

Технокубок 2017: Условия задач отборочного этапа

1 отборочный раунд

А. Превращение: из А в В

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

У Василя есть число a , которое он хочет превратить в число b . Для этого он может производить два типа операций:

- умножить имеющееся у него число на 2 (то есть заменить число x числом $2 \cdot x$);
- приписать к имеющемуся у него числу цифру 1 справа (то есть заменить число x числом $10 \cdot x + 1$).

Вам надо помочь Василию получить из числа a число b с помощью описанных операций, либо сообщить, что это невозможно.

Обратите внимание, что в этой задаче не требуется минимизировать количество операций. Достаточно найти любой из способов получить из числа a число b .

Входные данные

В первой строке записаны два целых положительных числа a и b ($1 \leq a < b \leq 109$) — число, которое есть у Василя, и число, которое он хочет

получить.

Выходные данные

Если получить число b из числа a невозможно, выведите «NO» (без кавычек).

В противном случае в первую строку выведите «YES» (без кавычек). Во вторую строку выведите число k — количество чисел в последовательности превращений. В третьей строке выведите последовательность превращений x_1, x_2, \dots, x_k , причём:

- x_1 должно быть равно a ,
- x_k должно быть равно b ,
- число x_i должно быть получено с помощью одной из двух операций из числа x_{i-1} ($1 < i \leq k$).

Если ответов несколько, разрешается вывести любой из них.

Примеры

входные данные

2 162

выходные данные

YES

5

2 4 8 81 162

входные данные

4 42

выходные данные

NO

входные данные

100 40021

выходные данные

YES

5

100 200 2001 4002 40021

B. Сумма чека

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Василий вышел из магазина и ему стало интересно пересчитать сумму в чеке. Чек представляет собой строку, в которой названия покупок и их цены записаны подряд без пробелов. Чек имеет вид «*name1price1name2price2...namenpricen*», где *name_i* (название *i*-го продукта) — это непустая строка длины не более 10, состоящая из строчных букв латинского алфавита, а *price_i* (цена *i*-го продукта) — это непустая строка, состоящая из точек и цифр. Продукты с одинаковым названием могут иметь разные цены.

Цена каждого продукта записана в следующем формате. Если продукт стоит целое количество рублей, то копейки не пишутся.

Иначе, после записи количества рублей к цене приписывается точка, за которой следом **ровно двумя цифрами** записаны копейки (если копеек менее 10, то используется лидирующий ноль).

Также, каждые три разряда (от менее значимых к более значимым) в записи рублей разделяются точками. Лишние лидирующие нули недопустимы, запись цены всегда начинается с цифры и заканчивается

цифрой.

Например, записи цен:

- «234», «1.544», «149.431.10», «0.99» и «123.05» являются корректными,
- «.333», «3.33.11», «12.00», «.33», «0.1234» и «1.2» не являются корректными.

Напишите программу, которая по содержимому чека найдет суммарную цену всех покупок.

Входные данные

В первой строке содержится непустая строка *s* длины не более 1000 — содержимое чека Василия. Гарантируется, что чек задан в формате, описанном в условии. Гарантируется, что каждая цена в чеке составляет не менее одной копейки и не более 106 рублей.

Выходные данные

Выведите суммарную цену всех покупок **строго в том же формате**, в котором задаются цены покупок во входных данных.

Примеры

входные данные

chipsy48.32televizor12.390

выходные данные

12.438.32

входные данные

a1b2c3.38

выходные данные

6.38

входные данные

aa0.01t0.03

выходные данные

0.04

С. Восстановление массива

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Это интерактивная задача. Вам нужно использовать операцию `flush` после вывода каждого запроса. Например, в C++ вы должны использовать функцию `fflush(stdout)`, в Java — использовать `System.out.flush()`, а в Паскале — `flush(output)`.

В этой задаче вам надо восстановить массив, который заранее вам неизвестен. Вы можете считать, что жюри загадало некоторый массив a , про который вам известна только его длина n .

Единственное допустимое действие — узнать сумму пары элементов, указав их индексы i и j (индексы должны быть **различными**). В результате запроса для индексов i и j вы получите сумму $a_i + a_j$.

Известно, что восстановить весь загаданный массив можно не более чем за n запросов. Напишите программу, которая восстановит загаданный жюри массив a длины n за не более чем n запросов на сумму двух элементов (в каждом запросе индексы двух элементов должны быть различны).

Протокол взаимодействия

Каждый тест в этой задаче состоит из одного массива, который ваша программа должна восстановить.

В первой строке входных данных следует целое положительное число n ($3 \leq n \leq 5000$) — длина загаданного массива. В первую очередь ваша программа должна прочитать это число.

Далее ваша программа должна выводить в стандартный вывод запросы на сумму двух элементов массива, либо сообщить о том, что загаданный жюри массив уже найден.

- В случае, если программа осуществляет запрос на сумму, то следует вывести строку вида «? i j » (i и j — **различные** целые числа от 1 до n) — индексы элементов массива, сумму которых ваша программа запрашивает.
- В случае, если программа сообщает восстановленный массив, то следует вывести строку вида «! $a_1 a_2 \dots a_n$ » (гарантируется, что все a_i в правильно восстановленном массиве — положительные целые числа и не превосходят 105), где a_i равно числу, стоящему в массиве в позиции i .

Результатом запроса на сравнение является единственное целое число, равное $a_i + a_j$.

Для массива длины n ваша программа должна сделать не более n запросов на сумму. Обратите внимание, что вывод строки вида «! $a_1 a_2 \dots a_n$ » не считается запросом и не учитывается при подсчете их количества.

Не забывайте использовать операцию flush после каждой выведенной строки.

После вывода ответа ваша программа должна завершиться.

Пример

входные данные

9

7

9

11

6

выходные данные

? 1 5

? 2 3

? 4 1

? 5 2

? 3 4

! 4 6 1 5 5

Примечание

Вы можете взламывать, задавая тесты следующего вида:

- в первой строке должно быть записано целое число n ($3 \leq n \leq 5000$) — длина массива,

во второй строке должны быть записаны целые числа a_1, a_2, \dots, a_n ($1 \leq a_i \leq 105$) — элементы загаданного массива.

D. Распределение футболок

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

В качестве сувениров на соревновании по программированию было решено вручить футболки. Всего в типографии были напечатаны футболки шести размеров: S, M, L, XL, XXL, XXXL (размеры перечислены в порядке возрастания). Для каждого размера от S до XXXL вам известно количество футболок такого размера.

Во время регистрации организаторы попросили каждого из n участников указать размер футболки. Если участник колебался между двумя размерами, то он мог указать два соседних — это означает, что ему подойдет футболка любого из двух размеров.

Напишите программу, которая определит, возможно ли из напечатанных в типографии футболок сделать подарок каждому участнику соревнования. Конечно, каждому участнику должна достаться футболка его размера:

- требуемого размера, если указан один размер;
- любого из двух размеров, если указаны два соседних размера.

В случае положительного ответа программа должна найти любой из вариантов раздачи футболок.

Входные данные

В первой строке входных данных содержится шесть целых неотрицательных чисел — количество футболок с размерами S, M, L, XL, XXL, XXXL, соответственно. Суммарное количество напечатанных футболок не превосходит 100 000.

Во второй строке содержится целое положительное число n ($1 \leq n \leq 100\,000$) — количество участников соревнований.

В следующих n строках содержатся указания участников — в i -й из них содержатся размеры, указанные i -м участником. Если участник указал только один размер, строка совпадает с его названием. Если же участник указал два размера, то они заданы своими названиями через запятую без

разделения пробелами. В этом случае сначала записан меньший размер, затем — больший. Гарантируется, что в таком случае эти размеры соседние.

Выходные данные

Если ответа не существует выведите «NO» (без кавычек).

В противном случае, в первой строке выведите «YES» (без кавычек). В следующих n строках выведите размеры футболок, которые получают участники. Порядок участников должен совпадать с тем порядком, который задан во входных данных.

Если решений несколько, выведите любое.

Примеры

входные данные

0 1 0 1 1 0

3

XL

S,M

XL,XXL

выходные данные

YES

XL

M

XXL

входные данные

1 1 2 0 1 1

5

S

M

S,M

XXL,XXXL

XL,XXL

выходные данные

NO

Е. Игры на диске

ограничение по времени на тест: 4 секунды

ограничение по памяти на тест: 512 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

У Толи было n компьютерных игр, и он решил записать их на один диск. После этого он решил написать маркером названия всех своих игр на этом диске по кругу **по часовой стрелке** друг за другом. Названия всех игр были различные, а длина каждого названия была ровно k . Написанные названия на диске не перекрываются между собой.

После того, как Толя написал названия всех игр, на диске получилась циклическая строка длины $n \cdot k$.

Прошло несколько лет и Толя уже и забыл, какие игры записаны на его диске. Он помнит, что всего в то время было g популярных игр, и на его диске могут быть только лишь эти игры, причем каждая из g игр может быть записана на диске **не более одного раза**.

Перед вами стоит задача восстановить любой корректный список игр, которые Толя мог записать на свой диск.

Входные данные

В первой строке следует два целых положительных числа n и k ($1 \leq n \leq 105$, $1 \leq k \leq 105$) — количество игр, которые были у Толи и длина названий этих игр.

Во второй строке следует строка, состоящая из строчных букв латинского алфавита — строка, написанная на диске, разорванная в произвольном месте. Длина этой строки равна $n \cdot k$. Гарантируется, что длина строки не превосходит 106.

В третьей строке следует целое положительное число g ($n \leq g \leq 105$) — количество популярных игр, которые могут быть записаны на диске. Гарантируется, что суммарная длина всех названий популярных игр не превосходит $2 \cdot 106$.

В следующих g строках следует по одному названию популярных игр. Длина каждого из названий равна k . Строки состоят из строчных букв латинского алфавита. Гарантируется, что все названия популярных игр различны.

Выходные данные

Если ответа не существует, выведите «NO» (без кавычек).

В противном случае, выведите в первую строку «YES» (без кавычек). Во вторую строку выведите n целых чисел — номера популярных игр, записанных на диске Толи. Выводить игры нужно в порядке их записи на диске, то есть **по часовой стрелке**, при этом начинать вывод можно **с любой игры**. Помните, что каждая популярная игра могла быть записана на диск не более одного раза. Если ответов несколько, разрешается вывести любой из них.

Примеры

входные данные

abc

4

b

a

c

d

выходные данные

YES

2 1 3

входные данные

4 2

aabbccdd

4

dd

ab

bc

cd

выходные данные

NO

F. Задачи Поликарпа

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Поликарп — опытный участник соревнований по программированию Codehorses. Теперь он решил попробовать себя в качестве автора задач.

Он отослал координатору раундов набор из n задач. Каждая задача характеризуется своим качеством, качество i -й задачи равно a_i (a_i может быть положительно, отрицательно или равно нулю). Задачи отсортированы по предполагаемой сложности, которая никак не связана с качеством. Таким образом, самая простая задача имеет номер 1, а самая сложная — номер n .

В настоящий момент настроение координатора равно q . Известно, что после чтения очередной задачи его настроение изменяется на качество этой задачи, то есть после того, как координатор прочитает задачу с качеством b , к его настроению добавляется величина b . Координатор всегда читает задачи подряд от самой простой к самой сложной, порядок чтения задач изменять нельзя.

Если в какой-то момент текущее настроение координатора становится отрицательным, то он немедленно прекращает чтение и полностью отклоняет весь комплект задач.

Поликарп хочет выбросить минимальное количество задач так, чтобы настроение координатора всегда было неотрицательным. Так как Поликарп не знает точно текущего настроения координатора, то у него есть m гипотез вида «текущее настроение координатора $q = b_i$ ».

Для каждой из m гипотез найдите минимальное количество задач, которое надо удалить из комплекта, чтобы при чтении оставшихся задач от самой простой к самой сложной настроение координатора всегда было больше или равно 0.

Входные данные

В первой строке входных данных записаны два целых числа n и m

$(1 \leq n \leq 750, 1 \leq m \leq 200\,000)$ — количество задач в комплекте и количество гипотез о настроении координатора соответственно.

Во второй строке записаны n целых чисел a_1, a_2, \dots, a_n ($-109 \leq a_i \leq 109$) — качество задач в комплекте в порядке увеличения сложности.

В третьей строке содержится m целых чисел b_1, b_2, \dots, b_m ($0 \leq b_i \leq 1015$) — возможные значения текущего настроения координатора q .

Выходные данные

Выведите m строк, в i -й из них выведите единственное целое число — ответ на задачу для $q = b_i$.

Пример

входные данные

6 3

8 -5 -4 1 -7 4

0 7 3

выходные данные

2

0

1

2 отборочный раунд

А. Интервью с Олегом

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Поликарп взял у Олега интервью и записал его себе в блокнот без знаков препинания и пробелов, чтобы сэкономить время и успеть все записать. В итоге, интервью представляет собой строку s , состоящую из n строчных букв латинского алфавита.

В речи Олега есть слово-паразит `ogo`, а также все слова, которые получаются из слова `ogo` приписыванием справа к нему слога `go`. Например, слова `ogo`, `ogogo`, `ogogogo` являются паразитами, а слова `go`, `og`, `ogog`, `ogogog` и `oggo` — не являются.

Слова-паразиты имеют максимальный возможный размер, то есть, например, в речи `ogogoogoo` нельзя считать, что слово-паразит это `ogo`, а `goo` является частью обыкновенной фразы интервью. В данном случае словом-паразитом является подстрока `ogogo`.

До печати Поликарпу необходимо заменить каждое слово-паразит на последовательность из трёх звездочек. Обратите внимание, что независимо от длины слова-паразита оно заменяется ровно на три звёздочки.

Поликарп быстро справился с этой задачей. А сможете ли это сделать вы? Время пошло!

Входные данные

В первой строке следует целое положительное число n ($1 \leq n \leq 100$) — длина интервью.

Во второй строке следует строка s длины n , состоящая из строчных букв латинского алфавита.

Выходные данные

Выведите текст интервью после замены каждого слова-паразита на «***». Допустимо, что в ответе подстрока «***» будет идти подряд несколько раз.

Примеры

входные данные

7

aogogob

выходные данные

a***b

входные данные

13

ogogmgogogogo

выходные данные

gmg

входные данные

9

ogoogoogo

выходные данные

Примечание

В первом примере одно слово-паразит ogo, поэтому интервью для печати выглядит как «a***b».

Во втором примере два слова-паразита ogo и ogogogo, поэтому интервью без слов-паразитов выглядит как «***gmg***».

В. Прожекторы

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Театральная сцена представляет собой прямоугольное поле размером $n \times m$. Директор театра выдал вам план сцены, согласно которому на ней будут располагаться актёры. На плане отмечено в каких клетках будут стоять актёры, а в каких нет.

Прожектор, установленный на сцену, будет светить в одном из четырёх направлений (если смотреть на план сцены сверху) — влево, вверх, вправо или вниз. Таким образом, под позицией прожектора понимается клетка, в которую он установлен, а также направление, в котором он светит.

Перед вами стоит задача поставить на сцену прожектор в *хорошую* позицию. Позиция называется *хорошей*, если одновременно выполняются два условия:

- в соответствующей ей клетке нет актёра;
- в направлении, в котором светит прожектор, находится хотя бы один актёр.

Перед вами стоит задача посчитать количество *хороших* позиций для установки прожектора. Две позиции установки прожектора считаются различными, если отличаются клетки расположения прожектора, или направление, в котором он светит.

Входные данные

В первой строке следует два целых положительных числа n и m ($1 \leq n, m \leq 1000$) — количество строк и количество столбцов в плане.

В следующих n строках следует по m целых чисел, каждое равно либо 0, либо 1, — описание плана. Если очередное число равно 1, то в соответствующей клетке находится актёр, а если 0, то клетка останется пустой. Гарантируется, что в плане есть хотя бы один актёр.

Выходные данные

Выведите единственное целое число — количество *хороших* позиций для установки прожектора.

Примеры

входные данные

2 4

0 1 0 0

1 0 1 0

выходные данные

9

входные данные

4 4

0 0 0 0

1 0 0 1

0 1 1 0

0 1 0 0

выходные данные

20

Примечание

В первом примере *хорошими* позициями для установки прожектора являются:

- клетка (1, 1) и направление вправо;
- клетка (1, 1) и направление вниз;
- клетка (1, 3) и направление влево;
- клетка (1, 3) и направление вниз;
- клетка (1, 4) и направление влево;
- клетка (2, 2) и направление влево;
- клетка (2, 2) и направление вверх;

- клетка (2, 2) и направление вправо;
- клетка (2, 4) и направление влево.

Таким образом, в данном примере всего 9 хороших позиций.

С. Дорога до кинотеатра

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Вася находится в центре проката машин и хочет как можно скорее добраться до кинотеатра. Сеанс, на который он уже купил билет, начнётся через t минут. Считайте, что есть прямая дорога от центра проката машин до кинотеатра длиной s километров. Введём систему координат так, что центр проката машин находится в точке 0, а кинотеатр находится в точке s . Известно, что по пути от центра проката машин до кинотеатра есть k заправочных станций, причём на всех можно заливать неограниченное количество топлива совершенно бесплатно! Считайте, что операция залива топлива осуществляется мгновенно.

В центре проката есть n машин, i -я из которых характеризуется двумя числами c_i и v_i — стоимостью аренды машины и вместимостью её бака. Таким образом, на заправке нельзя заливать в машину топлива больше вместимости её бака v_i . В центре проката все машины изначально полностью заправлены.

Каждая из машин может ехать в одном из двух скоростных режимов: обычном и ускоренном. В обычном режиме машина проезжает 1

километр за 2 минуты, при этом тратит на это 1 литр топлива. В ускоренном режиме машина проезжает 1 километр за 1 минуту, при этом тратит на это 2 литра топлива. Режим может быть изменён в любой момент. Изменять режим разрешается неограниченное количество раз.

Перед вами стоит задача выбрать машину с минимальной стоимостью аренды, на которой Вася успеет добраться до кинотеатра до начала своего сеанса, то есть не позднее, чем через t минут. Считайте, что в центре проката все машины изначально полностью заправлены.

Входные данные

В первой строке записаны четыре целых положительных числа n , k , s и t ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq k \leq 2 \cdot 10^5$, $2 \leq s \leq 109$, $1 \leq t \leq 2 \cdot 10^9$) — количество машин в центре проката, количество заправок на пути до кинотеатра, позиция кинотеатра и время, оставшееся до начала сеанса.

В следующих n строках следуют по два целых положительных числа c_i и v_i ($1 \leq c_i, v_i \leq 109$) — стоимость аренды i -й машины и объём её топливного бака.

В следующей строке следуют k **различных** целых чисел g_1, g_2, \dots, g_k ($1 \leq g_i \leq s - 1$) — позиции заправок на пути до кинотеатра в произвольном порядке.

Выходные данные

Выведите минимальную стоимость аренды подходящей машины, то есть такой, что Вася успеет добраться до кинотеатра до начала сеанса (не позднее, чем через t минут). Если ни одна из машин не подойдёт Васе, выведите -1.

Примеры

входные данные

3 1 8 10

10 8

5 7

11 9

3

выходные данные

10

входные данные

2 2 10 18

10 4

20 6

5 3

выходные данные

20

Примечание

В первом примере Вася успеет доехать до кинотеатра вовремя на первой и третьей машине, но выгоднее поехать на первой, стоимость аренды которой равна 10, а объём топливного бака равен 8. Тогда до первой заправки Вася сможет доехать в ускоренном режиме за 3 минуты, потратив на это 6 литров топлива, а затем, заправив полный бак, он сможет проехать 2 километра в обычном режиме за 4 минуты, потратив на это 2 литра бензина. Оставшиеся 3 километра он проедет в ускоренном режиме за 3 минуты, потратив на это 6 литров бензина.

D. Морской бой

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Галя играет в одномерный морской бой на поле размера $1 \times n$. В этой игре на клеточном поле расположены a кораблей, каждый состоит из b последовательных клеток. При этом одна клетка не может являться частью более чем одного корабля, однако, корабли **могут соприкасаться**.

Гале неизвестно положение кораблей. Галя может делать выстрелы по клеткам, при этом после каждого выстрела ей сообщается, является эта клетка частью какого-нибудь корабля (в таком случае считается, что Галя «попала»), или нет (тогда Галя «промахнулась»).

Галя уже сделала k выстрелов и все из них были промахами.

Перед вами стоит задача определить минимальное количество позиций, выстрелив в которые, Галя гарантированно попадет хотя бы в один из кораблей.

Гарантируется, что существует хотя бы одна расстановка кораблей, удовлетворяющая описанным условиям.

Входные данные

В первой строке находятся четыре целых положительных числа n, a, b, k ($1 \leq n \leq 2 \cdot 10^5, 1 \leq a, b \leq n, 0 \leq k \leq n - 1$) — длина поля, количество кораблей на поле, длина каждого корабля и количество выстрелов, которые Галя уже произвела.

Во второй строке следует строка длины n , состоящая из нулей и единиц. Если i -й символ строки равен единице, то Галя уже стреляла в эту клетку, в противном случае, нет. Гарантируется, что в этой строке ровно k единиц.

Выходные данные

В первую строку выведите минимальное количество позиций, выстрелив в

которые, Галя гарантированно попадёт хотя бы в один из кораблей.

Во вторую строку выведите позиции, в которые Галя должна стрелять.

Каждая позиция должна быть выведена ровно один раз. Позиции разрешается выводить в произвольном порядке. Позиции пронумерованы с 1 до n , начиная с самой левой.

Если существует несколько решений, выведите любое из них.

Примеры

входные данные

5 1 2 1

00100

выходные данные

2

4 2

входные данные

13 3 2 3

1000000010001

выходные данные

2

7 11

Примечание

В первом примере известно, что остался один корабль длины два. Он может располагаться как слева от уже сделанного выстрела (символа «1»), так и справа. Таким образом, чтобы наверняка попасть в него, надо произвести два выстрела: один в левую часть, другой в правую.

Е. Подчинённые

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

В крупной компании работают n сотрудников, каждый из которых имеет уникальный номер от 1 до n . Из них **ровно** один сотрудник является главным, его номер равен s . Также известно, что все сотрудники, кроме главного, имеют ровно одного непосредственного начальника.

Каждому из сотрудников было поручено подать информацию о том, сколько начальников у него есть (не только непосредственных). Под начальниками сотрудника понимается его непосредственный начальник, а также непосредственный начальник непосредственного начальника данного сотрудника, и так далее. Например, если в компании три сотрудника, первый из которых главный, у второго сотрудника непосредственный начальник — первый сотрудник, а у третьего сотрудника непосредственный начальник второй сотрудник, то у третьего сотрудника всего два начальника — один непосредственный и один не непосредственный. Главный сотрудник является начальником всех сотрудников, кроме самого себя.

Некоторые из сотрудников поторопились, ошиблись в подсчете и подали неверную информацию. Перед вами стоит задача определить минимально возможное число сотрудников, которые могли допустить ошибку при подаче информации.

Входные данные

В первой строке следует два целых положительных числа n и s ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq s \leq n$) — количество сотрудников и номер главного

сотрудника.

Во второй строке следует n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq n - 1$), где a_i равно количеству начальников (не обязательно непосредственных) у сотрудника i , о котором он сообщил при подаче информации.

Выходные данные

Выведите минимально возможное число сотрудников, которые могли допустить ошибку при подаче информации.

Примеры

входные данные

3 2

2 0 2

выходные данные

1

входные данные

5 3

1 0 0 4 1

выходные данные

2

Примечание

В первом примере мог ошибиться, например, только первый сотрудник, в том случае, если:

- у первого сотрудника начальником является второй сотрудник,
- у третьего сотрудника начальником является первый сотрудник, второй сотрудник является главным.

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 512 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Эта задача имеет нестандартное ограничение по памяти.

Финансистам Игорю и Жене стало скучно вечером, и они решили сыграть в игру. Для этого они приготовили n ценных бумаг, в которых содержится информация о доходе предприятия за какие-то промежутки времени. Обратите внимание, что доход может быть и положительным, и нулевым, и даже отрицательным.

Игорь и Женя выложили все бумаги в ряд и решили ходить по очереди. Игорь будет брать бумаги слева, а Женя справа. Первым ходит Игорь и берет 1 или 2 по своему выбору ценные бумаги слева. Далее, во время очередного хода игрок может взять k или $k + 1$ бумагу со своей стороны, если игрок, ходивший перед ним, взял ровно k бумаг. Ход пропускать не может ни один из игроков. Игра заканчивается, когда закончатся бумаги на столе, либо когда игрок не сможет сделать ход.

Перед вами стоит задача определить разность между суммой доходов бумаг, которые забрал себе Игорь, и суммой доходов бумаг, которые забрал себе Женя, если оба игрока играют оптимально. Игорь хочет максимизировать разность, а Женя — минимизировать.

Входные данные

В первой строке следует целое положительное число n ($1 \leq n \leq 4000$) — количество бумаг.

Во второй строке следует последовательность из n целых чисел a_1, a_2, \dots, a_n ($-105 \leq a_i \leq 105$), где a_i равно доходу, информация о котором

содержится на i -й бумаге.

Выходные данные

Выведите разность между суммой доходов бумаг, которые забрал себе Игорь, и суммой доходов бумаг, которые забрал себе Женя, если оба игрока играют оптимально. Игорь хочет максимизировать разность, а Женя — минимизировать.

Примеры

входные данные

3

1 3 1

выходные данные

4

входные данные

5

-1 -2 -1 -2 -1

выходные данные

0

входные данные

4

-4 -2 4 5

выходные данные

-13

Примечание

В первом примере Игорю выгодно забрать две бумаги слева, получив суммарный доход 4, тогда Женя не сможет сделать ход, так как осталась всего одна бумага, а он может взять либо 2, либо 3 на текущем ходе.

3 отборочный раунд

А. Санта-Клаус и место в классе

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Санта-Клаус пришел на рождественскую олимпиаду раньше всех, и ему предстоит первым занять свое место за партой! В аудитории, где находится его место, находятся n рядов по m парт, при этом за каждой партой находятся два рабочих места. Ряды пронумерованы слева направо от 1 до n , а парты в ряду — 1 до m начиная от доски. Обратите внимание, что в отличие от театров и кинотеатров ряды идут не параллельно экрану/доске, а перпендикулярно (см. рисунок).

Организаторы рождественской олимпиады пронумеровали все рабочие места числами от 1 до $2nm$. Места пронумерованы по рядам (т. е. сначала пронумерованы все места первого ряда, затем — все места второго и так далее), в одном ряду места пронумерованы начиная от доски (т. е. с первой парты этого ряда), за одной партой сначала пронумеровано место слева, затем — место справа.

		ряды							
		1		2		3		4	
п а р т ы	1	1	2	7	8	13	14	19	20
	2	3	4	9	10	15	16	21	22
	3	5	6	11	12	17	18	23	24

Рисунок иллюстрирует первый и второй примеры входных/выходных данных.

Санта-Клаус знает, что его место имеет номер k . Помогите ему узнать, в каком ряду за какую парту ему нужно сесть, а также слева его место или справа!

Входные данные

В единственной строке находятся три целых числа n , m и k ($1 \leq n, m \leq 10\,000$, $1 \leq k \leq 2nm$) — число рядов и число парт в каждом ряду, а так же номер места Санта-Клауса.

Выходные данные

Выведите два целых числа: номер ряда r , номер парты d , а также символ s , означающий сторону парты, за которую нужно сесть Санта-Клаусу. Символ s должен быть «L», если ему нужно сесть за левую сторону, и «R», если место Санта-Клауса справа.

Примеры

входные данные

4 3 9

выходные данные

2 2 L

входные данные

4 3 24

выходные данные

4 3 R

входные данные

2 4 4

выходные данные

1 2 R

Примечание

Первый и второй примеры показаны на картинке. Зеленым выделено место Санта-Клауса в первом примере, а голубым — его место во втором примере.

В третьем примере два ряда по четыре парты, а место Санта-Клауса — четвертое. Значит, его место в первом ряду за второй партой справа.

В. Санта-Клаус и проверка клавиатуры

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Санта-Клаус решил разобрать свою клавиатуру, чтобы её почистить. После того, как он поставил все клавиши обратно, он с ужасом понял, что что-то не так: некоторые пары клавиш перепутаны между собой! Таким образом, Санта-Клаус подозревает, что каждая клавиша либо стоит на своём месте, либо заняла место другой, а та другая — на месте первой.

Для того, чтобы убедиться в этом, найти ошибку и восстановить верное расположение, Санта-Клаус набрал текст своей любимой скороговорки,

смотря только на надписи на клавиатуре.

Вам даны любимая скороговорка Санта-Клауса и строка, которая получилась в результате набора. Определите, какие пары клавиш Санта-Клаус мог перепутать. Каждая клавиша должна принадлежать **не более чем одной паре** перепутанных клавиш.

Входные данные

Входные данные состоят из двух строк s и t — любимой скороговорки Санта-Клауса и строки, которая получилась после набора скороговорки. Строки s и t непусты и имеют одинаковую длину, которая не превышает 1000, включительно. Обе строки состоят только из строчных латинских букв.

Выходные данные

Если предположение Санта-Клауса неверно и клавиатура требует починки и её нельзя починить, поменяв местами буквы в нескольких непересекающихся парах, выведите одно число «-1» (без кавычек).

Иначе в первой строке выведите число k ($k \geq 0$) — количество пар букв, которые нужно поменять местами. Затем в следующих k строках выведите по две буквы, разделённые пробелом — буквы, которые необходимо поменять местами на клавиатуре. Все выведенные буквы должны быть различны.

Если ответов несколько, выведите любой. Как пары, так и буквы в парах можно выводить в любом порядке.

Каждая буква должна присутствовать не более чем в одной паре. Санта-Клаус считает, что клавиши расположены корректно, если он может набрать на клавиатуре текст своей любимой скороговорки без ошибок.

Примеры

входные данные

helloworld

ehoolwlroz

выходные данные

3

h e

l o

d z

входные данные

hastalavistababy

hastalavistababy

выходные данные

0

входные данные

merrychristmas

christmasmerry

выходные данные

-1

С. Санта-Клаус и его Робот

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

У Санта-Клауса есть Робот, который живёт на клетчатой плоскости и умеет перемещаться **по линиям сетки**. Если ему дать последовательность из m

точек p_1, p_2, \dots, p_m с целыми координатами, то он сделает следующее: обозначим точку, в которой он сейчас находится, через p_0 . Тогда Робот сначала поедет по некоторому кратчайшему пути из p_0 в p_1 (обратите внимание, поскольку Робот ездит только по линиям сетки, кратчайших путей может быть несколько), затем, доехав до p_1 , поедет к точке p_2 , опять же, по какому-то кратчайшему пути, затем к точке p_3 , и так далее, пока не пройдет все точки в заданном порядке. Некоторые из точек в последовательности могут совпадать, тогда Санта-Клаус должен посетить их несколько раз в порядке, соответствующем последовательности.

Пока Санта-Клауса не было, кто-то дал Роботу несколько точек. Эта последовательность точек была утерян, но у вас есть протокол перемещений Робота (каждое перемещение на единицу длины). Узнайте, пожалуйста, минимальную возможную длину последовательности, заданной Роботу.

Входные данные

В первой строке задано единственное натуральное число n ($1 \leq n \leq 2 \cdot 10^5$) — число перемещений Робота на единичный отрезок. Во второй строке дан протокол перемещений Робота в виде n символов, каждый из которых равен L, R, U или D, записанных без пробелов. k -й символ означает, что на k -м шаге Робот переместился на единицу длины в направлении, соответствующем этому символу: L означает, что он двигался влево, R — вправо, U — вверх и D — вниз. Смотрите иллюстрации к примерам для большего понимания.

Выходные данные

В единственной строке выведите минимальную возможную длину последовательности, заданной Роботу.

Примеры

входные данные

4

RURD

выходные данные

2

входные данные

6

RRULDD

выходные данные

2

входные данные

26

RRRULURURUULULLLDLDDRDRDL

выходные данные

7

входные данные

3

RLL

выходные данные

2

входные данные

4

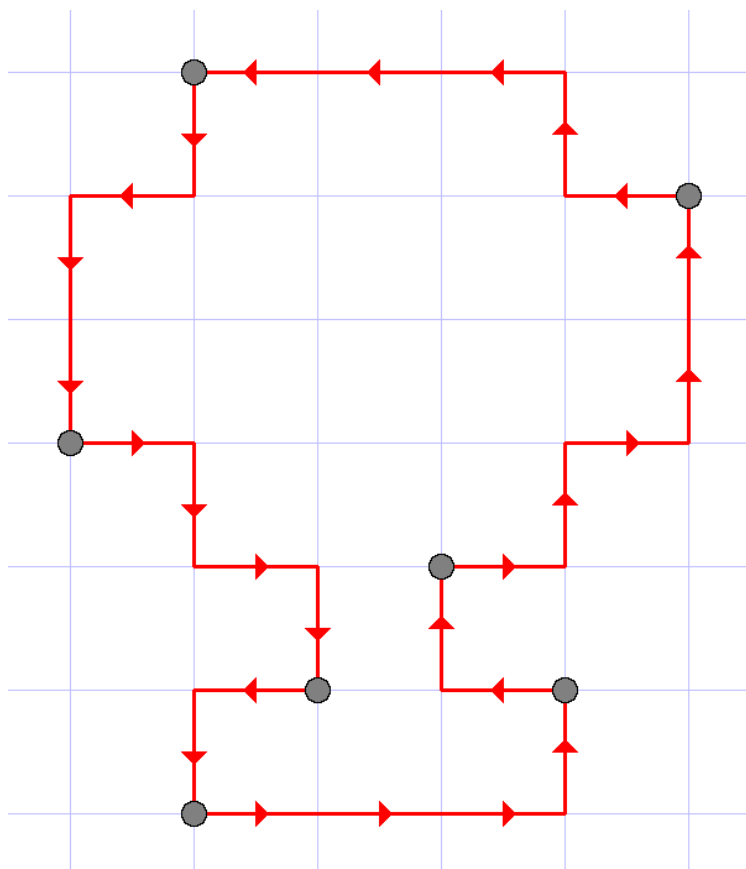
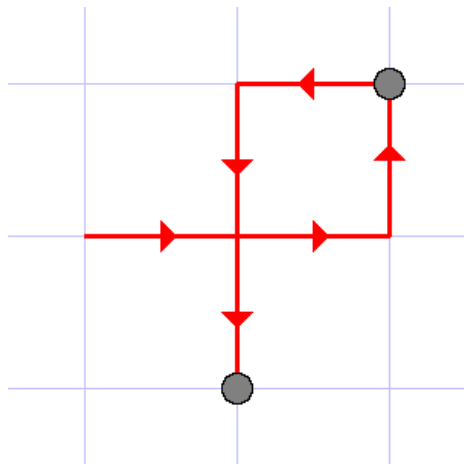
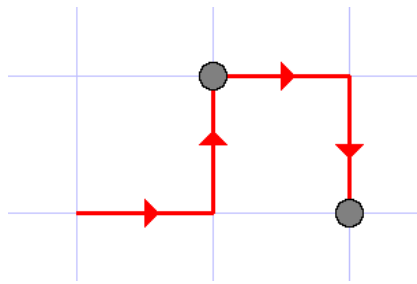
LRLR

выходные данные

4

Примечание

Ниже приведены иллюстрации к первым трём тестам.



Последний пример показывает, что каждая точка в последовательности должна быть посчитана столько раз, сколько она в ней встречается.

D. Санта-Клаус и палиндром

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Санта-Клаус очень любит палиндромы. Недавно у него был день рождения. В гости к Санта-Клаусу пришли k друзей, каждый из них подарил ему строку s_i одной и той же длины n , красота i -й строки равна a_i . Возможно, что a_i является отрицательным — это значит, что строка не является красивой по мнению Санта-Клауса.

Санта-Клаус без ума от палиндромов. Ему стало интересно: какую максимальную суммарную красоту может иметь палиндром, если склеить какие-то (возможно все) из подаренных строк? Каждый подарок можно использовать не более одного раза. Обратите внимание, что все подаренные строки имеют **одинаковую длину n** .

Напоминаем, что палиндром — это строка, которая не изменится, если её развернуть задом наперёд.

Так как пустая строка является палиндромом, то искомая максимальная красота — неотрицательна. Даже если все a_i отрицательны, Санта-Клаус может получить пустую строку в качестве палиндрома.

Входные данные

В первой строке задано два целых числа через пробел, k и n — количество друзей Санта-Клауса и длина строки, подаренной каждым другом ($1 \leq k, n \leq 100\,000$; $n \cdot k \leq 100\,000$).

Далее следуют k строк. i -я из них содержит подаренную строку s_i и её

красоту ai ($-10\,000 \leq ai \leq 10\,000$). Строка состоит из n строчных букв латинского алфавита, а её красота — целое число. Подаренные строки могут совпадать. Одинаковые строки могут иметь разную красоту.

Выходные данные

Выведите искомую максимальную суммарную красоту.

Примеры

входные данные

7 3

abb 2

aaa -3

bba -1

zyz -4

abb 5

aaa 7

хух 4

выходные данные

12

входные данные

3 1

a 1

a 2

a 3

выходные данные

6

входные данные

2 5

abcde 10000

abcde 10000

выходные данные

0

Примечание

В первом примере из условия Санта-Клаус может склеить палиндром `abbaaaxухааабba`, используя строки 5, 2, 7, 6, 3 (склеив их именно в этом порядке).

Е. Санта-Клаус и мандарины

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

У Санта-Клауса есть n мандаринов, причём i -й из них имеет a_i долек. Санта-Клаус пришел в школу, где учится k учеников. Он решил угостить их мандаринами.

Так как на всех мандаринов может не хватить, Санта-Клаус решил поделить их на части, чтобы никто не обиделся. Для этого он может делить мандарины пополам, а также делить любую часть пополам. Если количество долек в мандарине или в части, которую Санта-Клаус делит, нечётное, то в одной получившейся части окажется на одну дольку больше, чем в другой. Делить мандарин или часть мандарина можно лишь в том случае, если после деления каждая получившаяся часть будет иметь хотя бы одну дольку.

Санта-Клаус хочет дать каждому из школьников ровно один мандарин

или одну часть мандарина (в частности, каждый ученик должен получить положительное количество долек). Возможно, что у Санта-Клауса останется несколько мандаринов или частей после того, как он раздаст часть из них школьникам.

Пусть в результате угощения i -му ученику достанется b_i долек. В таком случае *радость* Санта-Клауса будет равна минимальному значению среди всех b_i .

Найдите максимальную возможную величину *радости* Санта-Клауса, которую он может получить после угощения учеников мандаринами.

Входные данные

В первой строке находятся два целых положительных числа n и k ($1 \leq n \leq 106$, $1 \leq k \leq 2 \cdot 109$) — количество мандаринов и количество учеников.

Во второй строке находится последовательность из n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 107$), где a_i равно количеству долек в i -м мандарине.

Выходные данные

Если невозможно раздать всем ученикам по мандарину или части мандарина, выведите -1. В противном случае выведите максимально возможную величину *радости* Санта-Клауса.

Примеры

входные данные

3 2

5 9 3

выходные данные

5

входные данные

2 4

12 14

выходные данные

6

входные данные

2 3

1 1

выходные данные

-1

Примечание

В первом примере Санта-Клаусу нужно разделить второй мандарин пополам, чтобы в одной части было 5 долек, а в другой 4. Тогда он сможет отдать одному ученику часть, в которой 5 долек, а второму целый первый мандарин, в котором также 5 долек.

Во втором примере Санта-Клаусу нужно разделить пополам оба мандарина, тогда он сможет отдать двум ученикам части по 6 долек, а двум другим ученикам — части по 7 долек.

В третьем примере Санта-Клаус не сможет дать всем ученикам хотя бы по одной дольке, так как у него есть всего 2 дольки и 3 ученика.

F. Санта-Клаусы и чемпионат по футболу

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

В Древляндии n городов, соединенных между собой $n - 1$ дорогой так, что

из каждого города можно по дорогам добраться до любого другого. В следующем году в Древляндии пройдет чемпионат по футболу среди команд, состоящих из Санта-Клаусов. Всего в чемпионате примут участие $2k$ команд из $2k$ различных городов.

На первом этапе команды разобьются на k пар, каждая пара сыграет два матча: один в городе первой команды из пары, другой — в городе второй команды из пары. Таким образом, в каждом из $2k$ городов участников будет проведен ровно один матч. Однако как разбить команды на пары, пока не решено.

Перед организаторами также стоит задача определить, в каких городах расселить участников на время чемпионата. Организаторы хотят поселить команды в как можно **меньшее** число городов, чтобы Санта-Клаусы побольше общались друг с другом и обменивались опытом.

Никакая из команд не хочет чересчур много передвигаться во время турнира, поэтому если команда будет играть в городах u и v , то жить она хочет в городе, лежащем на кратчайшем пути из u в v (в том числе, возможно, в городе u или городе v). Также команды из одной пары необходимо расселить в одном городе.

Таким образом, организаторы турнира хотят разбить $2k$ команд на пары и расселить участников по минимально возможному числу m городов так, чтобы команды из одной пары жили в одном городе, и чтобы для каждой команды город, в котором она будет жить во время турнира, лежал на кратчайшем пути между городами, в которых она будет играть.

Входные данные

В первой строке заданы два целых числа n и k ($2 \leq n \leq 2 \cdot 10^5$, $2 \leq 2k \leq n$) — количество городов в Древляндии и количество пар команд в турнире.

Следующие $n - 1$ строк задают описание дорог в Древляндии, каждая из

них содержит два целых числа a, b ($1 \leq a, b \leq n, a \neq b$), что значит, что города a и b соединены очередной дорогой. Гарантируется, что из любого города можно добраться в любой другой по дорогам.

Следующая строка содержит $2k$ различных целых чисел c_1, c_2, \dots, c_{2k} ($1 \leq c_i \leq n$), где c_i — город, из которого команда номер i . Все эти числа различны.

Выходные данные

В первой строке выведите целое число m — минимальное количество городов, по которым можно расселить участников.

Во второй строке выведите m различных чисел d_1, d_2, \dots, d_m ($1 \leq d_i \leq n$) — номера городов, в которых будут жить участники.

Далее выведите k строк. В j -й из них выведите 3 числа u_j, v_j, x_j , где u_j и v_j — города, в которых будет играть j -я пара, а x_j — номер города, в котором будут жить команды этой пары. Каждое из чисел c_1, c_2, \dots, c_{2k} должно встретиться среди чисел u_j и v_j ровно один раз. Каждое из чисел x_j должно принадлежать множеству $\{d_1, d_2, \dots, d_m\}$.

Если оптимальных ответов несколько, выведите любой из них.

Пример

входные данные

6 2

1 2

1 3

2 4

2 5

3 6

2 5 4 6

выходные данные

1

2

5 4 2

6 2 2

Примечание

В первом тесте из условия можно расселить всех участников в один город с номером 2. Разбить на пары можно любым образом, при этом все условия всегда будут выполнены, т. к. город 2 лежит на кратчайшем пути между любой парой городов из множества {2, 4, 5, 6}.