<https://yandex.ru/jobs/pages/hrtech_backend?utm_source=recruitment&utm_medium=referral&utm_campaign=hrtech>

<https://yandex.ru/jobs/pages/dev_interview>

<https://coderun.yandex.ru/selections?group=yainterview>

<https://contest.yandex.ru/contest/8458/enter?utm_source=career_portal>

1. Словарь синонимов

Вам дан словарь, состоящий из пар слов. Каждое слово является синонимом к парному ему слову. Все слова в словаре различны. Для одного данного слова определите его синоним.

## Формат ввода

Программа получает на вход количество пар синонимов N*N*. Далее следует N*N* строк, каждая строка содержит ровно два слова-синонима. После этого следует одно слово.

## Формат вывода

Программа должна вывести синоним к данному слову.

## Примечание

Эту задачу можно решить и без словарей (сохранив все входные данные в списке), но решение со словарем будет более простым.

## Ограничения

Ограничение времени

1 с

Ограничение памяти

256 МБ

## Пример 1

Ввод

3

Hello Hi

Bye Goodbye

List Array

Goodbye

Вывод

Bye

## Пример 2

Ввод

1

beep Car

Car

Вывод

beep

## Пример 3

Ввод

2

Ololo Ololo

Numbers 1234567890

Numbers

Вывод

1234567890

1. Ближайшее число

Напишите программу, которая находит в массиве элемент, самый близкий по величине к данному числу.

## Формат ввода

В первой строке задается одно натуральное число N*N*, не превосходящее 1000 — размер массива. Во второй строке содержатся N*N* чисел — элементы массива, целые числа, не превосходящие по модулю 1000. В третьей строке вводится одно целое число x*x*, не превосходящее по модулю 1000.

## Формат вывода

Вывести значение элемента массива, ближайшее к x*x*. Если таких чисел несколько, выведите любое из них.

## Ограничения

Ограничение времени

1 с

Ограничение памяти

64 МБ

## Пример 1

Ввод

5

1 2 3 4 5

6

Вывод

5

## Пример 2

Ввод

5

5 4 3 2 1

3

Вывод

3

3. Полиглоты

Каждый из N*N* школьников некоторой школы знает Mi*Mi*​ языков. Определите, какие языки знают все школьники и языки, которые знает хотя бы один из школьников.

## Формат ввода

Первая строка входных данных содержит количество школьников N*N*. Далее идет N*N* чисел Mi*Mi*​, после каждого из чисел идет Mi*Mi*​ строк, содержащих названия языков, которые знает i*i*-й школьник. Длина названий языков не превышает 1000 символов, количество различных языков не более 1000. 1≤N≤10001≤*N*≤1000, 1≤Mi≤5001≤*Mi*​≤500.

## Формат вывода

В первой строке выведите количество языков, которые знают все школьники. Начиная со второй строки - список таких языков. Затем - количество языков, которые знает хотя бы один школьник, на следующих строках - список таких языков.

## Ограничения

Ограничение времени

1 с

Ограничение памяти

64 МБ

## Пример 1

Ввод

3

3

Russian

English

Japanese

2

Russian

English

1

English

Вывод

1

English

3

Russian

Japanese

English

4. Наибольшее произведение двух чисел

Дан список, заполненный произвольными целыми числами. Найдите в этом списке два числа, произведение которых максимально.

## Формат ввода

В единственной строке через пробел вводятся целые числа — элементы списка. Список содержит не менее двух и не более 100 000 чисел. Сами элементы по модулю не превышают 1 000 000.

## Формат вывода

Выведите эти два числа в порядке неубывания.

## Примечание

Решение должно иметь сложность O(n)O(*n*), где n*n* — размер списка.

Гарантируется, что во всех тестах ответ однозначен.

## Ограничения

Ограничение времени

1 с

Ограничение памяти

64 МБ

## Пример 1

Ввод

4 3 5 2 5

Вывод

5 5

## Пример 2

Ввод

-4 3 -5 2 5

Вывод

-5 -4

## Пример 3

Ввод

12288 -10075 29710 15686 -18900 -17715 15992 24431 6220 28403 -23148 18480 -22905 5411 -7602 15560 -26674 11109 -4323 6146 -1523 4312 10666 -15343 -17679 7284 20709 -7103 24305 14334 -12281 17314 26061 25616 17453 16618 -24230 -19788 21172 11339 2202 -22442 -20997 1879 -8773 -8736 5310 -23372 12621 -25596 -28609 -13309 -13 10336 15812 -21193 21576 -1897 -12311 -6988 -25143 -3501 23231 26610 12618 25834 -29140 21011 23427 1494 15215 23013 -15739 8325 5359 -12932 18111 -72 -12509 20116 24390 1920 17487 25536 24934 -6784 -16417 -2222 -16569 -25594 4491 14249 -28927 27281 3297 5998 6259 4577 12415 3779 -8856 3994 19941 11047 2866 -24443 -17299 -9556 12244 6376 -13694 -14647 -22225 21872 7543 -6935 17736 -2464 9390 1133 18202 -9733 -26011 13474 29793 -26628 -26124 27776 970 14277 -23213 775 -9318 29014 -5645 -27027 -21822 -17450 -5 -655 22807 -20981 16310 27605 -18393 914 7323 599 -12503 -28684 5835 -5627 25891 -11801 21243 -21506 22542 -5097 8115 178 10427 25808 10836 -11213 18488 21293 14652 12260 42 21034 8396 -27956 13670 -296 -757 18076 -15597 4135 -25222 -19603 8007 6012 2704 28935 16188 -20848 13502 -11950 -24466 5440 26348 27378 7990 -11523 -26393

Вывод

29710 29793

5. Правильная скобочная последовательность

Рассмотрим последовательность, состоящую из круглых, квадратных и фигурных скобок. Программа дожна определить, является ли данная скобочная последовательность правильной.

* Пустая последовательность явлется правильной.
* Если A — правильная последовательность, то последовательности (A), [A], {A} — правильные.
* Если A и B — правильные последовательности, то последовательность AB — правильная.

## Формат ввода

В единственной строке записана скобочная последовательность, содержащая не более 100000 скобок.

## Формат вывода

Если данная последовательность правильная, то программа должна вывести строку yes, иначе строку no.

## Ограничения

Ограничение времени

1 с

Ограничение памяти

64 МБ

## Пример 1

Ввод

()[]

Вывод

yes

## Пример 2

Ввод

([)]

Вывод

no

## Пример 3

Ввод

(

Вывод

no

6. Туризм

Горная цепь состоит из отрезков, соединяющих точки на плоскости, каждая из которых находится правее предыдущей (координата x*x* строго возрастает).

Трассой на горной цепи называется маршрут по отрезкам цепи из одной точки в другую. Трассы ориентированы — то есть маршрут может быть как слева направо, так и справа налево.

Подъёмом на трассе называется такой отрезок, на котором высота (координата y*y*) возрастает, а высотой подъёма — разность высот между начальной и конечной точками отрезка.

Вам даны несколько трасс на одной горной цепи. Помогите для каждой трассы определить суммарную высоту подъёмов.

## Формат ввода

В первой строке содержится число N*N* (1≤N≤300001≤*N*≤30000) — количество точек в цепи, задающей горную цепь. Далее в N*N* строках содержатся координаты точек — два целых числа xi*xi*​ и yi*yi*​ ​(1≤xi,yi≤300001≤*xi*​,*yi*​≤30000, xi≤xi+1*xi*​≤*xi*+1​).

В следующей строке находится число M*M* — количество трасс (1≤M≤300001≤*M*≤30000).

Далее в M*M* строках содержатся описания трасс. Каждое описание представляет собой два целых числа, si*si*​ и fi*fi*​ (1≤si,fi,≤N1≤*si*​,*fi*​,≤*N*) — номера точек начала и конца трассы. Начало и конец трассы могут совпадать. Точки нумеруются с единицы.

## Формат вывода

Для каждой трассы выведите одно число — суммарную высоту подъемов на данной трассе.

## Пример 1

Ввод

7

2 1

4 5

7 4

8 2

9 6

11 3

15 3

1

2 6

Вывод

4

## Пример 2

Ввод

6

1 1

3 2

5 6

7 2

10 4

11 1

3

5 6

1 4

4 2

Вывод

0

5

4

1. Сумма номеров

Вася очень любит везде искать своё счастливое число K*K*. Каждый день он ходит в школу по улице, вдоль которой припарковано N*N* машин. Он заинтересовался вопросом, сколько существует отрезков из подряд идущих машин таких, что сумма их номеров равна K*K*. Помогите Васе узнать ответ на его вопрос.

## Формат ввода

В первой строке входных данных задаются числа N*N* и K*K* (1≤N≤1000001≤*N*≤100000, 1≤K≤1091≤*K*≤109).

Во второй строке содержится N*N* чисел, задающих номера машин. Номера машин могут принимать значения от 11 до 999999 включительно.

## Формат вывода

Необходимо вывести одно число — количество отрезков.

## Примечание

В первом примере существуют четыре подходящих отрезка машин:

* только первая машина — 17,
* машины 2 и 3 — 7, 10,
* машины 3 и 4 — 10, 7,
* машины 4 и 5 — 7, 10.

## Ограничения

Ограничение времени

1 с

Ограничение памяти

64 МБ

## Пример 1

Ввод

5 17

17 7 10 7 10

Вывод

4

## Пример 2

Ввод

5 10

1 2 3 4 1

Вывод

2

8. Поиск в глубину

Дан неориентированный граф, возможно с петлями и кратными ребрами. Необходимо найти компоненту связности, содержащую вершину с номером 1.

## Формат ввода

В первой строке записаны два целых числа N*N* (1≤N≤1031≤*N*≤103) и M*M* (0≤M≤5×1050≤*M*≤5×105) — количество вершин и ребер в графе.

В последующих M*M* строках перечислены ребра — пары чисел, определяющие номера вершин, которые соединяют ребра.

Вершины нумеруются с единицы.

## Формат вывода

В первой строке выведите число K*K* — количество вершин в компоненте связности.

Во второй строке выведите K*K* целых чисел — вершины компоненты связности, перечисленные в порядке **возрастания номеров**.

## Примечание

**Петля в графе** — это ребро, которое соединяет вершину с самой собой.

**Кратные рёбра в графе** — это рёбра, которые соединяют одну и ту же пару вершин.

**Компонента связности в неориентированном графе** — это подмножество вершин таких, что все вершины достижимы друг из друга.

## Ограничения

Ограничение времени

2 с

Ограничение памяти

256 МБ

## Пример 1

Ввод

4 5

2 2

3 4

2 3

1 3

2 4

Вывод

4

1 2 3 4

9. Длина кратчайшего пути

Дан неориентированный граф. Найдите длину минимального пути между двумя вершинами.

## Формат ввода

В первой строке записано целое число N*N* (1≤N≤1001≤*N*≤100) – количество вершин в графе.

Далее записывается матрица смежности — N*N* строк, в каждой из которых содержится N*N* чисел 0 или 1, разделённых пробелом. Число 0 означает отсутствие ребра, а 1 — наличие ребра.

В последней строке задаются номера двух вершин — начальной и конечной.

Вершины нумеруются с единицы.

## Формат вывода

Выведите длину кратчайшего пути — минимальное количество ребер, которые нужно пройти.

Если пути нет, нужно вывести -1.

## Ограничения

Ограничение времени

1 с

Ограничение памяти

64 МБ

## Пример 1

Ввод

10

0 1 0 0 0 0 0 0 0 0

1 0 0 1 1 0 1 0 0 0

0 0 0 0 1 0 0 0 1 0

0 1 0 0 0 0 1 0 0 0

0 1 1 0 0 0 0 0 0 1

0 0 0 0 0 0 1 0 0 1

0 1 0 1 0 1 0 0 0 0

0 0 0 0 0 0 0 0 1 0

0 0 1 0 0 0 0 1 0 0

0 0 0 0 1 1 0 0 0 0

5 4

Вывод

2

## Пример 2

Ввод

5

0 1 0 0 1

1 0 1 0 0

0 1 0 0 0

0 0 0 0 0

1 0 0 0 0

3 5

Вывод

3

10. Высота дерева

Реализуйте бинарное дерево поиска для целых чисел.

Программа должна последовательно обрабатывать вводимые числа. Если очередное число есть в дереве, ничего делать не нужно. Если числа в дереве нет, нужно добавить его в соответствующее место дерева.

Балансировку дерева производить не нужно.

Найдите высоту получившегося дерева.

## Формат ввода

На вход программа получает последовательность натуральных чисел. Последовательность завершается числом 0, которое означает конец ввода. Добавлять его в дерево не надо.

## Формат вывода

Выведите высоту получившегося дерева.

## Ограничения

Ограничение времени

2 с

Ограничение памяти

64 МБ

## Пример 1

Ввод

7 3 2 1 9 5 4 6 8 0

Вывод

4

11. Минимум на отрезке

Рассмотрим последовательность целых чисел длины n*n*.

По ней двигается «окно» длины k*k*: сначала в «окне» находятся первые k*k* чисел, на следующем шаге в «окне» уже будут находиться k*k* чисел, начиная со второго, и так далее до конца последовательности.

Требуется для каждого положения «окна» определить минимум в нём.

## Формат ввода

В первой строке входных данных содержатся два натуральных числа n*n* и k*k* (n≤150000*n*≤150000, k≤10000*k*≤10000, k≤n*k*≤*n*) — длины последовательности и «окна», соответственно.

На следующей строке находятся n*n* целых чисел — сама последовательность.

## Формат вывода

Выведите n−k+1*n*−*k*+1 строк. В каждой строке должно быть одно число — минимум для соответствующего положения «окна».

## Ограничения

Ограничение времени

1 с

Ограничение памяти

64 МБ

## Пример 1

Ввод

7 3

1 3 2 4 5 3 1

Вывод

1

2

2

3

1