

Задача А. В свете софитов

Имя входного файла: стандартный ввод или `input.txt`
Имя выходного файла: стандартный вывод или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Директор картинной галереи Антон Антонович в замешательстве. На носу юбилейный выставочный сезон, а для жемчужины коллекции — гигантской «Картины картин» — не готово освещение! Поскольку светотехник галереи в отпуске, Антон Антонович решил разработать свою собственную схему подсветки.

Полотно занимает целую стену одного из залов от пола до потолка и представляет собой прямоугольник шириной w метров и высотой h метров (w и h — целые числа). В распоряжении галереи имеются два прожектора. Каждый прожектор освещает все точки внутри и на границе угла величиной 90° . Антон Антонович решил установить по одному прожектору у каждой из боковых сторон картины так, чтобы каждый прожектор был направлен на противоположную боковую сторону и биссектрисы углов, освещаемых прожекторами, были строго горизонтальны (см. иллюстрацию к примеру). Он разместил прожектор слева от картины на высоте y_1 метров, а прожектор справа от картины — на высоте y_2 метров.

Теперь Антон Антонович хочет узнать, не осталось ли на картине неосвещённых точек. К сожалению, проверить это непосредственно он не может, потому что электричество временно отключили. Помогите ему определить, будет ли картина полностью освещена двумя включёнными прожекторами.

Формат входных данных

В единственной строке находятся четыре целых числа w, h, y_1, y_2 ($2 \leq w, h \leq 10^6$, $1 \leq y_1, y_2 \leq h - 1$).

Формат выходных данных

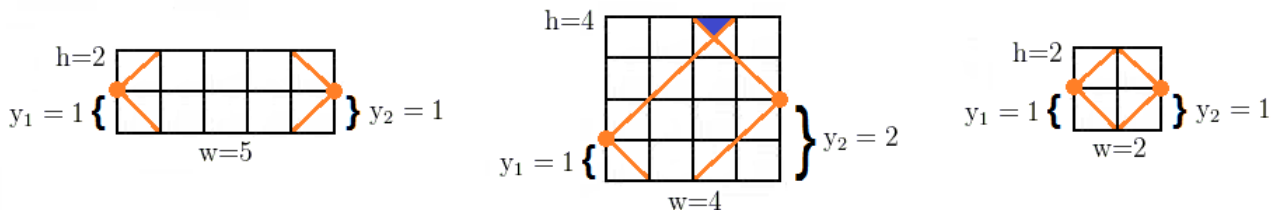
Если все точки картины будут освещены включёнными прожекторами, выведите «Yes» (без кавычек), иначе выведите «No» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
5 2 1 1	Yes
4 4 1 2	No
2 2 1 1	Yes

Замечание

Ниже представлены иллюстрации к примерам. На втором изображении заливкой выделен участок картины, который не будет освещён при включении прожекторов:



Задача В. Звёздное небо

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Юный астроном Даниэль любит смотреть на неисследованные части звездного неба и выискивать созвездия.

Будем считать, что звёздное небо представляет собой плоскость, на которой находятся n звёзд, являющихся различными точками, при этом никакие три из этих точек не лежат на одной прямой. Даниэль считает созвездием любой набор из k различных звёзд, являющихся вершинами выпуклого k -угольника (который возможно содержит другие звёзды внутри себя).

Ночь уже на исходе, а Даниэлю очень хотелось бы найти новое созвездие! Помогите ему это сделать, или определите, что на этот раз его ждёт разочарование.

Формат входных данных

В первой строке содержатся два числа n и k ($4 \leq n \leq 1000$, $4 \leq k \leq \min(6, n)$) — количество звёзд на небе и требуемый размер искомого созвездия.

Следующие n строк описывают координаты звёзд, i -я из этих строк содержит два целых числа x_i и y_i — координаты i -й звезды ($-10^6 \leq x_i, y_i \leq 10^6$).

Гарантируется, что никакие две звезды не находятся в одной точке и никакие три звезды не лежат на одной прямой.

Формат выходных данных

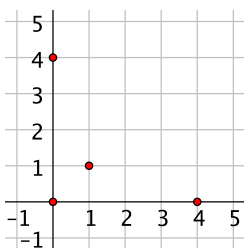
В первой строке выведите «Yes» (без кавычек), если у Даниэля есть шансы найти подходящее созвездие и «No» (без кавычек), если искомого созвездия не существует.

Если созвездие нужного размера присутствует на небе, последующие k строк должны содержать координаты k звезд, образующих искомое созвездие в порядке обхода против часовой стрелки. Если подходящих ответов несколько, разрешается вывести любой из них.

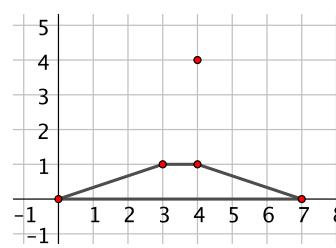
Примеры

стандартный ввод	стандартный вывод
4 4 0 0 1 1 0 4 4 0	No
5 4 0 0 7 0 3 1 4 1 4 4	Yes 0 0 7 0 4 1 3 1

Замечание



Первый тест из условия



Второй тест из условия

Задача С. Королевские вопросы

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

В одном средневековом королевстве бушует экономический кризис. Надои молока падают, экономические показатели каждый день пробивают новое дно, деньги из казны пропадают в неизвестном направлении. Чтобы исправить положение, король Карл Солнцеликий принял решение женить своих n сыновей-принцев на невестах s , по возможности, как можно бóльшим приданым.

В поисках кандидаток король отправил гонцов в соседние королевства, в результате чего ко двору прибыли делегации с m незамужними принцессами. Принимая гостей, Карл узнал, что приданое i -й принцессы составляет w_i золотых монет.

Хотя дело и происходит в средневековье, в обществе уже широко распространены прогрессивные представления, в соответствии с которыми никто не может заставить принцессу выйти замуж за принца, который ей не по нраву. Поэтому каждой принцессе была предоставлена возможность выбрать двух принцев, за каждого из которых она готова выйти замуж. Принцам повезло меньше, они в вопросе выбора невесты будут повиноваться воле своего отца.

Зная размер приданого и предпочтения каждой принцессы, Карл хочет сыграть свадьбы таким образом, чтобы суммарное приданое невест всех его сыновей было как можно больше. При этом женить всех принцев или выдавать всех принцесс замуж не обязательно. Каждый принц может жениться не более чем на одной принцессе, и наоборот каждая принцесса может быть выдана не более чем за одного принца.

Помогите королю организовать женитьбу своих сыновей наиболее выгодным для казны способом.

Формат входных данных

В первой строке находятся два целых числа n, m ($2 \leq n \leq 200\,000$, $1 \leq m \leq 200\,000$) — количество принцев и количество принцесс соответственно.

В каждой из последующих m строк находится по три целых числа a_i, b_i, w_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$, $1 \leq w_i \leq 10\,000$) — номера принцев, за которых готова выйти замуж i -я принцесса и размер её приданого.

Формат выходных данных

Выведите единственное целое число — максимальное количество золотых монет, которое может получить король, сыграв нужные свадьбы.

Примеры

стандартный ввод	стандартный вывод
2 3 1 2 5 1 2 1 2 1 10	15
3 2 1 2 10 3 2 20	30

Задача D. Национальное достояние

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Как вы все знаете, в стране Книгерии находится самая большая в мире библиотека. В ней есть десятки тысяч книг, написанных в самые разные периоды человеческой истории.

Прогресс не стоит на месте, и со временем руководство библиотеки приняло решение о создании электронной базы, содержащей названия всех книг, находящихся в библиотеке. Денис — программист, которому была поручена работа по созданию базы данных и по заведению в ней записей, отвечающих всем книгам в библиотеке.

Как типичный начинающий программист, вместо использования любого из имеющихся решений для этой задачи Денис решил написать свою базу данных с нуля. После нескольких месяцев кропотливого труда его база данных заработала, и он даже смог вбить в неё названия всех книг, но на презентации результата своих трудов он с удивлением узнал, что книги в базе данных должны следовать в лексикографическом порядке их названий. База данных Дениса ещё не поддерживает такую сложную процедуру, как сортировка, а перебивать все названия с нуля Денису очень не хочется, поэтому он решил схитрить.

Книгерийский алфавит настолько большой, что его буквы обозначаются целыми положительными числами. Каждая буква алфавита бывает строчной и прописной, прописная версия буквы, соответствующей числу x , обозначается как x' . Кодировка BSCII, используемая повсеместно в Книгерии, устроена таким образом, что прописные буквы упорядочиваются в порядке увеличения числа, обозначающего букву, аналогично упорядочиваются строчные буквы, но при этом все прописные буквы алфавита следуют до всех строчных букв алфавита. Например, верны следующие утверждения: $2 < 3$, $2' < 3'$, $3' < 2$.

Слово x_1, x_2, \dots, x_a не превосходит слова y_1, y_2, \dots, y_b в лексикографическом порядке если верно одно из двух условий:

- либо $a \leq b$ и $x_1 = y_1, \dots, x_a = y_a$, то есть первое слово является префиксом второго;
- либо есть такая позиция $1 \leq j \leq \min(a, b)$, что $x_1 = y_1, \dots, x_{j-1} = y_{j-1}$ и $x_j < y_j$, то есть, в первой позиции, в которой слова отличаются, в первом слове стоит меньшая буква.

Как следствие, слово «3' 7 5» следует в лексикографическом порядке до слова «2 4' 6». Говорят, что слова в последовательности следуют в лексикографическом порядке, если каждое слово последовательности лексикографически не превосходит следующего слова в последовательности.

На данный момент названия всех книг записаны строчными буквами. Денис хочет поменять некоторые буквы на прописные (такой процесс называется *капитализацией*) таким образом, чтобы названия книг стали следовать в лексикографическом порядке. Однако в процессе реализации этого плана возникла проблема — из-за неаккуратного устройства базы данных Денис не может капитализировать одну букву в конкретном названии. Единственная операция, которую он может проделать с буквой, это капитализировать все вхождения данной буквы в названия **всех** книг. Такую операцию можно проделать любое количество раз с произвольными буквами книгерийского алфавита.

Помогите Денису выбрать, какие буквы надо капитализировать, чтобы заданный порядок следования названий книг стал лексикографическим, либо определите, что это невозможно.

Обратите внимание, в библиотеке могут храниться книги с **одинаковыми** названиями.

Формат входных данных

В первой строке даются числа n и m ($2 \leq n \leq 100\,000$, $1 \leq m \leq 100\,000$) — количество книг в книгерийской библиотеке и количество букв в книгерийском алфавите соответственно. Буквы книгерийского алфавита обозначаются целыми числами от 1 до m .

Каждая из последующих n строк содержит описание книги в формате $l_i, s_{i,1}, s_{i,2}, \dots, s_{i,l_i}$ ($1 \leq l_i \leq 100\,000$, $1 \leq s_{i,j} \leq m$), где l_i обозначает длину названия книги, а $s_{i,j}$ задаёт последовательность букв в названии книги.

Гарантируется, что суммарная длина названий книг не превышает 100 000.

Формат выходных данных

В первой строке выведите «Yes» (без кавычек), если возможно капитализировать некоторое подмножество букв таким образом, чтобы последовательность названий книг стала отсортированной лексикографически. Иначе выведите «No» (без кавычек).

Если требуется возможно, во второй строке выведите k — количество букв, которые надо капитализировать, а в третьей строке выведите k различных чисел — номера этих букв. Обратите внимание, минимизировать значение k **не требуется**.

Номера букв можно выводить в любом порядке. Если существует несколько возможных ответов, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
4 3 1 2 1 1 3 1 3 2 2 1 1	Yes 2 2 3
6 5 2 1 2 2 1 2 3 1 2 3 2 1 5 2 4 4 2 4 4	Yes 0
4 3 4 3 2 2 1 3 1 1 3 3 2 3 3 2 3 1	No

Замечание

В первом примере названия книг после капитализации букв 2 и 3 выглядят следующим образом:

- 2'
- 1
- 1 3' 2'
- 1 1

Верно соотношение $2' < 1$, поэтому название первой книги лексикографически не превосходит названия второй. Название второй книги является префиксом названия третьей, следовательно они идут в лексикографическом порядке. Так как у названий третьей и четвёртой книги первая буква совпадает, а $3' < 1$, то название третьей книги не превосходит названия четвёртой книги лексикографически.

Во втором примере названия книг уже расположены в лексикографическом порядке, а значит можно не капитализировать буквы.

В третьем примере не существует набора букв, путём капитализации которых можно добиться, чтобы названия книг были расположены в лексикографическом порядке.

Задача Е. Сортировка монет

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Недавно Дима познакомился с Сашей в филателистическом магазине, и с тех пор они собирают монеты вместе. Их любимое занятие — сортировать коллекции монет. Саше нравится порядок, поэтому он хочет, чтобы монеты были расположены в ряд, причём сначала шли монеты, вышедшие из обращения, а потом монеты, всё ещё находящиеся в обращении.

Для упорядочивания монет Дима использует следующий алгоритм: за один шаг он просматривает все монеты слева направо, и если он видит, что i -я монета ещё в ходу, а $(i + 1)$ -я уже вышла из обращения, то он меняет эти две монеты местами, и продолжает смотреть на монеты дальше, начиная с $(i + 1)$ -й. Дойдя до конца, он спрашивает Сашу, доволен ли тот порядком монет. Дима повторяет эту операцию до тех пор, пока Саша не будет доволен. *Сложностью упорядочивания* Дима называет количество шагов, которые ему требуются в соответствии с процедурой, описанной выше, то есть, количество раз, которое он будет начинать просматривать монеты с начала. В частности, для уже упорядоченной исходной последовательности монет сложность упорядочивания равна единице.

Сегодня Саша в очередной раз позвал Диму в гости, чтобы предложить ему следующую игру. Сначала он выложил в ряд перед Димой n монет, все из которых вышли из обращения. Затем Саша n раз выбирает какую-то из монет, которая вышла из обращения, и заменяет её на монету, которая находится в обращении. В ходе этого процесса Саша постоянно интересуется у Димы, какова на данный момент сложность упорядочивания последовательности.

Задачу усложняет тот факт, что Диме нельзя трогать монеты, и поэтому определять сложность упорядочивания ему приходится в уме. Помогите Диме справиться с этим заданием.

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 300\,000$) — количество монет, которые Саша выложил в ряд перед Димой.

Следующая строка содержит n целых различных чисел p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$) — позиции монет, если смотреть слева направо, которые Саша меняет на монеты, находящиеся в ходу. Сначала Саша заменяет монету, находящуюся на позиции p_1 , затем монету, находящуюся на позиции p_2 и так далее.

Формат выходных данных

Выведите $n + 1$ чисел a_0, a_1, \dots, a_n , где a_0 — сложность упорядочивания последовательности в начале, a_1 — сложность упорядочивания после замены одной монеты и так далее.

Примеры

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

Замечание

Разберём первый пример. Будем обозначать за O монету, вышедшую из обращения, а за X — монету в обращении.

Изначально в ряду лежат монеты, вышедшие из обращения, поэтому Дима один раз просмотрит их слева направо, и не совершит никаких обменов.

После замены Сашей первой монеты на находящуюся в обращении Дима на первом шаге алгоритма три раза поменяет эту монету с последующей, после чего один раз просмотрит монеты и завершит процесс:

$$X000 \rightarrow 000X$$

После замены Сашей третьей монеты действия Димы выглядят следующим образом:

$$X0X0 \rightarrow 0X0X \rightarrow 00XX$$

После замены Сашей четвёртой монеты действия Димы выглядят следующим образом:

$$X0XX \rightarrow 0XXX$$

Наконец, после замены Сашей второй монеты весь ряд оказывается состоящим из монет, находящихся в обращении, и Дима один раз просмотрит ряд слева направо.

Задача F. Ор выше гор

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

Рик и Морти очень любят ходить на горный хребет Высокоорный для того, чтобы поорать — это там просто невероятное. Не так давно они нашли интересное акустическое свойство этого хребта: если Рик и Морти начнут одновременно орать с разных гор, то их ор будет слышен между этими горами вплоть до высоты, равной побитовому ИЛИ высот гор, на которые они взойшли, и всех гор между ними.

Побитовое ИЛИ — это бинарная операция, которая определяется следующим образом. Рассмотрим записи чисел x и y в двоичной системе счисления (возможно с ведущими нулями) $x = x_k \dots x_1 x_0$ и $y = y_k \dots y_1 y_0$. Тогда $z = x | y$ определяется следующим образом: $z = z_k \dots z_1 z_0$, где $z_i = 1$, если $x_i = 1$ или $y_i = 1$, иначе $z_i = 0$. Иными словами, нули в побитовом ИЛИ чисел находятся только в тех разрядах, в которых у обоих чисел находятся нули. Например, побитовое ИЛИ чисел $10 = 1010_2$ и $9 = 1001_2$ равняется $11 = 1011_2$. В языках программирования C/C++/Java/Python данная операция обозначается как «`|`», а в Pascal как «`or`».

Помогите Рик и Морти посчитать, сколькими способами они могут выбрать две различные горы так, что если они начнут орать с этих гор, ор их будет слышен выше этих гор и всех гор между ними. Формально говоря, требуется вычислить, сколько существует таких пар l и r ($1 \leq l < r \leq n$), что побитовое ИЛИ всех высот гор на отрезке от l до r включительно строго больше высоты любой горы на этом отрезке.

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 200\,000$) — количество гор в хребте. В следующей строке содержатся n целых чисел a_i ($0 \leq a_i \leq 10^9$) — высоты гор в порядке, в котором они следуют в хребте.

Формат выходных данных

Выведите одно число — искомое количество способов выбрать две различные горы.

Примеры

стандартный ввод	стандартный вывод
5 3 2 1 6 5	8
4 3 3 3 3	0

Замечание

В первом примере все искомые способы — это пары гор со следующими номерами (горы нумеруются с единицы):

$$(1, 4), (1, 5), (2, 3), (2, 4), (2, 5), (3, 4), (3, 5), (4, 5)$$

Во втором примере искомым пар не существует, поскольку для любой пары гор высота ора с них равна 3, и эта высота равна высоте любой из гор, следовательно она не выше их.

Задача G. Коробка конфет

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Председателю жюри в честь 15-летия московской командной олимпиады подарили коробку конфет с фантиками синего и красного цвета.

Председатель, пребывая в приятных размышлениях о предстоящей олимпиаде, через некоторое время все конфеты съел. При этом он придерживался следующего алгоритма: в каждый момент времени он выбирал конфету того цвета, которого в коробке было больше, а если красных и синих конфет было поровну, то председатель предпочитал красную.

Каждый раз, съедая очередную конфету, председатель клал фантик на стол справа от всех предыдущих пустых фантиков. Таким образом, когда всё содержимое коробки было съедено, все фантики были выложены на столе в порядке употребления содержащихся в них конфет. Правда из-за того, что некоторые фантики лежат лицом вниз, по ним нельзя точно сказать, какого они цвета.

У вас есть фотография стола, сделанная до того, как уборщица выкинула все фантики. Выясните, сколько существует различных коробок с конфетами, которые могли соответствовать полученной последовательности фантиков. Коробки считаются разными, если в них разное количество конфет с фантиками какого-то цвета.

Формат входных данных

В первой строке находится число n ($1 \leq n \leq 200\,000$) — число фантиков на столе.

Во второй строке записана строка длины n , задающая последовательность фантиков. Строка состоит из символов «r», «b» и «?», соответствующих красному фантику, синему и перевернутому (то есть, либо красному, либо синему).

Формат выходных данных

Выведите одно число — количество возможных коробок с конфетами, соответствующих полученной последовательности. Если последовательность фантиков не могла соответствовать никакой коробке конфет, выведите 0.

Примеры

стандартный ввод	стандартный вывод
4 ??rb	3
5 rrrrb	1
3 bbr	0

Замечание

В первом тесте подходит коробка, состоящая из 3 красных и 1 синей конфеты, коробка из 2 красных и 2 синих конфет, а также коробка состоящая из 1 красной и 3 синих конфет.

Во втором примере существует лишь одна коробка, подходящая под описание, в ней 4 красных и 1 синяя конфета.

В третьем примере задана несуществующая последовательность, если бы у председателя было 2 синих и 1 красная конфета, то он бы получил последовательность «brb».

Задача Н. Соляной рудник

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Влад приехал на экскурсию на соляной рудник, состоящий из n соляных пещер. Как Влад узнал от экскурсовода, эти пещеры соединены $n - 1$ туннелями, пронумерованными от 1 до $n - 1$. Кроме того, известно, что можно пройти от любой пещеры до любой другой пещеры, используя только данные туннели. Изучив карту рудника, Влад оценил для каждого туннеля его *красоту*. Для того, чтобы насладиться красотой природных объектов, необходимо смотреть на них с нужного ракурса. Поэтому красота туннеля зависит от того, в какую сторону идти по нему.

Для каждого туннеля i , соединяющего пещеры с номерами u_i и v_i , Влад определил два числа p_i и q_i — красоту туннеля, если пройти его от u_i до v_i и красоту туннеля, если пройти его от v_i до u_i соответственно. Красоты задаются целыми числами и могут быть отрицательными в случае, если соответствующий туннель показался Владу некрасивым.

Сейчас Влад находится в пещере с номером 1 и хочет пройти до пещеры с номером n , максимизировав сумму красот туннелей, которые он посмотрит за время своего путешествия. Однако Влад всё же боится заблудиться, поэтому вдоль каждого туннеля он хочет пройти не более, чем один раз в каждую сторону. Помогите Владу определить наиболее красивый маршрут.

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 100\,000$) — количество пещер в руднике.

В следующих $n - 1$ строках задано описание туннелей. Каждое описание состоит из четвёрки чисел u_i, v_i, p_i, q_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i, -10^4 \leq p_i, q_i \leq 10^4$) — номеров пещер, которые соединяет i -й туннель и значений его красоты при движении от u_i к v_i и при движении от v_i к u_i соответственно.

Формат выходных данных

Выведите одно целое число — максимально возможную красоту маршрута от пещеры с номером 1 до пещеры с номером n .

Примеры

стандартный ввод	стандартный вывод
3 1 2 5 2 2 3 1 3	6
5 1 2 1 1 2 5 1 1 1 4 2 -1 2 3 1 -2	3
4 1 4 1 1 2 3 -10 2 4 2 7 -6	2

Замечание

В первом тестовом примере Владу выгодно пойти из пещеры 1 в пещеру 2, а затем — из пещеры 2 в пещеру 3. В таком случае суммарная красота маршрута составит $5 + 1 = 6$.

Во втором тестовом примере Владу выгодно пройти из пещеры 1 в пещеру 4, затем вернуться в пещеру 1, а затем пройти до пещеры 5 через пещеру 2. Суммарная красота в таком случае равна $2 + (-1) + 1 + 1 = 3$.

В третьем тестовом примере Владу выгодно пройти из пещеры 1 в пещеру 4, затем из пещеры 4 в пещеру 2 и вернуться в пещеру 4. Красота в таком случае равна $1 + 7 + (-6) = 2$.

Задача I. Курьерский клуб

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Петя и Вася устроились работать курьерами. В течение очередного рабочего дня им нужно доставить пакеты в n различных точек на прямой. Согласно внутренним правилам компании, доставка пакетов должна быть произведена строго в определённом порядке. Исходно Петя находится в точке с координатой s_1 , Вася находится в точке с координатой s_2 , а клиенты находятся в точках x_1, x_2, \dots, x_n в порядке требуемого посещения.

Ребята заранее договариваются, кто из них доставит пакет для какого из клиентов, после чего они действуют следующим образом. Когда пакет для i -го клиента доставлен, тот из ребят, кто доставляет пакет для $(i + 1)$ -го клиента, отправляется в путь (это может быть как тот же из них, кто только что приехал в точку x_i , так и другой). Тот из друзей, который не занят в доставке текущего пакета, стоит на месте.

Для связи друг с другом ребятам выдали рации. Рации довольно плохо работают на больших расстояниях, поэтому Петя и Вася хотят распределить заказы так, чтобы максимальное расстояние между ними в течение дня было как можно меньше. Помогите Пете и Васе минимизировать максимальное расстояние между ними, соблюдая все правила доставки.

Формат входных данных

В первой строке даны три целых числа n, s_1, s_2 ($1 \leq n \leq 100\,000, 0 \leq s_1, s_2 \leq 10^9$) — количество точек доставки и начальные положения Пети и Васи.

Во второй строке расположены n целых чисел x_1, x_2, \dots, x_n — координаты клиентов ($0 \leq x_i \leq 10^9$), указанные в том порядке, в котором следует выполнять доставку.

Гарантируется, что среди чисел $s_1, s_2, x_1, \dots, x_n$ нет двух одинаковых.

Формат выходных данных

Выведите единственное целое число — минимальное возможное максимальное расстояние между ребятами в процессе доставки.

Примеры

стандартный ввод	стандартный вывод
2 0 10 5 6	10
3 2 1 3 4 5	1
1 4 5 2	2

Замечание

В первом примере изначальное расстояние между курьерами равно 10. Эта величина и будет ответом, например, Петя может выполнить обе доставки, а Вася останется в начальной точке.

Во втором примере оптимально можно действовать, например, так: Петя доставляет пакет первому клиенту, Вася — второму и, наконец, Петя выполняет доставку пакета третьему клиенту. При таком порядке доставки расстояние между курьерами никогда не превзойдет 1.

В третьем примере есть только два возможных варианта: если доставку единственного пакета осуществляет Петя, то максимальное расстояние между ними будет $5 - 2 = 3$. Если же пакет повезёт Вася, то максимальное расстояние составит $4 - 2 = 2$. Последний способ является оптимальным.

Задача J. Компания и побитовое И

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Маленькая девочка Алёна гуляла по лесу и увидела компанию из n программистов. Поговорив с ними она оценила *общительность* каждого человека в компании целым неотрицательным числом, и записала эти значения в виде последовательности a_1, a_2, \dots, a_n , i -й элемент которой соответствует общительности i -го программиста в компании.

Алёна считает, что *сила дружбы* двух программистов равна побитовому И величин их общительности. Так, сила дружбы i -го и j -го программистов из данной компании будет равна $a_i \& a_j$. *Сплочённостью* компании девочка называется сумму силы дружбы по всем парам двух различных программистов в компании. Помогите Алёне определить сплочённость встретившейся ей компании, то есть величину

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n a_i \& a_j$$

Дадим определение операции побитового И, обозначаемой в данном условии задачи как $\&$. Рассмотрим записи чисел x и y в двоичной системе счисления (возможно с ведущими нулями) $x = x_k \dots x_1 x_0$ и $y = y_k \dots y_1 y_0$. Тогда $z = x \& y$ определяется следующим образом: $z = z_k \dots z_1 z_0$, где $z_i = 1$, если $x_i = 1$ и $y_i = 1$, иначе $z_i = 0$. Иными словами, единицы в побитовом И чисел находятся в тех разрядах, в которых у обоих чисел находятся единицы. Например, побитовое И чисел $14 = 1110_2$ и $7 = 0111_2$ равняется $6 = 0110_2$. В языках программирования C/C++/Java/Python данная операция обозначается как « $\&$ », а в Pascal как « and ».

Формат входных данных

В первой строке находится число n ($1 \leq n \leq 150\,000$) — количество программистов в компании, встреченной Алёной.

Во второй строке находятся числа a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^8$), где a_i задаёт величину дружелюбия i -го человека.

Формат выходных данных

Выведите сплочённость встретившейся Алёне компании из n программистов.

Пример

стандартный ввод	стандартный вывод
5 4 3 5 2 3	13

Замечание

В тесте из условия сплочённость компании равна:

$$\begin{aligned} (4 \& 3) + (4 \& 5) + (4 \& 2) + (4 \& 3) + (3 \& 5) + (3 \& 2) + (3 \& 3) + (5 \& 2) + (5 \& 3) + (2 \& 3) \\ = 0 + 4 + 0 + 0 + 1 + 2 + 3 + 0 + 1 + 2 = 13 \end{aligned}$$