

**А. Остатки сладки**  
**Ограничение времени: 1 с**  
**Ограничение памяти: 64МБ**

Василий увлёкся одной интересной задачей, которую мы здесь не озвучиваем. При решении этой задачи он разработал алгоритм с применением динамического программирования. Но оказалось, что для выполнения компьютерной программы на его компьютере не хватает памяти.

Василий изменил код программы так, что теперь он прогоняет входные данные дважды и получает не одно число в диапазоне от 0 до 999985999948, а два – остатки от деления искомого числа на 999983 и 1000003.

На прошлом соревновании по программированию некоторые команды-участницы написали для него программы, которые по имеющимся остаткам достаточно быстро восстанавливают искомый ответ – число в диапазоне от 0 до 999985999948.

Теперь Василий решил расширить диапазон искомых величин и получает остатки от деления на целых 4 числа: 12263, 12269, 12253 и 10000000.

Помогите Василию в его исследовании, напишите программу, которая по имеющимся четырём остаткам будем не более чем за 1 секунду восстанавливать искомый ответ.

**Входные данные**

Через пробел четыре целых неотрицательных числа – остатки от деления ответа соответственно на 12263, 12269, 10000000 и 12253.

**Выходные данные**

Выведите одно целое число из диапазона от 0 до 1843522014990999999 – искомый ответ. Если ответов несколько, выведите любой, если искомого числа не существует – выведите «does not exist» (без кавычек).

**Примеры**

Входные данные	Выходные данные
10 10 10 10	10
0 12251 12251 12251	14280042094430012251
6034 1298 1269 1269	122530001269

## **В. Комплексный обед**

**Ограничение времени:** 1 с

**Ограничение памяти:** 64МБ

У Василия есть друг Пётр, который учится в классе с математическим уклоном. Как-то в столовой они взяли по комплексному обеду и Пётр сказал: «А знаешь, Вася, комплексным может быть не только обед».

Пётр рассказал, что сейчас они начали изучать комплексные числа, и он ещё мало чего знает про них, но они удивительны: оказывается, множество задач с их применением может быть решено гораздо проще. Их изучением занимались такие великие люди как Джероламо Кардано, Рафаэль Бомбелли, Рене Декарт, Вильгельм Лейбниц, Абрахам де Муавр, Роджер Котс, Леонард Эйлер, Альбер Жирар, Жан Ленор Д'Аламбер, Иоган Карл Фридрих Гаусс, Лазар Николя Маргерит Карно, Каспар Вессель, Жан Робер Арган, Уильям Роуэн Гамильтон, Огюстен Луи Коши, Чарльз Протеус Штейнмец и др.

Пётр указал, что комплексное число состоит из двух компонентов: вещественной части  $a$  и мнимой части  $b$ , а само число выглядит как  $a+bi$ . Пётр попросил Василия написать для него программу (или попросить других программистов сделать это на ближайшем соревновании), которая могла бы выполнять четыре основные арифметические операции над ними, и привёл следующие формулы сложения, вычитания, умножения и деления (по одной на каждой строке в том же порядке, как они перечислены в предложении):

$$(a+bi) + (c+di) = (a+c) + (b+d)i$$

$$(a+bi) - (c+di) = (a-c) + (b-d)i$$

$$(a+bi) \times (c+di) = (ac-bd) + (bc+ad)i$$

$$(a+bi) \div (c+di) = (ac+bd) \div (c^2+d^2) + ((bc-ad) \div (c^2+d^2))i$$

### **Входные данные**

Арифметическая операция над двумя комплексными числами. При этом комплексные числа заданы в формате (a,b). Гарантируется, что строка не содержит пробелов. Все числа имеют значения в диапазоне от  $-10^4$  до  $+10^4$  и заданы с точностью не выше  $10^{-3}$ .

### **Выходные данные**

Одно комплексное число в формате (a,b) – вычисленное значение арифметической операции, заданной во входных данных. Точность чисел  $a$  и  $b$  должна быть не хуже  $10^{-6}$ . Решение будет считаться верным, если каждая составляющая компонента комплексного числа в ответе будет отличаться от точного значения не более чем на  $10^{-6}$ .

Гарантируется, что ответ всегда есть.

Кубок Псковской области по программированию среди школьников — 2019  
Финальный этап, первый дивизион

**Примеры**

Входные данные	Выходные данные
$(3.4, 2.7) + (1.01, 2.5)$	$(4.41, 5.2)$
$(0.4, 2.5) - (1.01, 2.5)$	$(-0.61, 0)$
$(3, 4) / (4, 3)$	$(0.96, 0.28)$
$(1.001, 0) * (0, -1.001)$	$(0, -1.002001)$

## **С. Поиск столовой**

**Ограничение времени: 1 с**  
**Ограничение памяти: 64МБ**

В школе, где учатся Василий и Пётр, учится и их общий друг Михаил, который любит собирать роботов и программировать их. На данный момент он работает над роботом-официантом, чтобы тот разносил (развозил) комплексные обеды.

Михаил освоил на занятиях по программированию алгоритм поиска кратчайшего пути в двухмерных лабиринтах. Но школа не двухмерна и требуется написать программу, которая будет искать кратчайший путь именно в трёхмерном лабиринте. Сейчас Миша бьётся над этой проблемой и надеется, что ему кто-нибудь поможет.

Василий и Пётр на данный момент ходят по школам и фотографируют внутри обстановку, чтобы затем составить трёхмерное представление каждой школы. А пока они заняты, решите чуть более простую задачу, решение этой задачи поможет Мише быстрее реализовать задуманное.

Далее изложена «простая задача».

Дан трёхмерный лабиринт размером  $6 \times 6 \times 6$ . Сначала дан самый верхний уровень, затем следующий, чуть пониже и т.д., последним дан самый нижний уровень – 1-ый этаж. Каждый уровень представляет собой 6 строчек по 6 чисел в каждой – описание проходов из соответствующей комнаты. Между описаниями уровней всегда присутствует одна пустая строка.

Каждое число описывает возможные направления движения из данной комнаты. Число представляет собой сумму величин: 1 – имеется проход на верхний этаж, 2 – имеется проход на нижний этаж, 4 – имеется проход на восток (вправо), 8 – имеется проход на запад (влево), 16 – имеется проход на север (вверх), 32 – имеется проход на юг (вниз).

### **Входные данные**

216 чисел, описывающих этажи здания. Гарантируется, что все внешние комнаты не имеют прохода наружу здания (роботу выходить из здания запрещено).

### **Выходные данные**

Длина кратчайшего маршрута (количество посещённых комнат) из комнаты, описание которой находится в последнем числе входных данных, в комнату, описание которой дано первым числом входных данных. Если построить маршрут невозможно, то выведите «impossible» (без кавычек).

Кубок Псковской области по программированию среди школьников — 2019  
 Финальный этап, первый дивизион

**Примеры**

Входные данные	Выходные данные
2 0	18
3 0	
3 0	
3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
3 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
5 40 0 0 0 0 36 28 9 0 32 0 48 0 0 4 61 40 48 0 0 0 16 48 48 0 0 0 0 48 20 12 12 12 12 24	

## **D. Миллиметровки как гуталина**

**Ограничение времени:** 1 с

**Ограничение памяти:** 64МБ

Галина любит современные технологии и ей бесплатно достался от Михаила (см. предыдущую задачу) Arduino-модуль, с помощью которого у неё получается рассылать смс-сообщения.

У Галины есть родная бабушка Нина, которая любит вязать с узорами. Она живёт далеко и интернетом пользоваться не умеет, но умеет читать смс-сообщения. Галина решила помочь бабушке с рисунками для вязания.

На первом этапе она будет высылать изображения, описывая их координатами множества прямоугольников (не попиксельно же их описывать). Бабушка на миллиметровке (миллиметровки у бабушки как гуталина у дяди кота Матроскина из мультфильма «Трое из Простоквашино», см. серию «Каникулы в Простоквашино») нарисует по полученным координатам рисунок и любимой внучке свяжет красивые тёплые рукавички в точности по присланным координатам.

Галина бабушку свою очень любит и постарается прислать информацию как можно компактнее (поменьше чисел). Она решила, что будет посылать описание только таких прямоугольников одного цвета, что увеличение их размеров по любой из координат уже будет невозможно без пересечения с уже переданными (записанными) прямоугольниками.

### **Входные данные**

Первая строка содержит два числа  $M$  и  $N$  ( $1 \leq M, N \leq 200$ ) – размеры холста в точках. Далее идёт  $M$  строк по  $N$  знаков в каждой – описание холста. «.» (точка) – отсутствие рисунка (там будут вязать основным цветом), эти данные передавать не надо. Остальные цифры от 0 до 9 определяют какой-то закодированный цвет.

Гарантируется, что в описании холста имеется хотя бы одна цифра.

### **Выходные данные**

Вывести координаты прямоугольников без пересечений и без возможности увеличения каждого из них на момент кодирования. Каждый прямоугольник описывается 5 цифрами: координаты левого верхнего угла (номер строки и номер столбца, нумерация начинается с 1), координаты нижнего правого угла прямоугольника (аналогично) и его цвет (цифра от 0 до 9).

Кубок Псковской области по программированию среди школьников — 2019  
 Финальный этап, первый дивизион

**Примеры**

Входные данные	Выходные данные
22 21	2 6 2 8 8
.....	3 5 3 5 8
.....888.....	3 6 3 8 2
....82228.....	3 9 3 9 8
...82.2228.....	4 4 6 4 8
..82.2228888.....	4 5 6 5 2
...822222877788.....	4 7 6 9 2
....8222877177788.....	4 10 6 10 8
...87888777774778...	5 11 5 13 8
...8777377757778...	6 6 7 6 2
..877777777773778..	6 11 6 13 7
..877177747777778..	6 14 6 15 8
.87777677771776778.	7 5 7 5 8
.8777777577777778.	7 7 7 8 2
..888877787777788..	7 9 7 9 8
...82287782877788...	7 10 10 11 7
....82228822287828...	7 12 7 12 1
....8222222228828....	7 13 7 15 7
....82222222228....	7 16 7 17 8
....82222222228....	8 4 9 4 8
....8222222228....	8 5 13 5 7
.....88888888.....	8 6 8 8 8
.....	8 9 15 9 7
	8 12 8 14 7
	8 15 8 15 4
	8 16 9 17 7
	8 18 9 18 8
	9 6 10 7 7
	9 8 9 8 3
	9 12 14 12 7
	9 13 9 13 5
	9 14 11 15 7
	10 3 11 3 8
	10 4 13 4 7
	10 8 11 8 7
	10 13 15 13 7
	10 16 10 16 3
	10 17 11 18 7
	10 19 11 19 8
	11 6 11 6 1
	11 7 14 7 7
	11 10 11 10 4
	11 11 12 11 7
	11 16 15 16 7
	12 2 13 2 8
	12 3 13 3 7
	12 6 13 6 7
	12 8 12 8 6
	12 10 14 10 7
	12 14 12 14 1
	12 15 16 15 7
	12 17 12 17 6
	12 18 13 19 7
	12 20 13 20 8
	13 8 15 8 7
	13 11 13 11 5
	13 14 16 14 7
	13 17 14 17 7
	14 3 14 6 8
	14 11 14 11 8
	14 18 14 19 8
	15 4 16 4 8
	15 5 16 6 2
	15 7 15 7 8
	15 10 15 10 8
	15 11 20 11 2
	15 12 15 12 8
	15 17 15 18 8
	16 7 20 7 2
	16 8 16 9 8
	16 10 20 10 2
	16 12 20 12 2
	16 13 16 13 8
	16 16 16 16 8
	16 17 16 17 2
	16 18 16 18 8
	17 5 19 5 8

Кубок Псковской области по программированию среди школьников — 2019  
 Финальный этап, первый дивизион

	17 6 19 6 2 17 8 20 9 2 17 13 20 13 2 17 14 17 15 8 17 16 19 16 2 17 17 19 17 8 18 14 20 15 2 20 6 20 6 8 20 16 20 16 8 21 7 21 15 8
21 16 ..... .....888888.. .....8.....8.. .....8.....8.....8.. .....8.....88.....8.. .....8.....8.....8.. .....8.....8.....8.. .....8.....8.....8.. .....8.....8.....888.. .....8.....8.....8.....8.. .....8.....8.....8888.. .....88.....8.....8.. .....88.....88..... .....8.....8..... .....8.....8..... .....88.....8..... .....8.....8..... .....8.....8..... .....8.....8..... .....8.....8..... .....88.....888888.. .....	2 9 2 14 8 3 8 3 8 8 3 15 5 15 8 4 7 4 7 8 4 11 5 11 8 5 6 6 6 8 5 12 5 12 8 6 14 7 14 8 7 5 7 5 8 8 4 8 4 8 8 7 8 7 8 8 13 8 13 8 9 3 9 3 8 9 6 9 6 8 9 9 11 9 8 9 11 9 12 8 10 2 11 2 8 10 5 10 5 8 10 13 11 13 8 11 4 14 4 8 11 12 11 12 8 11 14 11 14 8 12 3 12 3 8 12 8 12 8 8 12 15 12 15 8 13 6 13 7 8 13 14 13 14 8 14 5 17 5 8 14 12 14 13 8 15 13 17 13 8 17 4 17 4 8 18 3 19 3 8 18 6 19 6 8 18 14 19 14 8 19 11 19 11 8 20 4 20 5 8 20 7 20 10 8 20 12 20 13 8



## **Е. Непопулярное мнение**

**Ограничение времени:** 1 с

**Ограничение памяти:** 64МБ

Георгий с Анастасией уже со школы мечтают стать журналистами, и они не теряют даром время, пишут статьи в своих блогах. Статей у каждого накопилось уже очень много, это и про то, как всем классом ходили в кинотеатр смотреть мультфильм, и про увлечения ребят программированием и робототехникой, и про шахматный кружок, и про достижения футболистов на областных сборах и многое, многое другое.

В личном кабинете каждого блогера-журналиста можно загрузить «сырые данные» по лайкам и дизлайкам по всем статьям. Формат имеющихся данных при выгрузке:

- строчка описания статьи: название статьи – текст на английском языке (вместо пробелов знак «\_», может содержать цифры), дата опубликования в формате ГГГГ-ММ-ДД чч:мм:сс (все даты верные из нашего столетия до 2019.11.01);

- строчка описывающая получение лайков и дизлайков: порядковый номер статьи, дата в вышеуказанном формате и одна цифра «+1» – лайк, «-1» – дизлайк.

Все строки выданы в хронологическом порядке.

Необходимо составить программу, которая выведет названия самой положительно оцененной статьи и самой отрицательно оцененной статьи.

Самая положительно оцененная статья – та, которая набрала наибольшее количество лайков, а в случае равенства, та, которая это сделала за меньшее время, в случае равенства и этого показателя, та, которая была опубликована раньше.

Самая отрицательно оцененная статья – та, которая набрала наибольшее количество дизлайков, а в случае равенства, та, которая это сделала за меньшее время, в случае равенства и этого показателя, та, которая была опубликована раньше.

### **Входные данные**

В первой строке одно число  $N$  ( $1 \leq N \leq 1000$ ) – количество строк описания статей и лайков, дизлайков. Далее  $N$  строк по вышеописанному формату. Гарантируется, что длина названия статей не превышает 100 символов.

### **Выходные данные**

Две строки, на первой – название самой положительно оцененной статьи, на второй – самой отрицательно оцененной.

Кубок Псковской области по программированию среди школьников — 2019  
Финальный этап, первый дивизион

**Примеры**

Входные данные	Выходные данные
13 School_2018 2018-10-08 13:59:01 1 2018-10-08 15:43:13 +1 1 2018-10-08 20:12:47 +1 1 2018-10-08 21:30:20 -1 Chess_Club 2018-11-05 05:12:01 1 2018-11-05 11:13:16 +1 2 2018-11-05 12:01:00 -1 1 2018-11-06 15:23:33 +1 cInEmA 2019-01-11 11:18:03 3 2019-01-11 20:12:47 +1 3 2019-01-11 20:17:47 +1 3 2019-01-11 20:28:22 +1 2 2019-10-31 23:59:59 +1	School_2018 Chess_Club

## **Е. Ключи и гайки**

**Ограничение времени: 1 с**  
**Ограничение памяти: 64МБ**

Ребята в школьном возрасте не только сидят за компьютерами, играют в шахматы, решают математические и логические головоломки, помогают родителям, бабушкам и дедушкам, но также гуляют и даже катаются на велосипедах.

Ребята, о которых мы повествуем, очень дружны и смекалисты. Они понимают, что при таком количестве велосипедов неизбежно потребуется время от времени настраивать механизмы «железных коней», будут появляться поломки и потребуется производить ремонт – нужны гаечные ключи.



Ребята решили с целью экономии средств купить двусторонние гаечные ключи с открытым зевом (ключи рожковые, см. рисунок), при этом купить наименьшее их количество, достаточное для покрытия всей номенклатуры размеров (перечня размеров) имеющихся у них в велосипедах гаек и болтов.

### **Входные данные**

В первой строке через пробел даны два натуральных числа  $M, N$  ( $1 \leq M, N \leq 100$ ). На второй строке через пробел указано  $M$  натуральных чисел  $W_i$  ( $1 \leq W_i \leq 200$ ) – номенклатура размеров гаек и болтов (все размеры различны). Далее даны  $N$  строк, в каждой из которых имеются по два натуральных числа  $D1_j$  и  $D2_j$  ( $1 \leq D1_j, D2_j \leq 200$ ) – размеры 2-х зевов  $j$  гаечного ключа.

Гарантируется, что размеры всех зевов различны и для каждого размера из номенклатуры размеров гаек и болтов обязательно найдётся ключ с нужным размером зева.

### **Выходные данные**

Одно натуральное число – минимальное количество двусторонних гаечных ключей, которые необходимо купить ребятам.

Кубок Псковской области по программированию среди школьников — 2019  
Финальный этап, первый дивизион

**Примеры**

Входные данные	Выходные данные
5 4 3 5 7 17 13 5 17 13 4 2 3 9 7	4
5 4 3 5 7 17 13 5 17 13 4 7 3 9 2	3

## **Г. Дорога**

**Ограничение времени:** 1 с

**Ограничение памяти:** 64МБ

Население на планете всё увеличивается и увеличивается, мегаполисы всё растут и растут вширь, обеспеченность граждан автотранспортом увеличивается, количество грузоперевозок растёт, ради новых магистралей сносятся строения целыми деревнями и посёлками, а что делать?

Один из учителей одной деревенской школы обеспокоен вышеописанным положением дел, и хотя в настоящий момент не планируется сносить дома и постройки деревни, в которой он проживает, он понимает, что уже сейчас нужно воспитывать в нынешних учениках бережное отношение, в том числе и к чужому имуществу.

Для практических занятий ему требуется некоторое программное обеспечение, которое позволит быстро подсчитывать количество зданий, которые потребуется снести, если прямую дорогу проложить в том или ином направлении.

Помогите учителю, напишите для него программу, которая нужна ему для проведения практических занятий.

### **Входные данные**

В первой строке одно натуральное число  $N$  ( $1 \leq N \leq 10$ ) – количество построек (зданий, сооружений, строений). Далее  $N$  строк, каждая из которых содержит описание одной постройки – последовательность пар целых чисел, описывающих координаты углов строения на плане.

Далее идёт одна строка, описывающая дорогу – прямую линию. Линия задана координатами двух точек  $(x_1, y_1, x_2, y_2)$ .

Все координаты на входе заданы целыми числами, по модулю не превосходящими одной тысячи условных единиц. Все здания представляют собой многоугольники (без самопересечений), их координаты (не более 10) даны в порядке обхода по часовой стрелке. Здания не пересекаются и не касаются друг друга. Координаты начинаются и заканчиваются в одной и той же точке.

### **Выходные данные**

Количество построек, через которые проходит планируемая дорога. Дорога проходит через строение, если прямая (задающая дорогу) имеет хотя бы одну общую точку с многоугольником (задающим здание).

Кубок Псковской области по программированию среди школьников — 2019  
Финальный этап, первый дивизион

**Примеры**

Входные данные	Выходные данные
3 0 0 0 1 1 1 1 0 0 0 3 3 3 -5 2 -5 2 3 3 3 5 5 9 9 12 6 8 2 5 5 -1 -2 3 4	2
1 0 0 -1 -1 2 -2 -3 -3 -4 10 0 0 10 0 0 10	0
1 0 0 -1 -1 2 -2 -3 -3 -4 10 0 0 0 10 10 10	1