

Задача A. Anime Creator

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Это интерактивная задача.

Великий сочинитель исекаев Поликарп Монокарпович недавно посмотрел аниме Наруто, оно ему очень понравилось и он решил написать своего Наруто. В его версии аниме «Наруто 2023» изначально есть N персонажей, которых мы будем обозначать, как $\alpha_1, \alpha_2, \dots, \alpha_n$. Из-за сложной механики ведения боя, силы персонажей не транзитивны (то есть, если персонаж A сильнее B , а B сильнее C , то из этого не следует, что A сильнее C). Более того, в первых сериях он продемонстрировал, что α_1 сильнее α_2 , α_2 сильнее α_3 , \dots , α_n сильнее α_1 .

Тем не менее после этого у Поликарпа не было идей для дальнейшего сюжета, в то время как аниме стало очень популярным. Поэтому он позвал вас на помощь! Каждая серия может быть, как флешбеком, так и относиться к текущему таймлайну, в каждой серии происходит бой между парой персонажей. Разница между флешбеками и основным таймлайном в том, что после боя в серии из текущего таймлайна более слабый персонаж умирает (и не может появляться в последующих сериях), а после флешбека оба остаются в живых, при этом побеждает всё равно более сильный. Таким образом, каждая серия описывается тройкой чисел (i, j, t) , где i и j – номера сражающихся персонажей, а t – индикатор, является ли серия флешбеком.

У Поликарпа есть свое мнение насчет каждой пары персонажей, которое не меняется со временем. Если в каком-то флешбеке или каких-то первых сериях α_i оказался сильнее α_j , то так будет, как в дальнейших флешбеках, так и в боях во время основного таймлайна.

Критикам аниме будет казаться интересным, если нельзя разбить всех живых персонажей на две группы A и B такие, что в показанных сериях ни один персонаж из группы B не победил ни одного персонажа из группы A (причём считается победа как во флешбеке, так и в серии из основного таймлайна).

Ваша цель заключается в том, чтобы показав не более, чем $2N$ флешбеков, уменьшить количество персонажей до 3, но при этом если в какой-то момент времени аниме окажется неинтересным, то вас гарантированно уволят с комментарием «Wrong Answer».

Протокол взаимодействия

В самом начале вашей программе будет доступно единственное число t на вход – количество наборов тестовых данных.

В начале обработки каждого набора тестовых данных на ввод будет доступно единственное число n ($4 \leq n \leq 500$) – количество персонажей. Затем вы сможете предлагать серии, выводя строки вида $i \ j \ t$ ($1 \leq i, j \leq n$, $0 \leq t \leq 1$, $i \neq j$), где $t = 1$, если серия является флешбеком и $t = 0$ в ином случае. После этого на ввод будет доступно единственное число x – номер победившего персонажа. После боя, в результате которого останутся единственные 3 выживших персонажа, обработка текущего тестового случая должна быть немедленно завершена.

Гарантируется, что сумма n по всем тестам не превышает 10^4 .

Если в какой-то момент времени программа жюри вывела -1 , то вы должны немедленно завершить работу своей программы, иначе жюри не гарантирует, какой именно вердикт она получит.

Не забывайте после каждого запроса выводить символ перевода строки, а также сбрасывать буфер вывода командой `flush`.

Пример

стандартный ввод	стандартный вывод
1 4 3 3	 1 3 1 3 4 0

Задача B. Binary String Transformation

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Вам даны две бинарные строки A и B , каждая длиной N . Ваша задача — проверить, возможно ли трансформировать строку A в строку B , используя следующие операции:

- Выбрать подстроку в A , начинающуюся на позиции L и заканчивающуюся позицией R , такую, что $R - L \geq 3$, $A_L = A_R = 0$, $A_{L+1} = \dots = A_{R-1} = 1$.
- Обменять цифру на позиции L с цифрой на позиции $L + 1$ и цифру на позиции R с цифрой на позиции $R - 1$.

Формат входных данных

Первая строка входных данных содержит одно целое число N ($1 \leq N \leq 10^5$) — длину входных битовых строк.

Вторая строка содержит битовую строку A длины N .

Третья строка содержит битовую строку B длины N .

Формат выходных данных

Выведите «Yes», если A может быть преобразована в B с использованием нуля или более описанных в условии операций, и «No» в противном случае.

Примеры

стандартный ввод	стандартный вывод
8 01101110 10101101	Yes
7 0111110 1101011	Yes
6 101010 111000	No
9 011011110 011011110	Yes

Замечание

В первом примере сначала применяется операция с $L = 1$ and $R = 4$, получаем строку 10011110, затем применяем операцию с $L = 3$ и $R = 8$, получая требуемую строку 10101101.

Задача C. Cotangent

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Пусть функция $S(k, l)$ от двух целых аргументов k и l , таких, что $k \bmod 3 \neq 0$ и $(k, l) = 1$ определена как сумма

$$\sum_{j=1}^{k-1} \operatorname{ctg}(\pi \cdot j/k) \cdot \operatorname{ctg}(l \cdot \pi \cdot j/k)$$

Вам даны k и l , подсчитайте $S(k, l)$.

Формат входных данных

В одной строке через пробел заданы два числа k и l ($2 \leq k, l \leq 10^9$, k не делится на 3, k и l взаимно просты).

Формат выходных данных

Выведите значение функции с абсолютной или относительной погрешностью не хуже 10^{-9} .

Примеры

стандартный ввод	стандартный вывод
10 3	0
11 7	-6
11345435 7	6129463102750

Задача D. Distinct Numbers

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Дана последовательность из n целых чисел v_i , каждое — случайное число, выбранное равномерно среди чисел от 1 до n .

Обработайте запросы. Каждый запрос — это два случайных отрезка $[a_l, a_r]$ и $[b_l, b_r]$. Концы каждого из отрезков — случайные целые числа, выбранные равномерно среди чисел от 1 до n и затем отсортированные по неубыванию.

Для каждой такой пары требуется определить количество различных чисел в этих отрезках.

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 200\,000$).

Вторая строка содержит n случайных целых чисел v_i ($1 \leq v_i \leq n$).

Третья строка содержит целое число m ($1 \leq m \leq 200\,000$).

В следующих m строках содержатся запросы a_l, a_r, b_l, b_r ($1 \leq a_l \leq a_r \leq n; 1 \leq b_l \leq b_r \leq n$).

Формат выходных данных

На каждый запрос выведите ответ в отдельной строке.

Пример

стандартный ввод	стандартный вывод
15	9
14 2 4 14 4 7 2 5 11 6 4 14 8 2 10	8
10	4
10 15 4 15	7
7 14 7 15	7
1 3 11 13	8
7 14 2 2	9
5 11 3 5	7
5 13 1 2	8
4 15 3 9	3
4 8 3 12	
3 10 8 13	
6 6 12 13	

Задача E. Encoded Palindrome

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Шестнадцатеричная строка, то есть строка, состоящая из цифр '0' - '9' и букв 'A'-'F' называется *закодированным палиндромом*, если двоичная строка, полученная в результате замены шестнадцатеричных цифр их **четырёхбитовым** представлением (без удаления ведущих нулей), является палиндромом.

Вам дана шестнадцатеричная строка. Ваша задача — переупорядочить буквы и выкинуть как можно меньшее их количество так, чтобы получившаяся строка была закодированным палиндромом. Если таких палиндромов несколько, выведите строку, которая является лексикографически наибольшей. Если получить закодированный палиндром нельзя, выведите пустую строку (так как она является закодированным палиндромом длины 0).

Формат входных данных

Входные данные содержат строку S , состоящую из цифр '0' - '9' и букв 'A'-'F' ($1 \leq |S| \leq 10^5$).

Формат выходных данных

Выведите **шестнадцатеричную** строку наибольшей длины, которая является закодированным палиндромом и может быть получена из строки S перестановкой составляющих её символов и, возможно, удалением какого-то их числа. Если таких строк несколько, выводить надо лексикографически наибольшую.

Примеры

стандартный ввод	стандартный вывод
10836006F	860F061
33	

Задача F. Fruitland

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Вы планируете поехать во Фрутлэнд, чтобы там купить большую партию фруктов и потом выгодно продать в своей стране. У вас с собой есть X монет, и по законам Фрутлэнда вы не можете вывозить из страны фрукты стоимостью более Y монет.

Всего в Фрутлэнде N городов, в каждом из которых находится по одному оптовому рынку фруктов. Города занумерованы последовательными целыми числами от 1 до N . На каждом рынке вы можете купить различные фрукты. Фрукты продаются в контейнерах; вы можете купить только целое количество контейнеров фруктов каждого типа, не большее, чем количество контейнеров c_i соответствующего типа фруктов имеется на складе рынка. Каждый тип фруктов стоит p_i монет и даёт прибыль v_i за один контейнер.

Прямая поездка из города i в город j стоит $m_{i,j}$ монет. Заметим, что возможна ситуация, когда $m_{i,j} \neq m_{j,i}$.

Ваша задача — стартовать в городе 1, купить фрукты таким образом, чтобы суммарная прибыль была максимизирована, и вернуться в город 1. При этом вы не можете потратить более X монет всего и более Y монет непосредственно на покупку фруктов.

Формат входных данных

Первая строка входных данных содержит три целых числа — количество городов N , количество монет X , которые у вас есть с собой и максимальное количество монет Y , которые по законодательству Фрутлэнда можно потратить на покупку фруктов, соответственно ($1 \leq N \leq 14$, $1 \leq X \leq 10^4$, $1 \leq Y \leq \min(1000, X)$).

Далее следуют N блоков с информацией о фруктовых рынках в городах. Первая строка i -го из этих блоков содержит одно целое число K_i ($1 \leq K_i \leq 300$) — количество типов фруктов, доступных на рынке i -го города.

j -я из последующих K_i строк содержит три целых числа $p_{i,j}$, $v_{i,j}$ и $c_{i,j}$ — цену одного контейнера фруктов, прибыль с этого контейнера и количество доступных контейнеров, соответственно ($1 \leq p_{i,j}, v_{i,j}, c_{i,j} \leq 1000$).

Далее следует матрица $N \times N$, в которой j -е число в i -й строке $m_{i,j}$ задаёт цену прямой поездки из i -го города в j -й ($0 \leq m_{i,j} \leq 10^4$, $m_{i,i} = 0$ для всех i).

Формат выходных данных

Выведите одно целое число — максимальную суммарную прибыль.

Примеры

стандартный ввод	стандартный вывод
1 12 12 3 1 12 1 2 22 3 3 32 4 0	130
2 20 20 3 1 16 1 2 27 3 3 38 2 1 5 179 1 0 2 3 0	314
3 20 10 1 2 2 2 1 5 5 5 1 7 7 7 0 1 0 1 0 0 0 1 0	10

Задача G. Game Of Risk of Rain

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

В игре-шутере **Рискованный дождь** есть следующая игровая механика. С некоторой вероятностью a выстрел может стать критическим.

Есть предмет — **Клевер**, который работает следующим образом: если выстрел не стал критическим, происходит ещё одна дополнительная проверка на то, является ли выстрел критическим.

Если **Клеверов** k штук, то проверка, является ли выстрел критическим, проходит k дополнительных раз.

Выясните, при каком минимальном проценте изначальной вероятности критического попадания a , имея k **Клеверов**, получится итоговая вероятность не менее b процентов.

Формат входных данных

В первой строке указано целое число t — количество тестовых примеров ($1 \leq t \leq 2020$).

Далее следует t строк. Каждая из них имеет следующий формат:

В строке указано два целых числа b и k ($0 \leq b \leq 100$, $1 \leq k \leq 20$) — итоговая вероятность критического выстрела (в процентах) и количество **Клеверов**.

Формат выходных данных

Выведите t строк, в каждой из которых ответ на текущий тест — одно целое число a .

Пример

стандартный ввод	стандартный вывод
2	30
50 1	90
99 1	

Задача Н. How Many Coins Are Taken?

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Есть прямоугольная доска из клеток размера $n \times m$. На пересечении i -й строки и j -о столбца находится клетка с координатами (i, j) .

Изначально Алиса положила на каждую клетку ровно одну монету.

Потом пришёл Боб и произвёл k действий из возможных двух типов:

- «+ s » — забрать все монеты, находящиеся на клетках доски таких, что $i + j = s$
($2 \leq s \leq n + m$)
- «- d » — забрать все монеты, находящиеся на клетках доски таких, что $i - j = d$
($1 - m \leq d \leq n - 1$)

Определите, сколько монет забрал Боб.

Формат входных данных

Первая строка содержит два целых числа n и m — размеры доски ($1 \leq n, m \leq 10^9$).

Вторая строка содержит целое число k ($0 \leq k \leq 300\,000$).

Следующие k строк содержат действия по взятию монет в формате, описанном выше.

Формат выходных данных

Выведите единственное число — количество взятых монет.

Пример

стандартный ввод	стандартный вывод
4 4 3 + 3 + 6 - 0	8

Задача I. Integers And Bitcount

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Придумайте два целых неотрицательных числа x и y таких, что $\text{bitcount}(x) = a$, $\text{bitcount}(y) = b$ и $\text{bitcount}(x + y) = c$.

$\text{bitcount}(x)$ — количество единиц в двоичной записи числа x .

Формат входных данных

В единственной строке содержатся три целых числа a , b и c ($0 \leq a, b, c \leq 10^5$).

Формат выходных данных

Если требуемых чисел не существует, выведите «No».

Иначе в первую строку выведите «Yes», во вторую — число x , а в третью — число y .

Числа следует выводить в двоичной системе счисления. Длина каждого числа не должна превышать 10^6 . Числам разрешается иметь ведущие нули.

Примеры

стандартный ввод	стандартный вывод
1 1 1	Yes 1 1
1 2 3	Yes 10 101
1 1 10	No

Задача J. Jumping Subsequences

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Задача про запросы на отрезке, задача про подпоследовательность, задача про запросы на отрезке, снова задача про подпоследовательность, ещё задача про запросы на отрезке...

Какое интересное чередование, не правда ли? А что же будет, если это объединить?

Вам дана строка S из строчных букв латинского алфавита. Также даны запросы, каждый из которых — отрезок l_i, r_i . На каждый запрос требуется определить длину наибольшей **чередующейся** подпоследовательности, принадлежащей подстроке $S[l..r]$.

Последовательность символов называется чередующейся, если она имеет вид «хухух...», для некоторых **различных** букв x и y .

Также, вам необходимо предоставить эти две буквы x и y , которые формируют наибольшую подпоследовательность. Если таких пар символов несколько, то выведите ту, у которой первый символ раньше в алфавитном порядке, а при равенстве первого символа ту, у которой второй символ раньше.

Формат входных данных

Первая строка содержит S ($1 \leq |S| \leq 1\,500\,000$).

Вторая строка содержит целое число m — количество отрезков ($1 \leq m \leq 100\,000$).

В следующих m строках содержатся целые числа l_i, r_i , означающие границы отрезка очередного запроса ($1 \leq l_i \leq r_i \leq |S|$).

Формат выходных данных

Для каждого запроса выведите ответ в отдельной строке: сначала длину искомой подпоследовательности, а следом буквы, которые её формируют.

Пример

стандартный ввод	стандартный вывод
abacaba	5 ab
6	3 ac
1 7	3 ab
2 6	2 ba
1 3	2 ac
2 3	1 ca
2 4	
4 4	

Задача К. Keep Them Equal

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Перед вами дерево — связный граф с n вершинами и $n - 1$ ребром.

Когда из дерева удаляется вершина, то оно распадается на несколько деревьев.

Для каждой вершины вы должны найти вторую вершину такую, что если эти две вершины удалить из дерева, то размер наибольшего из получившихся деревьев будет минимально возможным. Размером дерева считается количество вершин в этом дереве.

Формат входных данных

Первая строка содержит целое число n — количество вершин ($2 \leq n \leq 300\,000$).

Следующие $n - 1$ строк содержат по паре целых чисел a_i и b_i , означающих, что данные вершины соединены ребром ($1 \leq a_i, b_i \leq n$).

Формат выходных данных

Для каждого $1 \leq i \leq n$ выведите в отдельной строке номер требуемой вершины (то есть вершины, которая должна быть удалена вместе с i -й).

Разрешается выводить любую подходящую вершину.

Пример

стандартный ввод	стандартный вывод
10	3
9 8	3
7 6	2
6 5	3
2 1	9
3 10	9
3 7	9
2 9	3
4 2	7
9 3	9

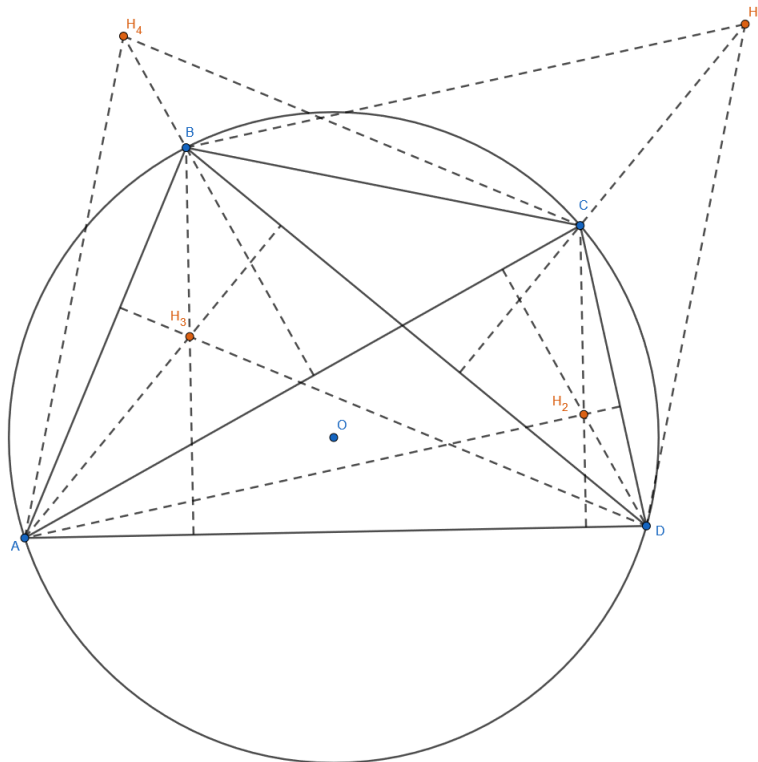
Задача L. Little Sasha and Circles

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Маленький Саша очень любит геометрию, он не очень хорошо в ней разбирается, но ему нравится рисовать красивые чертежи и видеть как различные простые факты сочетаются друг с другом, даря жизнь и смысл его картинкам.

В день перед олимпиадой Rucode Саша вернулся из детского садика и нарисовал следующую картинку. На белом листе бумаге он отметил точку O – центр декартовой системы координат с осями Ox и Oy , затем он нарисовал окружность с центром в точке O и отметил на ней четыре точки – A, B, C, D .

Затем он в каждом из треугольников BCD , ACD , ABD и ABC отметил точки пересечения высот H_1, H_2, H_3, H_4 , соответственно. Он уже хотел увидеть какой-нибудь замечательный факт, чтобы доказать его вечером, но пришли злые программисты и стерли весь рисунок, оставив только точки H_1, H_2, H_3, H_4 и O . Когда Саша увидел это, он расплакался и из-за эмоционального состояния уже не мог восстановить свой рисунок. Помогите ему это восстановить точки A, B, C и D или сообщите ему, что сделать это невозможно.



Так мог выглядеть рисунок Саши.

Формат входных данных

Входные данные состоят из четырёх строк. В i -й из этих строк указано по паре целых чисел x_i и y_i – координаты точки H_i . Все числа не превосходят 10^8 по абсолютной величине.

Формат выходных данных

Если однозначно восстановить точки, удовлетворяющие условию, возможно, то выведите «YES», а в последующих четырех строках координаты точек A, B, C, D по порядку, в ином случае выведите «NO». Координаты точек следует выводить с абсолютной или относительной погрешностью, не превышающей 10^{-9} .

Примеры

стандартный ввод	стандартный вывод
0 1 1 0 0 -1 -1 0	YES 0.000000000000 -1.000000000000 -1.000000000000 0.000000000000 0.000000000000 1.000000000000 1.000000000000 0.000000000000
0 1 0 2 0 -1 0 -2	YES 0.000000000000 -1.000000000000 0.000000000000 -2.000000000000 0.000000000000 1.000000000000 0.000000000000 2.000000000000

Задача M. Modular Recurrent Sequence

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Простой старшеклассник японской школы по имени Фибоначчи придумал рекуррентную последовательность $\{f_i\}$, у которой $f_0 = 0$, $f_1 = 1$, а $f_n = af_{n-1} + bf_{n-2} \bmod p$, где p – простое число и $1 \leq a, b < p$. Он изучил очень много ее свойств и решил выступить с докладом на фестивале школьных кружков. Но, к сожалению, на эту выставку приехала делегация 9 «В» класса школы имени Валентина Стрыкало, и девочка из этого класса по имени Альбина задала ему вопросы вида: «А есть ли такая позиция i , что $f_i = x$, а $f_{i+1} = y$ »? Фибоначчи, безусловно, знает свою последовательность наизусть и легко может ответить на все ее вопросы, но у него есть более интересные варианты досуга. Поэтому он просит вас ему помочь и написать программу, которая ответит на все вопросы Альбины.

Формат входных данных

В первой строке входных данных заданы 3 целых числа p, a, b ($0 < a, b < p \leq 10^6$), а во второй — одно целое число $1 \leq q \leq 10^6$ – количество запросов. В последующих q строках представлены вопросы Альбины, каждый из которых представляется парой чисел x и y ($0 \leq x, y < p$).

Формат выходных данных

Выведите ответы на запросы в порядке, в котором запросы заданы во входных данных. Ответом на запрос является число -1 , если пара (x, y) не встречается подряд в последовательности, или единственное натуральное число t_i – ответ на вопрос Альбины, если подходящих чисел несколько, то выведите минимальное.

Примеры

стандартный ввод	стандартный вывод
5 1 1 3 0 1 1 1 3 3	20 1 6
5 3 2 3 0 1 1 3 3 1	24 1 2