

Задача C. Concatenation of credits

Имя входного файла: c.in
Имя выходного файла: c.out
Ограничение по времени: 1 с
Ограничение по памяти: 256 Мб

Ире за ее выдающиеся достижения доверили вести пары у младшекурсников. Имея неплохой опыт занятия на парах посторонними вещами (задача A), Ира нашла себе другое развлечение. Пользуясь тем, что студенты писали зачет и сдали ей все зачетки, она стала искать закономерности в их оценках.

Преподаватели в университете, надо сказать, суровые. Во-первых, 100 баллов принципиально не ставят. А во-вторых, они никогда не поставят в зачётку оценку, которая там уже есть.

Таким образом, Ира видела в каждой зачётке шесть разных оценок от 10 до 99. И вместо того, чтобы поставить туда седьмую, она конкатенировала эти шесть оценок и делила на своё любимое число.

Саша, прогуливающий свою пару ради того, чтобы поприсутствовать на Ириной, следил за происходящим.

— Знаешь, комбинаций оценок, делящихся на это твое любимое число без остатка, довольно много.

— Аж три? — спросила Ира с плохо скрываемой издёвкой.

— Не совсем — у нас в университете студентов не хватит.

— Ну-ну.

Формат входного файла

Дано единственное любимое число I .

Формат выходного файла

Вывести количество способов выбрать упорядоченную шестерку различных двузначных чисел такую, что их конкатенация делится на I .

Ограничения

$$1 \leq I \leq 100$$

Пример

c.in	c.out
10	44828253360

Задача Е. Ехам

Имя входного файла: `e.in`
Имя выходного файла: `e.out`
Ограничение по времени: 1 с
Ограничение по памяти: 256 Мб

Неприятнь к философии сближает, необходимость сдавать экзамен — тем более. Но главное — четкое ощущение нежелания к нему готовиться.

— Смотри, Ира, подходишь к преподавателю и предлагаешь ему такую штуку: у нас есть 18 билетов, они лежат лицом вниз. Давайте K из них перевернем, потом Вы их хорошенько перетасуете и положите в одну стопку. А дальше я, в смысле ты, с закрытыми глазами разделишь их на две стопки так, что в них будет одинаковое количество открытых билетов. При этом тебе разрешается, скажем, поменять местами два билета в первой стопке, перевернуть какой-то билет в первой стопке ну и безвозвратно переложить билет из первой стопки во вторую. Ты все это делаешь с закрытыми глазами и о состоянии билетов не имеешь никакой информации кроме той, что ты знаешь число K и помнишь все свои действия.

— Ладно, и что дальше?

— Так вот, если у тебя это благополучно получается, то преподаватель дает тебе возможность отвечать любой среди открытых билетов.

— А если нет?

— Будем верить в чудо.

— Почему бы тебе самому не попробовать?

— Не то что бы у меня хорошая репутация.

— Поэтому хочешь испортить мою?

— Аж три раза.

Формат входного файла

Дано единственное четное число K — количество перевернутых лицом вверх билетов (изначально все 18 билетов находятся в первой стопке).

Формат выходного файла

Если нет возможности разделить билеты на две непустые стопки требуемым образом, вывести -1 . Иначе в первой строке вывести количество действий Q . Далее в Q строках описать порядок действий. Каждая строка должна содержать команду одного из трех типов:

`swap i j` — поменять местами билеты на позициях i и j первой стопки.

`rev i` — перевернуть билет на позиции i первой стопки.

`out i` — переместить билет на позиции i из первой стопки во вторую. После этой операции все билеты первой стопки, начиная с $(i + 1)$ -го, занимают позицию на единицу меньше.

Стопки должны оказаться непустые и содержать одинаковое количество открытых билетов.

Ограничения

$0 \leq K \leq 18$, K — четное

$0 \leq Q \leq 2^9 + 36$

$1 \leq i \neq j \leq 18$, не должны превышать текущий размер первой стопки

Пример

<code>e.in</code>	<code>e.out</code>
18	9 out 1 out 1 out 1 out 1 out 1 out 1 out 1 out 1 out 1

Задача F. Fate to hate

Имя входного файла: `f.in`
 Имя выходного файла: `f.out`
 Ограничение по времени: 1 с
 Ограничение по памяти: 256 Мб

Близился день рождения Иры. В ночь с 27-го на 28-е Саша долго не мог уснуть. Математически логичный бред дискретно лез в голову. Обыкновенное гадание на ромашке превратилось в жестокую игру чисел и битовых операций.

Снилось бесконечное поле абсолютно одинаковых ромашек. У каждой ромашки было N лепестков, а на каждом лепестке написано число. Время от времени страх перед тем, что Ира может сменить номер своего ICQ заставлял просыпаться в холодном поту. И каждый раз, когда новый номер приходил в голову, его непременно нужно было получить из лепестков, применяя к написанным на них числам операции *AND* и *OR*. Если номер удавалось получить, это было хорошим знаком, иначе же сон пропитывался ужасом от ненависти и зла.

Все ромашки одинаковые и их можно безвозмездно срывать для того, чтобы использовать нужные лепестки. То есть для получения одного номера можно сорвать несколько ромашек и взять непустое подмножество их лепестков. В своих снах Саша способен полностью контролировать порядок выполнения операций.

Формат входного файла

В первой строке дано число N — количество лепестков у ромашки. В следующей строке перечислены целые числа на лепестках a_i . Далее дано число Q — количество ICQ-номеров, которые необходимо получить. В каждой из следующих Q строк записано по одному числу b_i .

Формат выходного файла

Для каждого номера в отдельной строке выведите “Yes”, если это число можно получить из чисел на лепестках, применяя к ним операции побитового *AND* и *OR*, и “No”, если нельзя.

Ограничения

$$1 \leq N \leq 10^5$$

$$0 \leq a_i \leq 10^9$$

$$1 \leq Q \leq 10^5$$

$$0 \leq b_i \leq 10^9$$

Пример

f.in	f.out
3	Yes
1 4 5	No
6	No
1	Yes
2	Yes
3	No
4	
5	
6	

Задача J. Justice

Имя входного файла: j.in
Имя выходного файла: j.out
Ограничение по времени: 1 с
Ограничение по памяти: 256 Мб

— Давай сыграем в игру. Есть несколько кучек, в каждой из них сколько-то камешков. За один ход можно брать любое количество камешков из одной кучки, хоть все. Кто не может сделать ход, тот проиграл.

- Ладно, только я первая хожу.
- Ладно, тогда я выбираю сколько у нас будет камешков.
- Ладно, тогда я выбираю сколько будет кучек.
- А я тогда распределяю камешки по кучкам.
- Удачи.

Формат входного файла

Два числа N и K - количество камней и количество кучек.

Формат выходного файла

Если нельзя распределить N камешков ровно на K непустых кучек таким образом, что при оптимальной игре обоих выиграет второй игрок, вывести -1 . Иначе вывести ровно K натуральных чисел a_i — размеры кучек.

Ограничения

$$1 \leq N \leq 10^9$$
$$2 \leq K \leq 16$$

Пример

j.in	j.out
4 2	2 2

Задача К. Knife to me

Имя входного файла: k.in
 Имя выходного файла: k.out
 Ограничение по времени: 1 с
 Ограничение по памяти: 256 Мб

Праздничный торт был очень хорош, нежный кофейный бисквит с прослойками крема, залитый белым шоколадом — чудо кулинарного искусства. Одна проблема. Торт имеет форму призмы, в основании которой лежит правильный N -угольник — как раз по числу гостей. Но незапланированный Саша усложнил ситуацию. Теперь торт надо делить на $N + 1$ часть.

- Кто у нас тут любит геометрию?
- Ну ты тоже вроде на матфаке училась.
- Поразительная наглость. Ты во всем виноват — тебе и нож в руки.
- Тогда я предлагаю делать вертикальные разрезы.
- Ну ты прямо Капитан Очевидность.
- Это еще не все. Делим ровно на $N + 1$ часть так, чтобы объемы всех кусков были одинаковы.
- Ну ок.
- Это все еще не все. Делим так, чтобы площадь поверхности, залитой шоколадом, у всех кусков тоже была одинакова.
- Проще было самой разрезать.

Будем рассматривать проекцию торта на стол. Получается правильный многоугольник с центром в точке $(0, 0)$ и с одной из вершин в точке $(1, 0)$. Требуемые куски должны быть такими выпуклыми многоугольниками, что никакие три вершины куска не лежат на одной прямой. План разрезания должен удовлетворять условиям независимо от высоты торта. Напомним, что шоколадом залита только внешняя поверхность торта, то есть верхняя и все боковые грани.

Формат входного файла

Число N — количество гостей не считая Саши.

Формат выходного файла

Ровно $N + 1$ блок, в каждом из которых описан отдельный кусок. Описание куска начинается с числа K_i — количества вершин. Далее в K_i строках должны быть перечислены вершины куска в порядке обхода против часовой стрелки. Допускается погрешность 10^{-8} .

Ограничения

$$3 \leq N \leq 100$$

$$3 \leq K_i \leq 100$$

Пример

k.in	k.out
3	3 0.00000000000000 0.00000000000000 1.00000000000000 0.00000000000000 -0.12500000000000 0.6495190528383 4 0.00000000000000 0.00000000000000 -0.12500000000000 0.6495190528383 -0.50000000000000 0.8660254037844 -0.50000000000000 0.00000000000000 4 0.00000000000000 0.00000000000000 -0.50000000000000 0.00000000000000 -0.50000000000000 -0.8660254037844 -0.12500000000000 -0.6495190528383 3 0.00000000000000 0.00000000000000 -0.12500000000000 -0.6495190528383 1.00000000000000 -0.00000000000000

Задача L. Lie to me

Имя входного файла: 1.in
 Имя выходного файла: 1.out
 Ограничение по времени: 1 с
 Ограничение по памяти: 256 Мб

- Ничего личного, но что-то я не очень доверяю твоему способу разрезания торта.
- Ну и зря.
- Ну так докажи.
- Я не настолько крут, но вот мой ноутбук мне никогда не врет.
- Что-то я не вижу у тебя за спиной рюкзака.
- А, черт, неужели я его не взял!.. Шутка, взял, конечно.

Напомним, что план разрезания торта содержит следующую информацию: торт — правильный N -угольник с центром в точке $(0, 0)$ и одной из вершин в точке $(1, 0)$. Весь торт разрезан на $N + 1$ кусок, каждый из которых — выпуклый многоугольник.

Проверить, правда ли, что все куски — выпуклые многоугольники без трех точек на одной прямой, в порядке обхода против часовой стрелки. Проверить, все ли куски имеют равную площадь. Проверить, все ли куски имеют равную длину внешней границы, то есть той части границы, которая повторяет исходную границу торта. Правда ли, что все куски вместе составляют целый торт.

Формат входного файла

В первой строке дано число N — количество вершин торта. Далее идет описание $N + 1$ куска. Описание одного куска начинается с числа K_i — количества вершин в куске. Далее в K_i строках даны вершины куска двумя своими координатами (произвольный набор точек).

Формат выходного файла

Вывести “Yes”, если описание соответствует корректному разрезанию торта, иначе — “No”. Гарантируется, что при ответе “No” показатели, обеспечивающие этот ответ, не менее 10^{-6} .

Ограничения

$$3 \leq N \leq 100$$

$$3 \leq K_i \leq 100$$

Все координаты — вещественные числа, не более чем с 15 знаками после запятой, по модулю не превышают 100.

Пример

1.in	1.out
3 3 0.000000000000 0.000000000000 1.000000000000 0.000000000000 -0.125000000000 0.649519052838 4 0.000000000000 0.000000000000 -0.125000000000 0.649519052838 -0.500000000000 0.866025403784 -0.500000000000 0.000000000000 4 0.000000000000 0.000000000000 -0.500000000000 0.000000000000 -0.500000000000 -0.866025403784 -0.125000000000 -0.649519052838 3 0.000000000000 0.000000000000 -0.125000000000 -0.649519052838 1.000000000000 -0.000000000000	Yes

Задача М. MySpace

Имя входного файла: `m.in`
 Имя выходного файла: `m.out`
 Ограничение по времени: 1 с
 Ограничение по памяти: 256 Мб

Ситуация, аналогичная той, что описана в задаче D, сложилась на сайте MySpace.

Вкратце, два бота должны были увеличивать рейтинг фотографий. Только в связи со сложностью интерфейса сайта или просто лени, их задачи были немного упрощены. Первый бот плюсует номера всех фотографий, которые имеют остаток от деления на A равный I . Второй бот считает сумму плюсов всех фотографий, имеющих остаток от деления на B равный J . Фотографии нумеруются начиная с 0.

Формат входного файла

В первой строке даны четыре числа N , A , B и Q — количество фотографий, величины шагов для ботов, которые не меняются на протяжении всей их жизни, и суммарное количество итераций. Далее в каждой из Q строк записана команда для бота:

`s I` — первый бот плюсует фотографии $I, I + A, I + 2A, I + 3A \dots$

`g J` — второй бот суммирует плюсы фотографий $J, J + B, J + 2B, J + 3B \dots$

До действия ботов фотографии имели рейтинг 0.

Формат выходного файла

Для каждой команды второго бота вывести ее результат в отдельной строке.

Ограничения

$$1 \leq N \leq 10^5$$

$$1 \leq A \leq N$$

$$1 \leq B \leq N$$

$$0 \leq Q \leq 10^5$$

$$0 \leq I < A$$

$$0 \leq J < B$$

Пример

<code>m.in</code>	<code>m.out</code>
10 2 5 7	1
s 0	1
g 0	1
g 1	2
g 2	2
s 1	
g 3	
g 4	

Задача N. Now

Имя входного файла: `n.in`
Имя выходного файла: `n.out`
Ограничение по времени: 1 с
Ограничение по памяти: 256 Мб

Сейчас зима, во всех смыслах. И лишь воспоминания о прошедшем заставляют трезво смотреть на вещи.

— Неужели это конец?

— Кто знает...

— Ну тогда скорее к делу!

Дан неориентированный граф без петель и кратных ребер. Найти величину максимального паросочетания, то есть максимальный размер подмножества P ребер графа, что любой вершине инцидентно не более одного ребра из P .

Формат входного файла

В первой строке даны два числа N и K — количество вершин и ребер в графе. Каждая из следующих K строк содержит по два числа u и v — описание одного ребра. Гарантируется, что граф вполне случайный.

Формат выходного файла

Вывести единственное число — величину максимального паросочетания.

Ограничения

$$1 \leq N \leq 400$$

$$0 \leq K \leq \frac{N(N-1)}{2}$$

Пример

<code>n.in</code>	<code>n.out</code>
5 5 1 2 1 3 1 4 1 5 2 3	2