrow t$, где W, X, Y, T — символы из множества нетерминальных символов, t — символ из множества терминальных символов. Следующая строка содержит единственное целое положительное число M ($1 \leq M \leq 20$) — количество строк терминальных символов, подлежащих проверке на принадлежность языку, порожаемому грамматикой G . Далее сле-

дуют M строк. Каждая строка состоит только из терминальных символов грамматики G и имеет длину не более 50 символов.

Формат выходного файла

Для каждой строки терминальных символов выведите в стандартный поток YES, если эта строка принадлежит языку, порождаемому соответствующей грамматикой, или NO в противном случае.

Пример

grammar.in	grammar.out
2	YES
AB	NO
ab	NO
4	YES
S->AB	YES
A->BA	YES
A->a	YES
B->b	YES
5	NO
bbbbab	NO
bbbbbbbaab	
baaaaaaab	
bab	
ab	
A	
abc	
5	
S->AA	
S->a	
S->b	
S->c	
A->a	
5	
a	
b	
c	
abc	
aaaaaaaaabbbbbbbccccc	

Задача G. Лабиринт

Имя входного файла: `labyrinth.in`
Имя выходного файла: `labyrinth.out`
Ограничение по памяти: 64 мегабайта

— Итак, что Вы знаете о том, как выглядел лабиринт в вашем поместье? — спросил Холмс.

— Практически ничего. Сохранилось только несколько гравюр, на которых изображена часть лабиринта, да бумаги с отдельными распоряжениями сэра Артура Флойда, при котором и был устроен лабиринт. Достоверно известно, что лабиринт состоял из фигур трёх видов: круги, треугольники, эллипсы. Фигуры были расположены причудливым образом и территория лабиринта была поделена на несколько областей, — ответил сэр Мэттью Флойд.

— О да, про это я слышал, и, зная о фамильной склонности Флойдов к различным головоломкам, даже изучил некоторую литературу. Оказалось, что деление плоскости одинаковыми формами хорошо описано в литературе, — тут Холмс нарисовал несколько картинок.

Заметим, что четыре круга делят плоскость максимум на 14 областей, четыре эллипса делят плоскость максимум на 26 областей, а три треугольника делят плоскость максимум на 20 областей. Есть и более общая задача. Она состоит в выяснении максимального количества областей, которое можно получить при пересечении m форм одного типа. Например общая формула для кругов: $m^2 - m + 2$. Что характерно, Ватсон, всё это я прочитал в работе некоего Мориарти.

— Того самого...?

— Да, причём это была одна из последних его работ перед тем, как он окончательно решил сменить поле деятельности. Интересные совпадения, не так ли? Так вот, продолжим. Мориарти показал, что для гибридной ситуации (пересечение более чем одного вида форм) максимальное возможное количество возникающих областей также вычисляется не очень сложно. Например, при пересечении одного эллипса и одного треугольника образуется восемь областей. Но нам надо решить обратную задачу. Из распоряжений сэра Артура Флойда мы знаем, что конструкция лабиринта была максимальной для задействованных в ней фигур. Также мы знаем, сколько «комнат» было в лабиринте, то есть на сколько областей разбивалась плоскость. А надо нам «всего лишь» понять, сколько эллипсов, кругов и треугольников составляли лабиринт. И если вариантов несколько — перебрать все варианты.

Формат входного файла

Входной файл содержит менее 300 строк. Каждая строка содержит 32-разрядное целое беззнаковое число N , которое является максимальным количеством областей, созданных m эллипсами, n кругами и p треугольниками. Вход заканчивается строкой, которая содержит -1 . Эту строку не нужно обрабатывать. Все входные числа (кроме -1 в последней строке) — положительные.

Формат выходного файла

Для каждой входной строки вам придется вывести две или больше выходных строк. Описание выхода для каждой строки представлено ниже.

Первая строка имеет форму `Case L:` (где L — порядковый номер входного теста). Каждая из следующих строк содержит три целых числа: возможные значения m , n и p , для которых создано максимальное количество областей N . При наличии более, чем одного решения, их нужно отсортировать в порядке возрастания m , а затем в порядке возрастания n . Если правильное решение отсутствует, вместо этого следует вывести `Impossible`. Отметим, что для правильного решения $0 \leq m \leq 100$, $0 \leq n < 20000$ и $0 \leq p < 100$.

Пример

labyrinth.in	labyrinth.out
20 10 -1	Case 1: 0 0 3 0 1 2 1 0 2 1 3 0 Case 2: Impossible.

Задача Н. Дождь

Имя входного файла: `rain.in`
Имя выходного файла: `rain.out`
Ограничение по памяти: 64 мегабайта

— Что мы делаем в этом болоте?

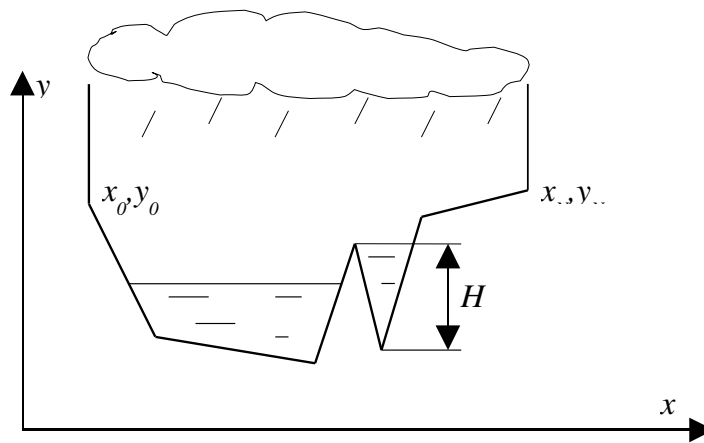
— Пытаемся обогнать Джона Доу, Ватсон. Если мои расчёты правильны, то он не появится здесь ранее, чем через четыре часа.

— То есть Вам известно, что Джон Доу должен быть здесь?

— Естественно. Он придёт сюда за кладом Флойдов. Который, согласно расшифрованным нами записям, находится «на самом дне» в точке пересечения девятого круга и первого треугольника старого лабиринта Флойдов.

— То есть «самое дно» — это указание на конкретное место?

— Естественно, — заметил Холмс. — Вот карта, показывающая профиль данной местности. Точки на ней задаются двумя измерениями: длиной x и высотой y . На этой карте рельеф местности представлен ломаной $(x_0, y_0), \dots, (x_N, y_N)$, состоящей из N отрезков. При этом $x_0 < x_1 < \dots < x_N$ и $y_i \neq y_j$, для любых $i \neq j$. Слева в точке x_0 и справа в точке x_N рельеф бывшего лабиринта, как мы видим, ограничен оградой поместья Флойдов.



Если бы рельеф был горизонтальным, то после дождя вся местность покрылась бы слоем воды глубиной H . Но поскольку рельеф — это ломаная, то вода стекает в углубления и образует водоемы. «Самое дно» — это там, где глубина максимальна. Так что сейчас нам требуется найти максимальную глубину в образовавшихся после дождя водоемах, — завершил свою мысль Холмс.

Формат входного файла

В первой строке входного файла записано число $T < 100$, далее записано T тестовых наборов, каждый из которых имеет следующую структуру: В первой строке каждого набора расположены два числа: целое число N ($1 \leq N \leq 100$) и вещественное число H ($0 \leq H \leq 1000$). В последующих $N + 1$ строках — по два целых числа x_i, y_i ($0 \leq i \leq N, 0 \leq x_i \leq 1000, 0 \leq y_i \leq 1000$). Числа в строках разделены пробелами.

Формат выходного файла

Выходной файл должен содержать T чисел, по одному числу (на отдельной строке) на каждый тестовый набор — искомую глубину H_{max} с точностью два знака после десятичной точки.

Пример

rain.in	rain.out
1	1.00
2 0.375	
0 1	
2 0	
4 2	

Задача I. Секрет

Имя входного файла: `secret.in`
Имя выходного файла: `secret.out`
Ограничение по памяти: 64 мегабайта

Когда найденный нами сундук был доставлен на второй этаж Флойд-хауза, Холмс тут же приступил к его осмотру.

— Посмотрите, Ватсон, — обратился он, вот доказательство того, что сундук принадлежал сэру Артуру Флойду. Сундук закрыт на кодовый замок, представляющий собой числовую головоломку. В нём n последовательных целых чисел от 1 до n расположены по кругу в некотором порядке. Числа с помощью хитрого механизма можно переставлять. Замок открывается, если на нём последовательно установить две такие расстановки, в которых сумма произведений всех соседних пар чисел соответственно является наименьшей и наибольшей.

Например, при $n = 6$ и расстановке $(1,3,6,5,2,4)$ (в порядке обхода по часовой стрелке) сумма произведений всех соседних пар равна $1x3 + 3x6 + 6x5 + 5x2 + 2x4 + 4x1 = 73$, но она не является ни наименьшей, ни наибольшей.

— Вы знаете, как найти эти расстановки?

— Да, но я не уверен в том, что не ошибусь. Так что, Ватсон, подсчитайте для заданного n наименьшую и наибольшую сумму, чтобы проконтролировать меня перед тем, как замок будет открыт...

Формат входного файла

Входной файл состоит из некоторого количества тестовых примеров, заданных по одному на строку. Каждая строка содержит одно целое число n ($2 < n \leq 10000$).

Формат выходного файла

Для каждого n вывести наименьшую и наибольшую возможную сумму произведений всех соседних пар.

Пример

<code>secret.in</code>	<code>secret.out</code>
3	11 11
6	58 82
100	171749 338153

Задача J. Засада

Имя входного файла: `hide.in`
Имя выходного файла: `hide.out`
Ограничение по памяти: 64 мегабайта

— Вы уверены, Лестрейд, что он от Вас не ускользнёт?

— Конечно, мистер Холмс! Смотрите, я всё учёл. В операции участвует S агентов. Каждый агент занимает свою позицию около входа в один из N домов, расположенных на левой стороне улицы. Чтобы не привлекать внимание, двум агентам занимать позицию около одного входа запрещено. А теперь — внимание. Чтобы уж точно не спугнуть Доу, я расставлю агентов так, чтобы минимальное расстояние между какими-нибудь двумя агентами — обозначим его за M — было как можно большим. А из всех подобных расстановок я выберу такую, при котором расстояние между двумя соседними агентами будет минимально возможным, естественно, при том, что оно будет не меньше M . Да, и ещё — при всём при этом я постараюсь разместить первого, если считать от начала улицы, агента как можно ближе к этому самому началу. И тогда Доу никуда не денется!

— Агенты уже расставлены?

— Ну... — смутился Лестрейд — пока что мы с Грегсоном пытаемся найти нужную расстановку. Кстати, Холмс, может быть Вы поможете нам в этом?

Формат входного файла

Первая строка входного файла содержит количество тестовых случаев T ($1 \leq T \leq 20$). Каждый тестовый случай состоит из нескольких строк. Первая строка каждого тестового случая содержит два натуральных числа, разделенных пробелом — количество $2 \leq N \leq 100000$ домов с левой стороны улицы, и количество агентов S ($2 \leq S \leq N$). Далее следуют N строк, содержащих координаты входных дверей p_1, p_2, \dots, p_N , около которых могут располагаться агенты так, что i -я строка содержит единственное целое число p_i ($0 \leq p_i \leq 10^9$). Координаты указаны в произвольном порядке. Все числа p_1, p_2, \dots, p_N различны.

Формат выходного файла

Для каждого тестового случая выведите ровно S строк. i -я строка должна содержать координату места, которое должен занять i -ый агент для того, чтобы минимальное расстояние между какими-либо двумя охотниками достигало максимально возможного значения M . При этом координаты должны идти в возрастающем порядке. Расстояние между соседними агентами должно быть минимально возможным, но не меньше величины M , а первый агент должен всегда располагаться в месте с наименьшей возможной координатой.

Пример

hide.in	hide.out
2	1
5 3	4
1	8
2	1
8	4
4	7
9	
8 3	
1	
2	
3	
4	
5	
6	
7	
8	