

Problem A. Aaa (Division 1 Only!)

Input file: a.in
Output file: a.out
Time limit: 7 seconds
Memory limit: 256 Mebibytes

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...

Input

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

Output

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

Example

a.in	a.out
1 4 7 5 6 3 8 2 9	7013953152
2 5 8 4 7 9 1 3 6	
3 6 9 1 2 8 5 4 7	

Problem B. Bachelor pursuing (Division 1 Only!)

Input file: **b.in**
Output file: **b.out**
Time limit: 3 seconds
Memory limit: 256 mebibytes

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

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

Input

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

Output

Для каждого запроса в отдельной строке вывести одно число — сумму K -ых степеней натуральных чисел от 1 до N по модулю $10^9 + 7$.

($1 \leq Q \leq 41\,000$, $1 \leq N \leq 10^9$, $1 \leq K \leq 1000$)

Example

b.in	b.out
4	3
2 1	30
4 2	36
3 3	675987247
11 11	

Problem C. Concatenation of credits

Input file: c.in
Output file: c.out
Time limit: 1 second
Memory limit: 256 Mebibytes

Ире за ее выдающиеся достижения доверили вести пары у младшекурсников. Имея неплохой опыт занятия на парах посторонними вещами (задача A), Ира нашла себе очередное 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 хватает.
- Ну-ну.

Input

Дано единственное любимое число I .

Output

Вывести количество способов выбрать упорядоченную шестерку различных двузначных чисел такую, что их конкатенация делится на I ($1 \leq I \leq 100$).

Example

c.in	c.out
10	44828253360

Problem D. DeviantArt (Division 1 Only!)

Input file: d.in
Output file: d.out
Time limit: 4 seconds
Memory limit: 256 mebibytes

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. При этом рейтинг соответствующей фотографии увеличится на 1.

— 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 рейтингов 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.

Input

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

$(1 \leq N \leq 10^5, 1 \leq A \leq N, 1 \leq B \leq N, 0 \leq Q \leq 10^5;$

$0 \leq I \leq N - 1, 1 \leq K \leq N, I + (K - 1)A \leq N - 1;$

$0 \leq J \leq N - 1, 1 \leq L \leq N, J + (L - 1)B \leq N - 1)$

Output

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

Example

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	

Problem E. Exam

Input file: e.in
Output file: e.out
Time limit: 1 second
Memory limit: 256 Mebibytes

Неприязнь 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.

Input

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

Output

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.

($0 \leq K \leq 18$, K — четное; $0 \leq Q \leq 2^9 + 36$; $1 \leq i \neq j \leq 18$, i и j не должны превышать текущий размер первой стопки)

Example

e.in	e.out
18	9 out 1 out 1 out 1 out 1 out 1 out 1 out 1 out 1 out 1 out 1

Problem F. Fate to hate

Input file: f.in
Output file: f.out
Time limit: 1 second
Memory limit: 256 Mebibytes

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.

Input

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_j .

($1 \leq N \leq 10^5$, $0 \leq a_i \leq 10^9$, $1 \leq Q \leq 10^5$, $0 \leq b_j \leq 10^9$)

Output

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.

Example

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

Problem G. Genealogic tree (Division 1 Only!)

Input file: g.in
Output file: g.out
Time limit: 2 seconds
Memory limit: 256 Mebibytes

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.

Input

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 . Вершины нумеруются с единицы. ($0 \leq k \leq 12, 1 \leq a_i < i + 1$)

Output

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.

Example

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

Problem H. Happy Birthday (Division 1 Only!)

Input file: h.in
Output file: h.out
Time limit: 2 seconds
Memory limit: 256 Mebibytes

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.

Input

В первой строке даны два числа N и K — число остановок и действующих маршруток. Далее в каждой из K строк описана информация о маршрутках: 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.

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

Output

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.

Example

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

Problem I. I love Ira (Division 1 Only!)

Input file: `i.in`
Output file: `i.out`
Time limit: 12 seconds
Memory limit: 256 mebibytes

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 придётся 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, при этом весь шаблон должен находиться в пределах таблицы.

Input

В первой строке даны два целых 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 оставить нерешённой.

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

Output

В единственной строке вывести максимальную сумму оценок сложности решенных задач.

Example

i. in	i. out
<pre>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 ###...#..... .#...#..... .#...#.#.#.#.# .#...#.#.#.#.# .#...#.#.#.#.# .#...#.#.#.#.# ###...#.#.#.####..... ..#..... ...#.#.#.#.# ...#...#.#.# ...#...#.#.# ...#...#.#.# ...#...#.#.# ..###...#.#.#</pre>	<pre>69</pre>

Problem J. Justice

Input file: j.in
Output file: j.out
Time limit: 1 second
Memory limit: 256 Mebibytes

— 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.

Input

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

$(1 \leq N \leq 10^9, 2 \leq K \leq 16)$

Output

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.

Example

j.in	j.out
4 2	2 2

Problem K. Knife to me (Division 2 Only!)

Input file: k.in
Output file: k.out
Time limit: 1 second
Memory limit: 256 Mebibytes

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.

Input

Число N — количество гостей не считая Саши ($3 \leq N \leq 100$).

Output

Ровно $N + 1$ блок, в каждом из которых описан отдельный кусок. Описание куска начинается с числа K_i — количества вершин ($3 \leq K_i \leq 100$). Далее в K_i строках должны быть перечислены вершины куска в порядке обхода против часовой стрелки. Допускается погрешность 10^{-8} .

Example

k.in	k.out
3	3
	0.000000000000 0.000000000000
	1.000000000000 0.000000000000
	-0.125000000000 0.6495190528383
	4
	0.000000000000 0.000000000000
	-0.125000000000 0.6495190528383
	-0.500000000000 0.8660254037844
	-0.500000000000 0.000000000000
	4
	0.000000000000 0.000000000000
	-0.500000000000 0.000000000000
	-0.500000000000 -0.8660254037844
	-0.125000000000 -0.6495190528383
	3
	0.000000000000 0.000000000000
	-0.125000000000 -0.6495190528383
	1.000000000000 -0.000000000000

Problem L. Lie to me (Division 2 Only!)

Input file: 1.in
Output file: 1.out
Time limit: 1 second
Memory limit: 256 Mebibytes

- 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.

Input

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).

($3 \leq N \leq 100$, $3 \leq K_i \leq 100$, все координаты — вещественные числа, не более чем с 15 знаками после запятой, по модулю не превышают 100)

Output

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

Example

1.in	1.out
3	Yes
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	

Problem M. MySpace (Division 2 Only!)

Input file: m.in
Output file: m.out
Time limit: 2 seconds
Memory limit: 256 Mebibytes

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.

Input

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.

$(1 \leq N \leq 10^5, 1 \leq A \leq N, 1 \leq B \leq N, 0 \leq Q \leq 10^5, 0 \leq I < A, 0 \leq J < B)$

Output

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

Example

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

Problem N. Now (Division 2 Only!)

Input file: n.in
Output file: n.out
Time limit: 1 second
Memory limit: 256 Mebibytes

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 .

Input

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.

$$(1 \leq N \leq 400, 0 \leq K \leq \frac{N(N-1)}{2})$$

Output

Вывести единственное число — величину максимального паросочетания.

Example

n.in	n.out
5 5 1 2 1 3 1 4 1 5 2 3	2