Задача A. Любимые числа

Имя входного файла:	`number.in`
Имя выходного файла:	`number.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

— Какое у вас любимое число, мейстер Эйемон? — спросил Сэм.

— Конечно, 2068965517241379310344827586, — ответил мейстер.

— Почему?

— Это первое число, такое, что оно заканчивается на цифру 6, и если перенести эту цифру в начало, оно увеличится ровно в 3 раза! Нахождение всех таких чисел было задачей на моем вступительном экзамене в академию. Если ты хочешь поступить в академию, ты должен быть достаточно подготовлен. Сумеешь ли ты найти k-е по порядку число с таким свойством? Дабы не утруждать себя, посчитай только остаток от деления этого числа на $10^9 + 7$.

Формат входного файла

В первой и единственной строке входного файла записано одно целое число k ($1 \le k \le 10^9$) — порядковый номер искомого числа, которое Вам нужно найти.

Формат выходного файла

В единственной строке выходного файла выведите остаток от деления искомого числа на $10^9 + 7$.

Пример

number.in	number.out
1	34482049

Задача B. Баскетбол

Имя входного файла:	`basketball.in`
Имя выходного файла:	`basketball.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В Вестеросе существует игра, сильно похожая на баскетбол. Тирион Ланнистер не может в неё играть, зато он любит следить её ходом.

В вестероский баскетбол играют n человек. Есть два кольца, каждый игрок может забить мяч в любое из них и получить за это одно, два или три очка. Тирион ведет записи, в которых он фиксирует, как изменился счет, и кто забил мяч. Счёт представляет собой два числа — сколько очков было забито в первое и во второе кольцо соответсвенно. Для красоты он разделяет эти два числа двоеточием. Таким образом одна строчка в записях Тириона может выглядеть так: «30:41 Lannister».

После игры Тирион захотел узнать, кто же набрал больше всех очков. Но, к сожалению, обязанности мастера над монетой отнимают у него много времени. Ланнистеры всегда платят долги, поэтому у Вас есть все основания, чтобы помочь ему.

Формат входного файла

В первой строке входного файла задано число n ($1 \le n \le 50$) — количество игроков.

В следующих n строках записаны имена игроков s_i ($1 \le |s_i| \le 15$). Имена состоят только из больших и маленьких букв латинского алфавита.

В строке с номером $n + 2$ дано число m ($1 \le m \le 500$) — количество записей Тириона.

В следующих m строках даны записи в виде «a_i:b_i t_i» без кавычек, где a_i — очки, забитые в первое кольцо, b_i — во второе, t_i — имя игрока, сделавшего счёт таким. Тирион никогда не ошибается в подсчетах, поэтому гарантируется, что его записи верны.

Начальный счет, как и во всех играх, 0:0.

Формат выходного файла

В единственной строке выходного файла выведите имя игрока, который набрал больше всех очков, и количество очков, которые он набрал. При наличии нескольких ответов разрешается вывести любой.

Пример

basketball.in	basketball.out
3 Stark Barateon Lannister 4 3:0 Stark 3:2 Barateon 4:2 Lannister 7:2 Lannister	Lannister 4

Задача C. Набор в армию

Имя входного файла:	`army.in`
Имя выходного файла:	`army.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Роберт Баратеон понимает, что войны с Таргариенами не избежать. Но сейчас зима, они не будут наступать, пока не начнется лето. Лето наступит через год, то есть через 366 дней — не так уж и много. Поэтому Роберт хочет успеть собрать себе за это время армию.

У него есть n рыцарей, у рыцаря с номером i есть a_i солдат в подчинении. Роберт отдал приказ своим рыцарям собирать больше солдат, и вечером каждого дня ему приходит донесение, которое имеет следующий вид: «Рыцари с номерами с l по r нашли себе еще по одному солдату».

Также в любой момент времени Роберт может посмотреть на отряды солдат рыцарей с номерами с l по r и, исходя из этой информации, посчитать количество способов выбрать главнокомандующих у каждого из этих $r - l + 1$ отрядов солдат. Все солдаты в отрядах равноценно могут быть главнокомандующими, два способа считаются различными, если в них отличаются хотя бы два главнокомандующих. Число может получиться довольно большим, поэтому Роберт хочет знать только его остаток от деления на 1000003.

Король Семи Королевств не силен в математике, поэтому он попросил Вас помочь ему в решении этой задачи.

Формат входного файла

Первая строка входного файла содержит число n ($1 \le n \le 10^5$) — количество рыцарей у Роберта.

Вторая строка содержит n целых чисел a_i ($1 \le a_i \le 10^9$) — начальное количество солдат у i-го рыцаря.

В третьей строке входного файла дано число q ($1 \le q \le 10^5$) — количество запросов. В следующих q строках входного файла описаны запросы. Каждый запрос описывается тремя числами a, l, r ($0 \le a \le 1, 1 \le l \le r \le n$).

$a = 0$ означает увеличение количества солдат у всех рыцарей с номерами с l по r включительно на один.

$a = 1$ означает запрос на подсчет количества способов выбрать главнокомандующих у отрядов рыцарей с номерами с l по r включительно.

Гарантируется, что донесенение об увеличении количества солдат приносили Роберту на чаще одного раза в день, то есть не более 366 раз.

Формат выходного файла

На каждый запрос с $a = 1$ в выходной файл выведите ответ на запрос.

Пример

army.in	army.out
5 1 2 3 4 5 6 1 1 5 0 3 3 0 2 4 1 1 3 1 1 4 1 1 5	120 15 75 375

Задача D. Проблема

Имя входного файла:	`problem.in`
Имя выходного файла:	`problem.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

И вновь в Семи Королевствах проблемы! До Роберта Баратеона — короля государства — дошли слухи о появлении драконов. Он хочет как можно скорее оповестить об опасности все n городов, расположенных на территории государства. К несчастью, Джоффри — сын короля — перестрелял из рогатки почти всех почтовых голубей. Остался всего один. Теперь король хочет выбрать город, чтобы послать из него гонца, который оповестит все оставшиеся города, затратив на это минимальное время.

Но гонцы в Семи Королевствах чрезвычайно ленивые, они не утруждаются составлением маршрута обхода городов. Однако они придерживаются определенной стратегии:

Пускай гонец находится в городе i. И до этого он посетил множество городов M. Тогда следующим он посетит город j, ближайший к городу i, и при этом не принадлежащий множеству M. Если таких городов несколько, он посетит город с минимальным номером.

Города представлены как точки на плоскости с координатами x_i, y_i в декартовой системе координат. Расстояние и время, затрачиваемое гонцами на передвижение между городами, определяется как квадрат евклидового расстояния между точками, которые соответствуют городам.

Помогите Роберту Баратеону выбрать город, гонец из которого посетит все остальные города за минимальное время.

Формат входного файла

В первой строке входного файла дано число n — число городов ($2 \le n \le 300$).

В следующих n строках дано описание городов — координаты x_i, y_i ($-300 \le x_i, y_i \le 300$).

Формат выходного файла

В единственной строке выходного файла выведите минимальное время, за которое все города могут быть оповещены.

Пример

problem.in	problem.out
7 2 4 4 3 7 2 7 5 9 7 5 8 2 7	51

Задача E. Короли

Имя входного файла:	`kings.in`
Имя выходного файла:	`kings.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Армия Джоффри состоит из n батальонов по a воинов, армия Робба — из m батальонов по b воинов, армия Станниса — из k батальонов по c воинов.

Помогите аналитикам Цитадели выяснить, чья армия сильнее. Силой армии считается число воинов в ней.

Формат входного файла

В первой и единственной строке входного файла записаны 6 целых чисел: n, m, k, a, b, c ($1 \le n, m, k, a, b, c \le 10^3$) — число батальонов у Джоффри, Робба, Станниса, и число воинов в них соответственно.

Формат выходного файла

Если у короля с самой сильной армией есть x воинов, в единственной строке выходного файла выведите имена всех королей, у которых x воинов, через пробел в лексикографическом порядке.

Пример

kings.in	kings.out
`2 4 3 6 3 4`	`Joffrey Robb Stannis`
`2 3 3 6 3 4`	`Joffrey Stannis`

Задача F. Противостояние

Имя входного файла:	`opposition.in`
Имя выходного файла:	`opposition.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Неожиданно для всех между Волантисом и Пентосом разразилась война. Жители Волантиса, зная, что вот-вот будет первое наступление, решили усовершенствовать оборонную систему города. А именно, они решили установить противопехотные ежи.

Для того чтобы добиться максимального эффекта от баррикады, жители Волантиса решили составить план конструкции. Система ежей представлена на плане в виде непересекающихся отрезков на прямой.

Благодаря шпионам, этот план попал в руки пентосского военачальника. Узнав о том, что жители Волантиса не собираются сдаваться без боя, военачальник огорчился. К тому же он уже составил собственный план наступления: сформировал шеренги солдат, отметив их на своем плане системой непересекающихся отрезков на прямой, подобно тому, как это сделали люди из Волантиса. Но, взяв себя в руки, военачальник решил сдвинуть своих солдат так, чтобы после нанесения отрезков, соответствующих солдатам, на прямую с отрезками, соответствующими противопехотным ежам, суммарная длина пересечения отрезков была минимальна.

Однако, если военачальник сдвигает какой-либо из отрезков, все остальные отрезки сдвигаются на столько же в том же направлении. Помимо того, левые и правые границы всех отрезков должны быть не меньше заданного l и не больше r. Также сдвигать отрезки можно только на целое число.

Помогите пентосскому военачальнику, найдите длину минимального суммарного пересечения отрезков.

Формат входного файла

В первой строке входного файла дано два числа n и m — число отрезков с противопехотными ежами и число отрезков с солдатами соответственно ($1 \le n, m \le 300$).

В следующей строке даны два числа l и r — границы, описанные в условии ($-10^9 \le l, r \le 10^9$).

Далее следуют n строк с описанием системы противопехотных ежей в Волантисе. Противопехотный ёж описывается двумя числами a_i, b_i — отрезком на прямой с началом в точке a_i и концом в точке b_i ($l \le a_i \le b_i \le r$).

Далее следуют m строк с аналогичным описанием шеренг солдат Пентоса.

Все отрезки даны в порядке возрастания координат их левых концов. Для всех отрезков, описывающих шеренги солдат, координата начала $i+1$-го отрезка всегда больше координаты конца i-го отрезка. Аналогичное утверждение верно и про отрезки, описывающие систему противопехотных ежей.

Формат выходного файла

В единственной строке выходного файла выведите значение длины минимального суммарного пересечения отрезков при каком-то сдвиге отрезков, описывающих шеренги солдат. Сам сдвиг выводить не надо.

Пример

opposition.in	opposition.out
3 2 0 5 0 1 2 3 4 5 0 1 2 3	0
2 3 0 7 1 3 6 7 1 2 3 5 6 7	1

Комментарий

Пусть отрезки первого типа — система противопехотных ежей, второго типа - шеренги солдат.

Тогда в первом примере можно сдвинуть отрезки второго типа на 1 вправо. Тогда отрезки первого и второго типа не будут пересекаться, следовательно ответ будет равен 0.

Во втором примере выгодно сдвинуть отрезки второго типа на 1 влево.

Отрезки первого типа двигать нельзя.

Задача G. Игра

Имя входного файла:	`suffixgame.in`
Имя выходного файла:	`suffixgame.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Джейме и Серсея играют в игру.

У них есть отсортированный по возрастанию массив. За один ход игрок может отнять единицу от каждого числа на некотором суффиксе массива. Проигрывает игрок, после хода которого в массиве появляется два одинаковых числа, либо одно из чисел становится равным нулю.

Джейме ходит первым, Серсея — второй. Кто выиграет, если оба игрока будут играть оптимально?

Формат входного файла

В первой строке входного файла находится целое число n $(1 \le n \le 10^5)$ — количество чисел в массиве.

Во второй строке через пробел записаны n различных целых чисел a_i $(1 \le a_i \le 10^9, a_{i-1} < a_i)$.

Формат выходного файла

В единственной строке выходного файла выведите `First`, если выиграет Джейме и `Second`, если выиграет Серсея.

Пример

suffixgame.in	suffixgame.out
5 1 2 3 4 5	Second
1 100	First

Задача H. Передача сообщения

Имя входного файла:	`message.in`
Имя выходного файла:	`message.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Кейтилин Старк хочет отослать своему мужу Эддарду тайное сообщение, которое является палиндромом. Но сообщение нельзя отсылать просто так — его определенно перехватят Ланнистеры, и тогда Кейтилин и её мужу несдобровать. Поэтому Кейтилин решила зашифровать своё послание. Она добавила в него несколько букв, и теперь её сообщение можно получить, взяв некоторую подстроку нового послания и удалив из неё не более k символов.

Когда Эддард получил письмо с зашифрованным сообщением, он не смог однозначно определить, что хотела передать ему жена. Но он хочет найти хотя бы самую длинную подстроку, являющуюся кандидатом на сообщение его жены — подстроку, из которой можно удалить не больше k символов так, чтобы она являлась палиндромом. Число k Эддард знает — это их любимое с Кейтилин число.

Помогите деснице короля решить эту непростую задачу, ведь незнание информации в присланном сообщении может грозить ему смертью.

Формат входного файла

В первой строке входного файла дано число k ($0 \le k \le 5000$) — максимальное количество символов, которое можно удалить.

Во второй строке входного файла дана строка s ($1 \le |s| \le 5000$) — сообщение, которое получил Эддард.

Сообщение состоит только из строчных букв латинского алфавита.

Формат выходного файла

В единственной строке выходного файла выведите через пробел два числа l, r ($1 \le l \le r \le |s|$), разделенные пробелом — левый и правый конец наидлиннейшей подстроки, которая является кандидатом на зашифрованное сообщение Кейтилин.

Если возможных ответов несколько, выведите тот, у которого левый конец левее всего.

Пример

message.in	message.out
0 abac	1 3
1 bacb	1 4

Задача I. Крестики-нолики

Имя входного файла:	`tic-tac-toe.in`
Имя выходного файла:	`tic-tac-toe.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Однажды, прогуливаясь по Королевской Гавани, лорд Петир Бейлиш нашел листок бумаги в клетку, исписанный странными символами. Оказалось, что кто-то играл на этом листке в крестики-нолики.

Лорд Бейлиш прекрасно знает правила этой игры. Игроки ходят по очереди, один из них ставит в любую свободную клетку поля крестик, второй — нолик. Петир Бейлиш не знает, кто играл эту партию в крестики-нолики, но ему очень хочется узнать, доиграна она или нет. Партия считается доигранной, если существует горизонтальная, вертикальная или диагональная линия из пяти крестиков или ноликов. Позиция может быть неккоректной, Петира это не волнует.

Лорд Бейлиш считает вас достойной кандидатурой, чтобы помочь ему. Ваша задача — написать программу, которая сможет определить, доиграна партия или нет.

Формат входного файла

В самой первой строке входного файла заданы числа n и m ($1 \le n, m \le 1000$) — размеры игрового поля.

В следующих n строках записано по m символов «X», «O» или «.», которые означают, что в данной клетке стоит крестик, нолик, либо она пуста, соответственно. Обращаем ваше внимание, что «X» и «O» — заглавные буквы латинского алфавита.

Формат выходного файла

В единственной строке выходного файла выведите `Yes`, если игра доиграна, и `No`, если нет.

Пример

tic-tac-toe.in	tic-tac-toe.out
5 6 .OXOXO OXX.O. XXOOXX ..O..O OOXX.X	Yes