

Задача А. Экономия на мыле

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Снижение расходов в ресторанном деле — залог успешного бизнеса.

Каждое утро дозатор вместимостью W миллилитров полностью заполняется мылом. В течение дня им пользуются посетители. После каждого нажатия из дозатора выливается X миллилитров жидкости. После любого использования можно долить в дозатор любое **целое** число миллилитров воды, не превышая при этом его вместимость W .

Если мыло окажется слишком сильно разбавлено водой или если в дозаторе будет меньше X миллилитров жидкости, посетители это заметят, рассердятся и перестанут ходить в ресторан. Поэтому перед каждым использованием дозатора в нём должно быть хотя бы X миллилитров жидкости, а доля мыла во всей жидкости должна быть не меньше $\frac{A}{B}$.

Требуется найти максимальное количество нажатий на дозатор, которое можно обеспечить так, чтобы никто из посетителей не рассердился.

Формат входных данных

В единственной строке записаны четыре целых числа W , X , A , B ($1 \leq W \leq 3 \cdot 10^8, 1 \leq X \leq W, 1 \leq A \leq B, 1 \leq B \leq 3 \cdot 10^8$).

Формат выходных данных

Выведите одно целое число — максимальное количество нажатий, которое вы можете обеспечить.

Система оценки

В задаче 50 наборов тестовых данных. За каждый тест, на который был найден правильный ответ, начисляется 2 балла.

Гарантируется, что решения, корректно работающие при $W, X, A, B \leq 10$, наберут не менее 30 баллов.

Также гарантируется, что решения, корректно работающие на тестах, ответы в которых не превосходят $4 \cdot 10^5$, наберут не менее 60 баллов.

Примеры

стандартный ввод	стандартный вывод
2 1 1 2	3
2 1 2 3	2
5 2 1 7	6

Замечание

В первом примере вместимость дозатора равна 2 миллилитрам, за одно использование из него выливается 1 миллилитр жидкости, а доля мыла должна быть не меньше $\frac{1}{2}$.

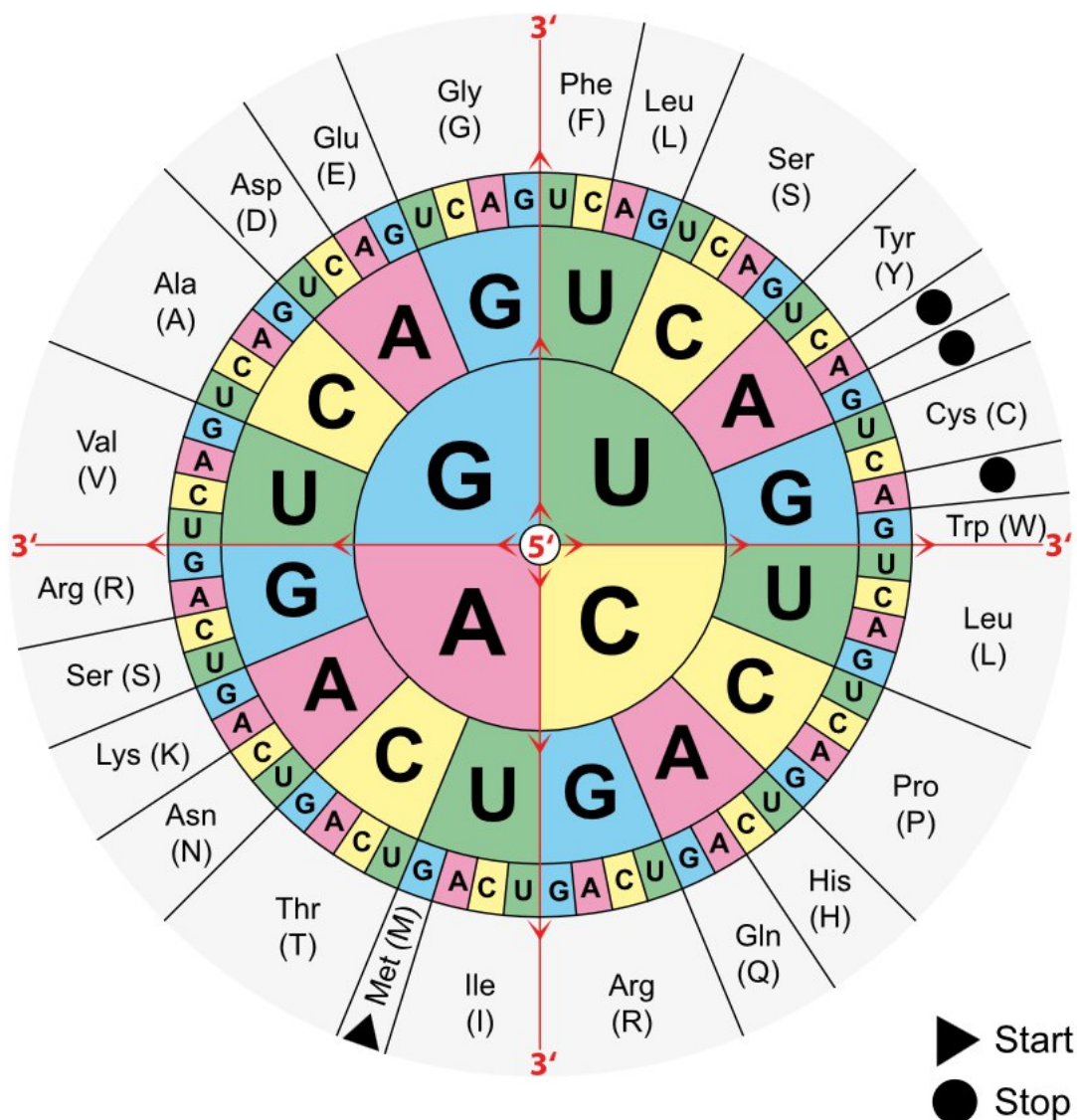
После первого использования можно долить 1 миллилитр воды. Тогда доля мыла станет равна $\frac{1}{2}$, и после этого доливать воду уже не получится. Оставшихся 2 миллилитров жидкости с долей мыла 50% хватит ещё на 2 использования.

Таким образом, ответ равен 3.

Задача В1. Мутация белка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	1024 мегабайта

РНК можно рассматривать как длинную цепочку, составленную из *нуклеотидов*. В этой задаче *нуклеотидом* будем называть один из символов 'U', 'C', 'A', 'G'. Три подряд идущих нуклеотида образуют *кодон* (порядок нуклеотидов в кодоне важен). Каждому кодону соответствует некоторая *аминокислота*, при этом нескольким различным кодам может соответствовать одна и та же аминокислота:



Соответствие кодонов аминокислотам

Выберем первый нуклеотид в центре, далее будем переходить на следующее кольцо по соответствующему нуклеотиду, соседнему со стороной фигуры.

Например, «AUG» — это кодон, который соответствует аминокислоте «М».

Обратите внимание, что старт кодон выглядит как «AUG», а стоп кодон — это один из вариантов: «UAA», «UAG», «UGA».

Будем говорить, что *белок* — это последовательность аминокислот, которая начинается с аминокислоты «М» и заканчивается одной из аминокислот, являющихся стоп-кодоном (такие аминокислоты

обозначены в таблице кружком). Стоп-кодон не может встречаться в середине белка. Аминокислота 'М', напротив, может встречаться несколько раз.

Биолог исследует белок длины n . При синтезе РНК произошла ошибка: в неё было вставлено ещё k нуклеотидов в произвольные места. В результате получилась последовательность из $3n + k$ нуклеотидов, и биолог хочет восстановить какой-нибудь белок длины n .

Формат входных данных

В первой строке записано одно целое число t — количество наборов входных данных. В этой версии задачи во всех тестах $t = 1$.

Во $2 \cdot (t + 1)$ -й строке записаны два целых числа n и k ($2 \leq n \leq 10^5$, $1 \leq k \leq 10^5$).

В $2 \cdot (t + 1) + 1$ -й строке записана строка длины $3n + k$, которая является последовательностью нуклеотидов. Гарантируется, что строка содержит только символы 'U', 'C', 'A', 'G'.

Формат выходных данных

Выведите t строк, в каждой из них k различных целых чисел — номера удаляемых нуклеотидов в тесте t . Номера можно выводить в любом порядке.

Система оценки

В этой версии задачи 40 тестов помимо тестового примера из условия, каждый оценивается в 1 балл. Тестовый пример не оценивается.

Длина восстановленного белка len определяется следующим образом.

Рассмотрим строку, полученную после удаления некоторых нуклеотидов, и разобьём её на кодоны. Затем найдём первое вхождение старт-кодона и первое вхождение стоп-кодона, расположенное после него. Тогда длиной белка будем считать количество аминокислот от 'М' до этого стоп-кодона включительно.

Количество баллов за тест определяется по формуле

$$score = \frac{len}{n}.$$

Обратите внимание, что если у вас выполнится одно из трёх условий:

1. Нет ни одного старт кодона
2. Нет ни одного стоп кодона
3. Все старт кодоны находятся после всех стоп кодонов

То вы получите 0 баллов за тест.

Подзадача	Баллы	Доп. ограничения
0	0	тесты из условия
1	8	$2 \leq n \leq 6$, $1 \leq k \leq 6$
2	8	$k = 1$
3	8	$k = 2$
4	8	$k = 3$
5	8	Вставка происходит только между соседними кодонами

Пример

стандартный ввод	стандартный вывод
1 2 1 AUUGUGA	3

Замечание

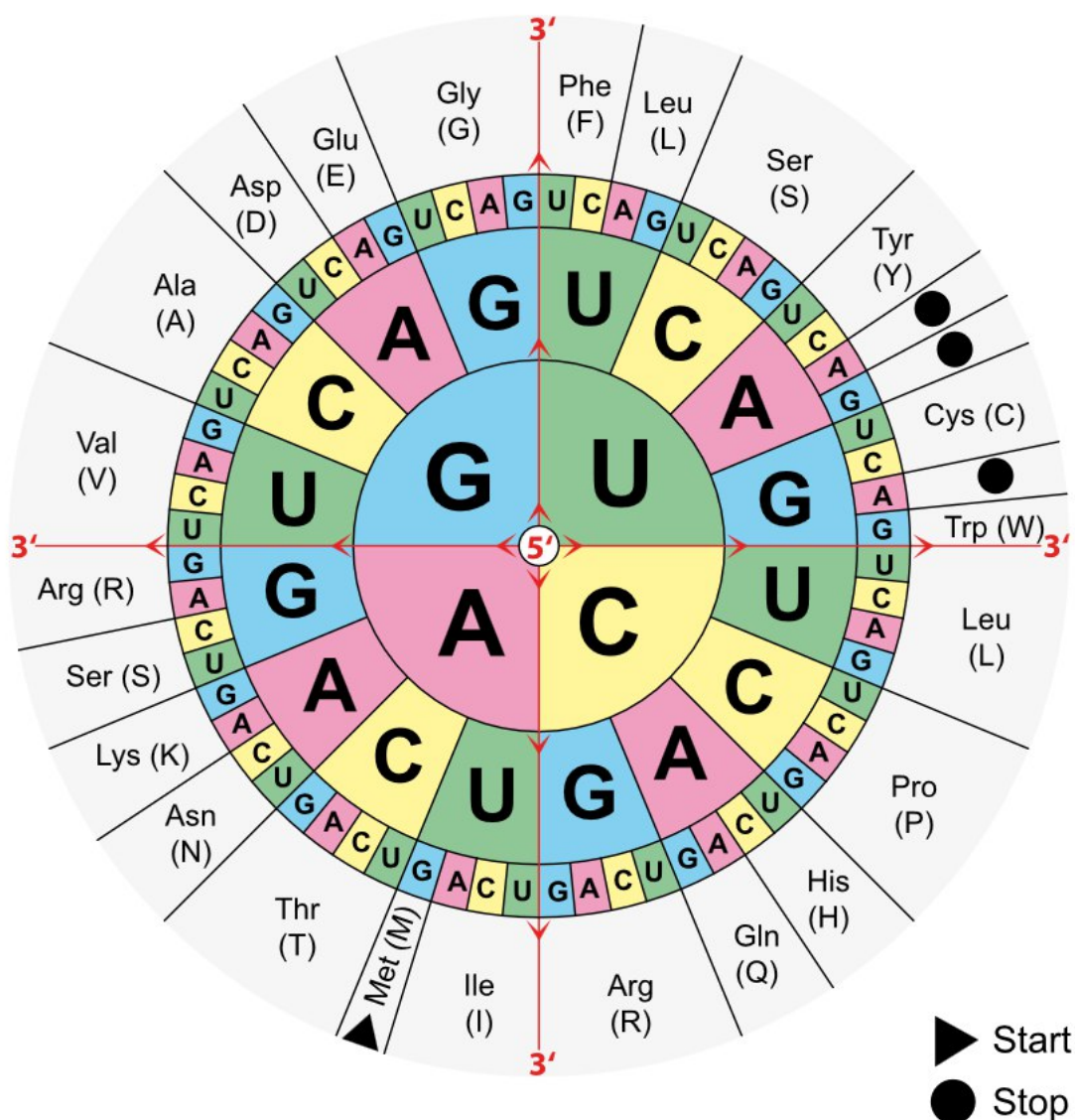
Для удобства обозначим стоп кодон как 'е'.

В первом примере после удаления нуклеотидов под номером 3 получим последовательность нуклеотидов AUGUGA, разобьём её на кодоны AUG, UGA — это соответствует белку Ме. Его длина равна 2.

Задача В2. Мутация белка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	15 секунд
Ограничение по памяти:	1024 мегабайта

РНК можно рассматривать как длинную цепочку, составленную из *нуклеотидов*. В этой задаче *нуклеотидом* будем называть один из символов 'U', 'C', 'A', 'G'. Три подряд идущих нуклеотида образуют *кодон* (порядок нуклеотидов в кодоне важен). Каждому кодону соответствует некоторая *аминокислота*, при этом нескольким различным кодам может соответствовать одна и та же аминокислота:



Соответствие кодонов аминокислотам

Выберем первый нуклеотид в центре, далее будем переходить на следующее кольцо по соответствующему нуклеотиду, соседнему со стороной фигуры.

Например, «AUG» — это кодон, который соответствует аминокислоте «М».

Обратите внимание, что старт кодон выглядит как «AUG», а стоп кодон — это один из вариантов: «UAA», «UAG», «UGA».

Будем говорить, что *белок* — это последовательность аминокислот, которая начинается с аминокислоты «М» и заканчивается одной из аминокислот, являющихся стоп-кодоном (такие аминокислоты

обозначены в таблице кружком). Стоп-кодон не может встречаться в середине белка. Аминокислота ‘М’, напротив, может встречаться несколько раз.

Биолог исследует белок длины n . При синтезе РНК произошла ошибка: в неё было вставлено ещё k нуклеотидов в произвольные места. В результате получилась последовательность из $3n + k$ нуклеотидов, и биолог хочет восстановить какой-нибудь белок длины n .

Формат входных данных

В первой строке записано одно целое число t — количество тестов. В этой версии задачи $t = 20$.

Во $2 \cdot (t + 1)$ -й строке записаны два целых числа n и k ($2 \leq n \leq 10^5$, $1 \leq k \leq 10^5$).

В $2 \cdot (t + 1) + 1$ -й строке записана строка длины $3n + k$, которая является последовательностью нуклеотидов. Гарантируется, что строка содержит только символы ‘U’, ‘C’, ‘A’, ‘G’.

Формат выходных данных

Выведите t строк, в каждой из них k различных целых чисел — номера удаляемых нуклеотидов в тесте t . Номера можно выводить в любом порядке.

Система оценки

Длина восстановленного белка len определяется следующим образом.

Рассмотрим строку, полученную после удаления некоторых нуклеотидов, и разобьём её на кодоны. Затем найдём первое вхождение старт-кодона и первое вхождение стоп-кодона, расположенное после него. Тогда длиной белка будем считать количество аминокислот от ‘М’ до этого стоп-кодона включительно.

Количество баллов за тест определяется по формуле

$$score = \frac{len}{n} \cdot 3.$$

Обратите внимание, что если у вас выполнится одно из трёх условий:

1. Нет ни одного старт кодона;
2. Нет ни одного стоп кодона;
3. Все старт кодоны находятся после всех стоп кодонов;

то вы получите 0 баллов за тест.

Замечание

Для удобства обозначим стоп кодон как ‘e’.

В первом примере после удаления нуклеотидов под номером 3 получим последовательность нуклеотидов AUGUGA, разобьём её на кодоны AUG, UGA — это соответствует белку Me. Его длина равна 2.

Задача С. Пирамидворд

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Пирамидворд — особый вид пространственного кроссворда.

Рассмотрим правильную четырехугольную пирамиду. Ее основание является квадратом, боковые ребра имеют такую же длину. На ребрах пирамиды можно записывать слова из словаря, крайние буквы слова записываются в вершинах пирамиды (концах ребра). Слово на ребре может быть записано в любом направлении. Нельзя записывать одно и то же слово на нескольких ребрах.

Все слова в словаре имеют длину L . Гарантируется, что в словаре нет пары слов, таких, что второе слово является развернутым первым.

Определите, сколько пирамидвордов можно составить из слов заданного словаря. Два пирамидворда считаются различными, если существует ребро, на котором последовательность букв отличается. В частности, если после поворота нашлось такое ребро, то такие пирамидворды следует считать различными.

Формат входных данных

Первая строка содержит два целых числа n и L ($1 \leq n \leq 10^5, 3 \leq L \leq 10$) — количество слов в словаре и длина каждого слова в нем. Следующие n строк содержат описание слов из словаря. Каждое слово состоит из строчных букв английского алфавита. Гарантируется, что все слова в словаре различны, имеют одинаковую длину L , и не найдется двух различных слов, таких, что одно из них является развернутым вторым.

Формат выходных данных

В качестве ответа выведите число различных пирамидвордов, которые можно составить из слов этого словаря. Так как это число может быть слишком большим, выведите его по модулю $10^9 + 7$.

Система оценки

Оценка за задачу — 100 баллов. Тестирование производится онлайн. Каждый набор тестовых данных оценивается в 2 балла.

Гарантируется, что решения, работающие с $n \leq 12$, набирают не менее 20 баллов.

Гарантируется, что решения, работающие со словарями, состоящими из слов с невозрастающей/неубывающей последовательностью букв, у которых первая и последняя буквы различны, набирают не менее 30 баллов.

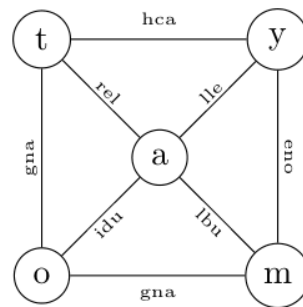
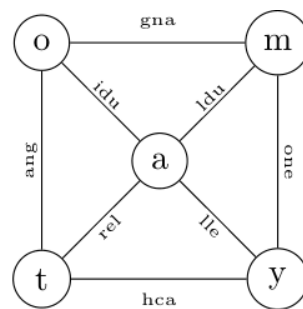
Гарантируется, что решения, работающие со словарями, в которых не более 13 различных букв, набирают не менее 90 баллов.

Пример

стандартный ввод	стандартный вывод
8 5 album alert alley audio money yacht tango mango	8

Замечание

Ниже приведены все 8 пирамидвордов, которые можно получить из набора слов в примере.



Задача D1. Взаимнопростая таблица

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В клетки таблицы $n \times n$ необходимо расставить числа от 1 до n^2 , каждое по одному разу. Расстановка тем лучше, чем меньше в ней *плохих* пар соседних клеток. Соседними считаются клетки, имеющие общую сторону. Пара соседних клеток считается *плохой*, если числа в этих клетках **не** являются взаимно простыми. Два числа называются взаимно простыми, если их наибольший общий делитель равен 1.

Составьте таблицу с наименьшим количеством плохих пар клеток.

Формат входных данных

В первой строке задается количество наборов входных данных T . В этой задаче T всегда равно 1.

В следующих T строках содержатся описания входных данных. Каждое описание состоит из одного целого числа n ($2 \leq n \leq 100$) — размера таблицы.

Формат выходных данных

Для каждого набора входных данных выведите n строк, в каждой из которых должно быть по n целых чисел — выбранная таблица.

Все числа от 1 до n^2 должны встретиться в таблице ровно один раз.

Система оценки

Оценка за эту задачу — 60 баллов.

В этой задаче 60 тестов, и каждый тест оценивается максимум в 1 балл. Оценка за тест вычисляется как средняя оценка по наборам входных данных в тесте. В этой задаче оценка за тест равна оценке за единственный набор входных данных соответственно. Оценка за набор входных данных вычисляется по формуле:

$$grade = 1 - 1.7 \cdot \frac{bad}{n^2}$$

где bad и n — количество плохих пар и размер таблицы соответственно.

Пример

стандартный ввод	стандартный вывод
1	1 4
2	2 3

Задача D2. Егор и взаимнопростая таблица — 2

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В этой задаче на проверку необходимо сдать текстовый файл с ответом. Входные данные вы можете скачать, нажав на кнопку с изображением стрелки справа-сверху рядом с кнопкой «Объявления жюри». Обратите внимание на увеличенные ограничения на n в этой задаче по сравнению с предыдущей версией.

Давным-давно Егор и его друзья соревновались в решении задач по информатике. Все задачи с единственным ответом они на тот момент уже решили и придумали для себя новое испытание. У них была таблица $n \times n$, и они расставляли в ее клетки числа от 1 до n^2 , каждое по одному разу. Расстановка считалась тем лучше, чем меньше в ней *плохих* пар соседних клеток. Пара соседних клеток считается *плохой*, если числа в этих клетках **не** являются взаимно простыми. Егор и его друзья уже выросли, так что теперь решать эту задачу предстоит вам.

Два числа называются взаимно простыми, если их наибольший общий делитель равен 1.

Клетки считаются соседними, если они имеют общую сторону.

Формат входных данных

В первой строке задается количество наборов входных данных T . В этой задаче T всегда равно 5.

В первой строке каждого описания набора входных данных дано одно целое число n ($2 \leq n \leq 1000$) — размер таблицы.

Формат выходных данных

Для каждого набора входных данных выведите n строк, в каждой из которых должно быть по n целых чисел — выбранную таблицу.

Все числа от 1 до n^2 должны встретиться в таблице ровно один раз.

Система оценки

Оценка за эту задачу — 40 баллов.

В этой задаче 1 тест, оценивающийся максимум в 40 баллов. Оценка за тест вычисляется как средняя оценка по наборам входных данных в тесте, умноженная на максимальный балл за тест. В этой задаче в единственном тесте — 5 наборов входных данных. Оценка за набор входных данных вычисляется по формуле:

$$grade = \max(1 - 1.7 \cdot \frac{bad}{n^2}, 0)$$

где bad и n — количество плохих пар и размер таблицы соответственно.

Задача Е. Трудоголики

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	6 секунд
Ограничение по памяти:	1024 мегабайта

Андрей и Олег очень любят работать, но у них нет своего офиса. Поэтому они работают в коворкингах.

Всего в доме n коворкингов, расположенных на одной прямой. i -й коворкинг имеет координату a_i на этой прямой. Никакие два коворкинга не находятся в одной точке. У каждого коворкинга есть режим работы: i -й коворкинг открывается в момент времени o_i и закрывается в момент времени c_i .

Андрей и Олег хотят провести максимально возможное время в коворкингах. За единицу времени они могут переместиться влево либо вправо, изменив свою координату на -1 или 1 соответственно. Помогите им найти максимальное время пребывания в коворкингах. В момент времени 0 они находятся в точке с координатой 0 .

Формат входных данных

В первой строке задано одно число n ($1 \leq n \leq 10^5$) — количество коворкингов. В каждой из следующих n строк находится по три целых числа a_i, o_i, c_i ($1 \leq a_i \leq 10^8, 1 \leq o_i < c_i \leq 10^8$) — координата i -го коворкинга, время открытия и закрытия i -го коворкинга.

Формат выходных данных

Выведите одно число — максимальное время пребывания Андрея и Олега в коворкингах.

Система оценки

В этой задаче проверка осуществляется по подгруппам. Баллы за первую и вторую подгруппы начисляются в случае прохождения всех тестов в них. Подгруппы 3 и 4 содержат 10 тестов, каждый из которых оценивается в 3 балла.

Обозначим максимальное значение c_i по всем i от 1 до n как C .

Обозначим максимальное значение a_i по всем i от 1 до n как A .

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи
1	30	$n \leq 1000, A, C \leq 10000$	-
2	40	$n \leq 1000$	1
3	18	$A, C \leq 2 \cdot 10^5$	1, 2
4	12	нет	1, 2, 3

Примеры

стандартный ввод	стандартный вывод
2 1 1 6 2 5 8	6
4 4 5 8 1 11 16 6 11 15 8 17 19	9

Замечание

В первом примере оптимальной стратегией для Олега и Андрея будет следующая: с самого начала (момент 0 , координата 0) им нужно выдвинуться в координату 1 (в первый коворкинг).

Они придут туда в момент времени 1. С момента времени 1 до момента 6 они пробудут в первом коворкинге, после чего он закроется. В момент 6 Андрей и Олег начнут движение из координаты 1 в координату 2, во второй коворкинг. Они придут в коворкинг в момент времени 7, и он закроется в 8. Итоговый ответ: $(6 - 1) + (8 - 7) = 5 + 1 = 6$.