

Задача А. Ааа

Имя входного файла: `a.in`
Имя выходного файла: `a.out`
Ограничение по времени: 2 с
Ограничение по памяти: 256 Мб

Эта и все последующие задачи посвящены истории о девушке, которая сыграла немаловажную роль в жизни авторов.

А началось все так:

Староста группы математиков на очередной абсолютно неважной паре занималась тем, что равномерно распределяла в области ближайшего окружения листочки с sudoku. Саша не оказался обделенным вниманием, что послужило поводом заговорить.

— А, надеюсь, это самый сложный уровень?

— Как раз для тебя.

— Ну-ну, — видя глубокую иронию в ее глазах и то, что первые три строки sudoku уже заполнены.

— Решешь — еще дам.

— Я тебе не просто решу, а скажу еще, сколько решений существует, — лихо заметил Саша, не подозревая, какие проблемы себе создал.

— Ну-ну.

Следующие два дня Саша занимался только тем, что искал количество решений sudoku. Кстати, sudoku — это головоломка, в которой предлагается заполнить таблицу 9×9 числами от 1 до 9 так, чтобы в каждой строке, столбце и каждом из 9 квадратов 3×3 все числа были различны. Изначально некоторые клетки уже заполнены и остается вписать числа в пустые клетки.

— А как девушку-то зовут?

— Ааа...

Формат входного файла

В трех строках дано по 9 чисел от 1 до 9 в каждой — первые три строки sudoku.

Формат выходного файла

Единственное число — количество вариантов решения. Гарантируется, что хотя бы один вариант существует.

Пример

<code>a.in</code>	<code>a.out</code>
1 4 7 5 6 3 8 2 9 2 5 8 4 7 9 1 3 6 3 6 9 1 2 8 5 4 7	7013953152

Задача В. Bachelor pursuing

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

Не узнав имени девушки в предыдущей задаче, Саша решил навести справки, благо база данных университета и ВКонтакте всегда под рукой. Оказалось, что Ира — уже четыре года как отличница — твердо хотела получить степень бакалавра, и, как выяснилось впоследствии, степень магистра и степень кандидата наук. И мало ли что дальше?..

— Сколько же это будет в сумме? — с восторгом думал Саша, при этом подсознательно вспоминая задачу с недавнего конкурса, которую он так и не решил: даны два числа N и K , вычислить

$$\sum_{i=1}^N i^K$$

Формат входного файла

В первой строке дано число Q — количество запросов. В каждой из Q следующих строк даны два числа N и K .

Формат выходного файла

Для каждого запроса в отдельной строке вывести одно число — сумму K -ых степеней натуральных чисел от 1 до N по модулю 1,000,000,007.

Ограничения

$$1 \leq Q \leq 41000$$

$$1 \leq N \leq 10^9$$

$$1 \leq K \leq 1000$$

Пример

<code>b.in</code>	<code>b.out</code>
4	3
2 1	30
4 2	36
3 3	675987247
11 11	

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

<code>c.in</code>	<code>c.out</code>
10	44828253360

Задача D. DeviantArt

Имя входного файла: d.in
 Имя выходного файла: d.out
 Ограничение по времени: 2 с
 Ограничение по памяти: 256 Мб

Фотография — еще одно увлечение Иры. Ее произведения занимают достойное место на страницах DeviantArt. Каждый свободный пользователь интернета может зайти в ее галерею и плюсануть понравившиеся изображения.

— Ммм! Неплохая работа для бота.

Суть бота заключалась в следующем: специально созданный пользователь плюсует K фотографий, начиная с I -ой, с шагом A . Второй бот был создан для контроля результата. Он проверяет сумму всех плюсов L фотографий, начиная с J -ой, с шагом B . Фотографии нумеруются с 0.

Боты работают независимо до тех пор, пока администрация DeviantArt не заметит накрутки вклада и не забанит анонимусов.

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 K — первый бот плюсует K фотографий, начиная с I -ой.

g J L — второй бот суммирует плюсы L фотографий, начиная с J -ой.

До действия ботов фотографии имели рейтинг 0.

Формат выходного файла

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

Ограничения

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

$$1 \leq A \leq N$$

$$1 \leq B \leq N$$

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

$$0 \leq I \leq N - 1$$

$$1 \leq K \leq N, I + (K - 1)A \leq N - 1$$

$$0 \leq J \leq N - 1$$

$$1 \leq L \leq N, J + (L - 1)B \leq N - 1$$

Пример

d.in	d.out
10 2 5 7	1
s 0 5	1
g 0 2	1
g 1 2	1
g 2 2	1
s 1 3	
g 3 1	
g 4 2	

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

<code>f.in</code>	<code>f.out</code>
3	Yes
1 4 5	No
6	No
1	Yes
2	Yes
3	No
4	
5	
6	

Задача G. Genealogic tree

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

На дне рождения Иры было много родственников, так что знакомство с ними оказалось непростой задачей. Для того, чтобы получить общую картину, возникла идея составить генеалогическое дерево. В сумбурной обстановке как-то так получилось, что дерево оказалось не бинарным, хотя количество вершин было вполне логичным — $2^k - 1$.

- Что-то тут не так.
- Да ну?
- Я предлагаю все перепроверить.
- Конкретнее.
- Разобьем дерево на связные части по 2^i вершин и поручим каждому проверить свою часть.
- Почему именно так?
- Не знаю, люблю степени двойки и к тому же в сумме будет ровно то, что надо. Хотя тут есть маленькая проблема, это можно сделать кучей способов.
- Да, я помню sudoku и зачетки. Сколько же на этот раз?
- У тебя великолепная память, дай мне несколько минут на подумать.
- Лови.

Формат входного файла

В первой строке дано число $N = 2^k - 1$ — количество вершин в дереве. В следующей строке дано $N - 1$ число. a_i означает наличие ребра между вершинами $i + 1$ и a_i .

Формат выходного файла

Вывести одно число — количество способов разделить дерево ровно на k связных блоков размеров $1, 2, 4, \dots, 2^{k-1}$. Каждая вершина должна попасть ровно в один блок.

Ограничения

$$0 \leq k \leq 12$$

$$1 \leq a_i < i + 1$$

Пример

g.in	g.out
7 1 1 2 2 3 3	4

Задача Н. Happy Birthday

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

Настал долгожданный день рождения. Осталось только одно испытание — добраться к Ире домой.

Самый популярный вид транспорта в городе — маршрутки. У каждой маршрутки есть два водителя, один из них любит один маршрут, а второй — другой. Правда место и время отправления маршрутки одинаковы для обоих водителей. Каждый день один из водителей работает, а второй отдыхает. Если Саша окажется на какой-то остановке, то он сразу же узнает, какие из водителей работают сегодня на маршрутках, отправляющихся с этой остановки. Однако до того, как он попадет на остановку, он знает только расписание возможного движения маршруток и вероятность того, что работает первый водитель. Саша живет возле остановки с номером 1 и может оказаться на ней в любое время, а Ира живет рядом с остановкой N . Требуется найти математическое ожидание времени прибытия к Ире на день рождения. При этом Саша никогда не воспользуется способом, который может привести к тому, что он не попадет к Ире вообще, а среди всех остальных выбирает тот, который минимизирует ожидаемое время прибытия.

Важные факты:

Сеть движения маршруток представляет собой ациклический ориентированный граф.

С маршрутки на маршрутку можно пересаживаться мгновенно.

При определении оптимальной стратегии Саша использует в том числе и то, что по прибытии на каждую остановку он узнает сегодняшних шоферов.

О своей остановке Саша тоже изначально ничего не знает.

Формат входного файла

В первой строке даны два числа N и K — число остановок и действующих маршруток. Далее в каждой из K строк описана информация о маршрутах: семь целых чисел $u\ d\ p\ v_1\ a_1\ v_2\ a_2$ — номер остановки и время отправления, вероятность того, что работает первый водитель, место и время прибытия, если работает первый водитель, место и время прибытия, если работает второй водитель.

Формат выходного файла

Единственное число — математическое ожидание времени прибытия на остановку N с абсолютной или относительной погрешностью 10^{-6} или -1 , если есть ненулевая вероятность туда не попасть.

Ограничения

$2 \leq N \leq 10^5$
 $0 \leq K \leq 10^5$
 $1 \leq u, v_1, v_2 \leq N$
 $u \neq v_1$
 $u \neq v_2$
 $0 \leq d, a_1, a_2 \leq 1440$
 $d < a_1$
 $d < a_2$
 $0 < p < 100$

Пример

h.in	h.out
5 6 1 60 50 2 200 3 150 1 100 25 2 160 3 150 1 200 50 5 350 4 300 2 180 50 5 300 4 280 3 400 80 5 600 5 660 4 350 50 5 500 5 550	423.4375000000000

Задача I. I love Ira

Имя входного файла: `i.in`
Имя выходного файла: `i.out`
Ограничение по времени: 3 с
Ограничение по памяти: 256 Мб

Саша хотел сделать Ире совершенно необычный подарок на день рождения. Романтическое сообщение, выложенное решенными задачами, на самом популярном украинском сайте, посвященном спортивному программированию.

Задачи на сайте располагаются в виде прямоугольной таблицы, сданные задачи подсвечиваются приятным зеленым цветом. Это однажды натолкнуло на мысль зажечь в таблице только те ячейки, которые бы сформировали задуманную картинку или надпись...

И если с содержанием сообщения неоднозначностей не возникало, то его расположением в таблице можно было управлять. Конечно же от места расположения надписи зависело то, какие задачи придется решать. Поэтому Саша назначил каждой задаче оценку сложности — число от 0 до 9. И решил выбрать такое расположение, чтобы сумма оценок решенных им задач была максимальна. Надпись может располагаться в таблице где угодно, но только чтобы весь шаблон находился в пределах таблицы.

Формат входного файла

В первой строке даны два числа R и C — количество строк и столбцов в таблице. Далее в R строках описана карта задач. В каждой строке ровно по C символов — сложности задач. Сложность задачи измеряется цифрой от 0 до 9. В следующей строке даны два числа H и W — высота и ширина надписи. Далее в H строках дан шаблон надписи. Каждая строка состоит из W символов: '#' означает необходимость решения задачи, '.' означает, что задачу надо не решать.

Формат выходного файла

В единственной строке вывести максимальную сумму оценок сложности решенных задач.

Ограничения

$$1 \leq R, C \leq 800$$
$$1 \leq H \leq R$$
$$1 \leq W \leq C$$

Пример

i.in	i.out
22 18 000000000001000000 003001000000000000 000000000000000003 020000000010000000 090000000901000100 300400001911323500 000000000000000000 000000000000000000 000001000001000049 020100113210100000 200000006220004100 600009000217100009 000001000031004210 020101600100000190 110001002000000000 000092096071021990 000094000120010100 201003007430040300 100000010000100000 101040008000020001 404000000040043200 000000010000001000 17 18 ###...#..... .#...#..... .#...#...#...#...# .#...#...#...#...# .#...#...#...#...# .#...#...#...#...# ###...#...#...####..... ...#..... ...#...#...#...# ...#...#...#...# ...#...#...#...# ...#...#...#...# ...#...#...#...# ...###...#...#...#...	69

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