

Задача A. Squares

Имя входного файла: `a.in`
Имя выходного файла: `a.out`
Ограничение по времени: 6 seconds
Ограничение по памяти: 256 Mebibytes

Во время строительства комплекса зданий Новосибирского Государственного университета произошёл следующий случай. Уже практически построенным зданием заинтересовалось могущественное и при этом очень секретное ведомство. Высокопоставленным офицерам приглянулось здание, и они, использовав рычаги в партийных органах, уже практически добились передачи здания под свой контроль. Но Президент Сибирского Отделения Академии Наук Михаил Алексеевич Лаврентьев на свой страх и риск заявил, что здание уже используется. Сразу же после заседания обкома, уже почти в полночь, Лаврентьев позвонил Илье Несторовичу Векуа и сообщил о сложившейся ситуации. За ночь требовалось перевезти основную часть оборудования НГУ из здания школы, которое было выделено университету первоначально, в новое здание, а с утра быть готовыми к приёму комиссии из очень секретного ведомства, которая собиралась проверить, действительно ли здание уже используется. Илья Несторович собрал сотрудников и они приняли решение действовать...

Дороги к новому зданию были не только не заасфальтированы, но даже далеко не везде утрамбованы. Представим дорогу в виде длинной прямоугольной полосы шириной N одинаковых квадратных клеток. В каждом из N рядов утрамбованным было только некоторое количество подряд идущих клеток (как оказалось, это было требование службы безопасности, гарантирующее непопадание врага на объект).

Требуется утрамбовать минимальное количество клеток, для того, чтобы, во-первых, требование службы безопасности по-прежнему было выполнено, а, во-вторых, существовал утрамбованный прямоугольник во всю ширину дороги со стороной заданной длины, который бы сотрудники могли использовать как промежуточный пункт для переносимого оборудования.

Формат входного файла

В первой строке входного файла задано целое число d ($1 \leq d \leq 100$) — количество тестовых примеров.

В первой строке каждого тестового примера заданы целые числа N ($1 \leq N \leq 10^5$) — ширина обозначающей дорогу полосы и K ($1 \leq k \leq 10^9$) — длина стороны требуемой площадки.

В следующей строке указаны изначально утрамбованные участки, заданные N парами целых чисел a_i и b_i — соответственно x -координатами начала и конца участка, утрамбованного в i -м сверху ряду ($0 \leq a_i < b_i \leq 10^9$).

Формат выходного файла

Для каждого теста выведите одно число — минимальное количество клеток, которое необходимо утрамбовать для получения площадки ширины N и стороной заданной длины с соблюдением указанных в условии дополнительных требований.

Пример

<code>a.in</code>	<code>a.out</code>
1	5
3 2	
0 3 3 4 5 8	

Задача В. Cyclic letters

Имя входного файла: `b.in`
Имя выходного файла: `b.out`
Ограничение по времени: 4 seconds
Ограничение по памяти: 256 Mebibytes

Уже в новом здании сотрудники университета заметили небольшие, но неприятные недоделки. В частности, в генеральном плане помещения были обозначены строками из N строчных латинских букв. Такие же буквы были написаны мелом на соответствующих дверях. Но строители, спеша выполнить задание к 10-й годовщине 25-летия первого аврала, в одном из корпусов написали мелом не те буквы.

Однако один из сотрудников, специализирующийся на разработке ПО для ЭВМ, заметил, что строители переставляли буквы «по циклу» (скорее всего, прораб просто повторял фразу несколько раз — вот и написали, что услышали), и что в изначальной нумерации строка была лексикографически наименьшая. И значит, осталось найти лексикографически наименьшую строку, которую можно получить из данной многократным применением операции сдвига, заключающейся в переносе последней буквы строки в её начало.

Формат входного файла

В первой строке входного файла задано натуральное число d ($1 \leq d \leq 100$) — количество тестовых примеров.

Каждый тестовый пример состоит из одной строки, состоящей не менее, чем из 1 и не более, чем из 10^6 строчных латинских букв.

Формат выходного файла

Для каждого теста выведите лексикографически наименьшую строку, получаемую из заданной указанным в условии способом.

Пример

<code>b.in</code>	<code>b.out</code>
3	xzy
zyx	kk1
kk1	cicp
icpc	

Задача C. Shortest Path (Division 1 Only!)

Имя входного файла: c.in
Имя выходного файла: c.out
Ограничение по времени: 4 seconds
Ограничение по памяти: 256 Mebibytes

И вот оборудование было перенесено... уже настало утро, так что сотрудники и студенты университета должны были оставаться на занятиях — иначе комиссия не сочтёт, что здание эксплуатируется.

Представитель прибывшей комиссии зашёл в одну из аудиторий физического факультета. Из того, что говорили участники семинара, знакомым показалось одно слово — «космос». В своё время представитель комиссии работал в отделе по обеспечении секретности некоторого заведения космической специализации.

Он взял слово, и сказал, что близок тот час, когда партия отправит энтузиастов на штурм пояса астероидов, и что уже сейчас разрабатываются средства передвижения по их поверхности. И что если уважаемые физики решат задачу нахождения самого короткого пути между двумя точками на поверхности астероида, то они внесут большой вклад...

Последовала масса уточнений, после которой задачу удалось свести к следующей: задан выпуклый многогранник, состоящий из N граней, и две точки на его поверхности. Найти кратчайший путь между двумя заданными точками, проходящий по поверхности многогранника.

Напишите программу, делающую аналогичные вычисления.

Формат входного файла

В первой строке входного файла задано натуральное число d ($1 \leq d \leq 150$) — количество тестовых примеров.

Первая строка тестового примера содержит N — количество граней многогранника ($4 \leq N \leq 15$). В последующих N строках описаны грани, по одной на строку. Описание грани начинается с числа K — количество вершин грани ($3 \leq K \leq N - 1$). Далее последовательно перечислены тройки координат x, y, z всех K вершин грани. Гарантируется, что никакие две различные грани не лежат в одной плоскости.

Последняя строка каждого тестового примера содержит шесть чисел — x_a, y_a, z_a — координаты начальной точки и x_b, y_b, z_b — координаты конечной точки.

Все координаты во входном файле заданы целыми числами, по модулю не превосходящими 1000.

Формат выходного файла

Для каждого теста выведите с точностью до 2 знаков длину кратчайшего пути по поверхности многогранника между заданными точками.

Пример

c.in	c.out
1	2.24
6	
4 0 0 0 0 0 1 0 1 1 0 1 0	
4 1 0 0 1 0 1 1 1 1 1 1 0	
4 0 0 0 0 1 0 1 1 0 1 0 0	
4 0 0 1 0 1 1 1 1 1 1 0 1	
4 0 0 0 0 0 1 1 0 1 1 0 0	
4 0 1 0 0 1 1 1 1 1 1 1 0	
0 0 0 1 1 1	

Задача D. Tournament

Имя входного файла: `d.in`
Имя выходного файла: `d.out`
Ограничение по времени: 4 seconds
Ограничение по памяти: 256 Mebibytes

Ещё один член комиссии всю свою карьеру курировал разные спортивные организации. Так что он пошёл не в аудитории, а в шахматную секцию университета.

В секции внешне всё было в порядке — шахматные столики, портреты чемпиона мира Михаила Ботвинника на стенах... Присутствовал даже председатель секции. У проверяющего появилась идея в ближайшие выходные организовать для строителей Академгородка показательный шахматный турнир силами секции Университета.

Для проведения показательного турнира требуется отобрать 3 шахматистов из N , указанных в списках. Каждый из шахматистов секции имеет некий разряд, при этом все N разрядов попарно различны, и шахматист с лучшим разрядом всегда обыгрывает игрока с худшим разрядом. В секции, недавно начавшей работу, нет сведений о разряде каждого игрока, но зато есть результаты игр для некоторых пар шахматистов секции.

Проверяющий и председатель секции хотят отобрать 3 игрока таким образом, чтобы по имеющимся данным невозможно было заранее вычислить результат игры ни между какими двумя из них.

Формат входного файла

В первой строке входного файла задано натуральное число d ($1 \leq d \leq 50$) — количество тестовых примеров.

В первой строке каждого тестового примера задано N — количество заявившихся шахматистов ($3 \leq N \leq 1000$). Шахматисты пронумерованы от 1 до N в порядке вступления в секцию. Далее идёт N строк, каждая из которых состоит из N символов 0 или 1. 1 в j -м столбце i -й строки обозначает, что в секции есть информация, что шахматист i выиграл у шахматист j , в противном случае там располагается 0. Гарантируется, что имеющаяся в секции информация непротиворечива.

Формат выходного файла

Для каждого тестового примера в случае, если выбрать тройку игроков указанным в задаче способом можно, выведите в отдельной строке слово YES, а за ним номера шахматистов, составляющих какую-либо из таких троек. Если выбрать тройку игроков соответствующим способом нельзя, выведите NO.

Пример

d.in	d.out
2	YES 6 5 7
8	NO
00111111	
00011011	
00000100	
00001011	
00000001	
00000000	
00000000	
00000000	
6	
011111	
001111	
000010	
000001	
000000	
000000	

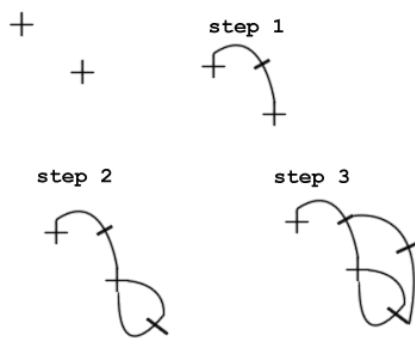
Задача E. Brussels Sprouts (Division 2 Only!)

Имя входного файла: e.in
Имя выходного файла: e.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 Mebibytes

Тем временем в аудиториях занятия протекали вполне обыденно — некоторые студенты даже развлекались, играя в простые игры. Например, в такую.

Игра «Brussels sprouts» — это игра для двух игроков со следующими правилами. Игра начинается с N «крестиков» — объектов с четырьмя свободными концами. Каждый ход представляет из себя соединение двух свободных концов кривой (не пересекающей ни одну из существующих кривых) и добавления короткого штриха поперёк кривой (чтобы создать два новых свободных конца). Игроки делают ходы по очереди, проигрывает игрок, который в свою очередь не может сделать хода.

На рисунке изображён пример первых 3 ходов для $N = 2$.



Вошедшая на лекцию комиссия обратила внимание преподавателя, читавшего курс дискретной математики, на это нарушение. Прозвучали слова «тунеядство», «саботаж» и «отчисление». Преподаватель ответил, что эти студенты выполняют специальное задание — им задана игра и они должны по заданному N определить игрока, который выиграет при оптимальной игре обоих игроков.

Студенты, благодарные лектору за спасение, тут же рассказали решение этой задачи аудитории. А сможете ли Вы решить такую задачу?

Формат входного файла

В первой и единственной строке входного файла содержится целое число N — количество «крестиков» ($1 \leq N \leq 13$).

Формат выходного файла

Если выигрывает первый игрок, выведите слово “First”, если выигрывает второй — выведите “Second”.

Пример

e.in	e.out
1	First

Задача F. Puzzle (Division 1 Only!)

Имя входного файла:	<code>f.in</code>
Имя выходного файла:	<code>f.out</code>
Ограничение по времени:	15 seconds
Ограничение по памяти:	256 Mebibytes

Но не во всех корпусах дела шли так гладко. Например, перед транспортировкой громоздкого оборудования в корпуса химического факультета был разобран паркет... и возникла проблема — как до прихода комиссии вернуть всё на место. Один профессор, побывавший в загранкомандировке, сказал, что видел в ГДР похожую настольную игру под названием «Puzzle», и что он как-то раз её складывал...

Паркет на химфаке представлял собой прямоугольник, разбитый на $m \times n$ квадратных блоков. Каждая сторона блока может или быть ровной, или содержать выемку, или содержать выступ. Два блока могут прилегать друг к другу сторонами только в том случае, если на этих сторонах у одного из них выемка, а у другого — выступ. Перед укладкой на место блоки можно вращать вокруг центра квадрата в горизонтальной плоскости, но нельзя переворачивать на другую сторону. Ваша задача — по заданному набору блоков определить, сколько существует различных способов сборки паркета. Способы считаются одинаковыми, если в каждой из mn позиций находятся одинаково ориентированные блоки одинаковой конфигурации. Предостережение: два способа сборки, переводимые друг в друга поворотом, всё равно считаются различными.

Формат входного файла

В первой строке входного файла задано натуральное число d ($1 \leq d \leq 100$) — количество тестовых примеров.

Каждый тестовый пример начинается со строки, содержащей m и n — размеры паркета ($1 \leq n \leq 6$, $1 \leq m \leq 5$). В следующих mn строках описаны конфигурации заданных блоков. Каждый блок описывается четырьмя числами, задающими тип сторон в порядке обхода: 0, если сторона ровная, 1, если на стороне имеется выступ, и 2, если выемка.

Формат выходного файла

Для каждого тестового примера в отдельной строке выведите одно число — количество различных способов сборки паркета.

Пример

f.in	f.out
2	1
3 3	0
0 0 1 2	
0 0 1 2	
0 0 1 2	
0 0 1 2	
0 1 1 2	
0 1 1 2	
0 1 1 2	
0 1 1 2	
2 2 2 2	
3 3	
0 0 1 2	
0 0 1 2	
0 0 1 2	
0 0 1 2	
0 1 1 2	
0 1 1 2	
0 1 1 2	
0 1 2 2	
2 2 2 2	

Задача G. Sum of numbers

Имя входного файла: `g.in`
Имя выходного файла: `g.out`
Ограничение по времени: 6 seconds
Ограничение по памяти: 256 Mebibytes

После беседы с председателем шахматной секции о перспективах советских шахмат и о шансах гроссмейстеров на предстоящих международных турнирах, сопровождавшейся несколькими тостами во славу различных дебютных школ, спортивный куратор попал на занятия по математическому анализу.

Обсуждалось понятие «нормы». Ведущий семинар преподаватель узнал куратора (оба они были перворазрядниками)... тот сказал, что знает «отличную задачу про норму и шахматную доску». Тот удивился, но слово предоставил.

Куратор тут же вспомнил только что завершившуюся беседу и рассказал следующую задачу.

На шахматной доске $m \times n$, на каждом поле которой стоит рюмка с целым неотрицательным числом грамм, двое играют в следующую игру. Первый игрок рисует на доске многоугольник без самопересечений, стороны которого идут по границам полей, а также под углами 45, 135, 225 и 315 градусов к горизонтали, а вершины являются углами полей. Второй игрок выпивает содержимое рюмок на тех полях, которые как минимум наполовину принадлежат многоугольнику, записывает общее количество выпитого в этот раз, после чего многоугольник стирается, рюмки наполняются так же, как и в самом начале, процесс «многоугольник-выпивка-запись» повторяется заново, и так k раз.

Так как второй игрок должен соблюдать норму выпитого, а записей он не помнит по понятным причинам, надо восстановить k записанных вторым игроком сумм.

Студенты вместе с преподавателем и проверяющим посмеялись над этой задачей — проверяющий уже понял, что слово «норма» тут имело другое значение.

Тем не менее, решите задачу в поставленной шахматистом формулировке.

Формат входного файла

В первой строке входного файла задано натуральное число d ($1 \leq d \leq 10$) — количество тестовых примеров.

Первая строка каждого тестового примера содержит ширину доски n , длину доски m и количество последовательно нарисованных многоугольников k ($1 \leq n, m \leq 1000, 1 \leq k \leq 10^4$). Каждая из следующих m строк содержит n чисел $g_{i,j}$ ($i = 1 \dots m, j = 1 \dots n, 0 \leq g_{i,j} \leq 2^{16}$). $g_{i,j}$ — количество грамм в рюмке на поле с координатами i, j . В следующих k строках последовательно задаются многоугольники, i -я из этих строк задаёт многоугольник, нарисованный в i -й раз. Описание многоугольника начинается с количества вершин l ($3 \leq l \leq 100$). Далее заданы l пар чисел x, y — координаты последовательных вершин многоугольника ($0 \leq x \leq n, 0 \leq y \leq m$).

Формат выходного файла

Для каждого тестового примера выведите в отдельной строке k чисел, разделённых пробелами. i -е число обозначает количество выпитого в граммах, подсчитанное для i -го на входе многоугольника.

Пример

<code>g.in</code>	<code>g.out</code>
1 7 6 1 1 2 4 1 2 4 0 1 4 1 1 4 1 1 4 2 1 2 4 1 2 4 2 1 4 1 1 3 4 4 4 2 4 1 4 4 2 1 1 1 1 5 6 1 0 1 6 3 6 5 4 3 2 3 0	42

Задача H. Subset

Имя входного файла: `h.in`
Имя выходного файла: `h.out`
Ограничение по времени: 4 seconds
Ограничение по памяти: 256 Mebibytes

Кое-где проверяющие вынуждены были только делать умный вид. Так, например, на семинаре по теории множеств студентам была задана следующая задача:

Найти g -е лексикографически m -элементное подмножество множества $n = \{1, 2, \dots, n\}$.

Один из студентов вышел к доске и стал рассказывать решение. Проверяющий с умным видом кивал, а потом задал один уточняющий вопрос... «Вот вы говорите — множество. А множество чего? И сколько это — тысяча, две, три?».

Невыспавшиеся студенты еле сдержали смех до ухода проверяющего...

Надеемся, что вы не только знакомы с основными элементами теории множеств, но и сможете, подобно студентам НГУ, решить эту задачу.

Формат входного файла

В первой строке входного файла задано натуральное число d ($1 \leq d \leq 100$) — количество тестовых примеров.

Каждый тестовый пример состоит из трёх чисел m , n и g ($1 \leq m \leq n \leq 500$, $1 \leq g$). Гарантируется, что как минимум g различных m -элементных подмножеств существует.

Формат выходного файла

Для каждого тестового примера в выходной файл выведите g -е лексикографически подмножество. Элементы внутри подмножества также должны быть упорядочены лексикографически.

Пример

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

Задача I. Number (Division 1 Only!)

Имя входного файла: `i.in`
Имя выходного файла: `i.out`
Ограничение по времени: 4 seconds
Ограничение по памяти: 256 Mebibytes

Проверяющий на занятиях по теории чисел оказался более компетентен — он был кандидатом физико-математических наук и специализировался в области криптографии. Так что практически сразу, как его представили студентам, он задал им задачу, которую ему самому пришлось решать во время Второй Мировой войны при расшифровке закодированной переписки резидентур абвера в Западной Европе с Центром.

Задача звучала так:

Задано число m , простое число p и целое неотрицательное число a , меньшее p . Существует ли такое целое число n , для которого выражение $n^n + m^m$ даёт остаток a по модулю p ? Если существует, выведите любое такое число.

Проверяющий заметил, что задача очень сложная, но тем большим было его удивление, когда один из студентов высказал несколько идей, ведущих к правильному решению задачи.

А сможете ли Вы, обладая мощной вычислительной техникой, решить эту задачу?

Формат входного файла

В первой строке входного файла задано натуральное число d ($1 \leq d \leq 300$) — количество тестовых примеров.

Каждый тестовый пример состоит из одной строки, в которой заданы три целых числа p , a и m ($2 \leq p \leq 10^9$, $0 \leq a < p$, $1 \leq m \leq 20$, $m < p$). Гарантируется, что p — простое.

Формат выходного файла

Для каждого тестового примера в отдельной строке выведите через пробел YES и число n , если существует $n < 10^{1000}$, удовлетворяющее требованию задачи, и NO в противном случае.

Пример

<code>i.in</code>	<code>i.out</code>
2	YES 567
11 3 1	YES 2
11 8 2	

Задача J. Calendar

Имя входного файла:	<code>calendar.in</code>
Имя выходного файла:	<code>calendar.out</code>
Ограничение по времени:	4 seconds
Ограничение по памяти:	256 Mebibytes

На занятиях по астрономии обсуждалось построение календаря. Ведущий семинара решил задать задачу, которая будет как минимум интересно звучать и для проверяющих... Например, про обитаемые планеты.

В недавно открытой звёздной системе k обитаемых планет. Планеты занумерованы от 1 до k , при этом чем меньше номер планеты, тем ближе планета находится к звезде. Календарь на каждой из планет устроен схожим образом: для планеты задано простое число p_i — «основание» календаря. Неделя на каждой планете состоит из 7 дней, называющихся так же, как и на Земле, месяц — из p_i недель, год — из p_i месяцев.

В календаре существуют также високосные годы. Год является високосным, если его номер делится на $(p_i)^3$ или же делится на p_i , но не делится на $(p_i)^2$. В високосные годы — в отличие от земных — последняя неделя каждого месяца с простым номером (месяца нумеруются с единицы) состоит из 8 дней — после воскресенья снова идёт воскресенье.

Длина суток на всех планетах согласована с земной, то есть понятие «те же сутки, что и на Земле» корректно определено и имеет смысл. Известно, что когда на земле было 1 января 2000 года, на всех k планетах системы был первый день первого месяца первого года, понедельник. Ваша задача по заданной дате (число, месяц, год) на Земле вычислить, какой день недели будет на каждой из k планет системы.

Напоминаем, что на Земле обычный год состоит из 365 дней, в январе 31 день, в феврале 28, в марте 31, в апреле 30, в мае 31, в июне 30, в июле 31, в августе 31, в сентябре 30, в октябре 31, в ноябре 30, в декабре 31. Каждый год, кратный 400 или кратный 4, но не кратный 100, является високосным, то есть содержит 29 дней в феврале, итого 366 дней. Также напоминаем, что сегодня воскресенье, 27 сентября 2009 года.

Формат входного файла

В первой строке входного файла задано натуральное число d ($1 \leq d \leq 100$) — количество тестовых примеров.

Каждый тестовый пример состоит из двух строк. В первой задано число k ($1 \leq k \leq 8$) — количество планет в системе, последующие k целых чисел задают p_i ($2 \leq p_i < p_{i+1} \leq 50$) для каждой планеты. Все p_i являются простыми. Во второй задана дата земного календаря в формате $d\ mmm\ yyyu$, где d — день месяца, mmm — буквенное обозначение месяца (`jan` — январь, `feb` — февраль, `mar` — март, `apr` — апрель, `may` — май, `jun` — июнь, `jul` — июль, `aug` — август, `sep` — сентябрь, `oct` — октябрь, `nov` — ноябрь и `dec` — декабрь), $yyyu$ — номер года ($2000 \leq yyyu \leq 3000$).

Формат выходного файла

Для каждого тестового примера в отдельной строке выведите $k+1$ трёхсимвольных сокращений (`mon` — понедельник, `tue` — вторник, `wed` — среда, `thu` — четверг, `fri` — пятница, `sat` — суббота, `sun` — воскресенье) для дня недели на Земле (первая аббревиатура) и на k планетах (аббревиатура с $i+1$ -м номером относится к i -й планете).

Пример

calendar.in	calendar.out
3	sat mon mon mon mon
4 5 7 11 13	thu sat thu sat sat
1 jan 2000	wed fri thu fri fri
4 2 7 19 31	
10 feb 2084	
4 11 23 43 47	
25 jul 2987	

Задача К. Magic Temple (Division 2 Only!)

Имя входного файла: `k.in`
Имя выходного файла: `k.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 Mebibytes

Комиссия добралась до исторического факультета. Там на семинаре по истории древнейших народов Сибири рассматривался сложный вопрос датировки некоторого события.

Дело в том, что в одной древней рукописи упоминался магический храм, возводимый каждый Новый Год сибирскими шаманами в честь могущественного духа-покровителя племени. Храм строился из льда и имел форму правильного многоугольника, причём каждый год количество сторон в храме увеличивалось на 1.

Далее в этой рукописи было сказано, что в некий год храм был захвачен воинственными пришельцами с юга, и вождь победившего племени повелел разделить храм внутренними стенами на равнобедренные треугольники — тотемные знаки своего племени, и треугольников тех получилось ровно на 2 меньше, чем было в тот год в храме сторон...

У историков есть информация о количестве внешних стен в храме в некоторый год по летоисчислению племени, также им известно, что завоевание произошло не ранее этого года.

Председатель комиссии, впечатлённый увиденным на естественнонаучных факультетах, сказал, что партия учит искать истину на стыке наук, и порекомендовал историкам обратиться к математикам — возможно, они помогут вычислить год завоевания.

Вычислите ближайший год, в который завоевание могло произойти.

Формат входного файла

Во входном файле заданы два целых числа N и C , где N — количество стен в храме в году C по летосчислению племени ($3 \leq N \leq 10^{1000}$, $0 \leq C \leq 10^{1000}$).

Формат выходного файла

Выведите одно число — ближайший год, в который могло произойти завоевание.

Примеры

<code>k.in</code>	<code>k.out</code>
6 2009	2009
7 2009	2010

Задача L. DNA (Division 2 Only!)

Имя входного файла:	1.in
Имя выходного файла:	1.out
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 Mebibytes

Последним факультетом, который посетила комиссия, был биологический факультет. Дело в том, что доминирование сторонников академика Лысенко в биологии всё ещё продолжалось, и члены комиссии опасались, что высказанные ими идеи могут быть истолкованы как поддержка опальной генетики. И всё-таки один из проверяющих — тот самый спортивный куратор — привлечённый запахом спирта, добрался до биофака.

Там биологи проводили эксперимент с модификацией ДНК колоний бактерий, выращиваемых в расположенных в ряд чашках. С помощью изменений ДНК учёные добились того, что бактерии реагируют на плотность популяции на соседних чашках. Плотность популяции измеряется целыми числами от 0 до 3. Информация ДНК представляется в виде десяти целых чисел d_i , также лежащих в диапазоне от 0 до 3, и интерпретируется следующим образом:

- Обозначим за k сумму плотностей популяции в некоторой чашке и соседних с ней справа и слева. Тогда на следующий день в этой чашке плотность популяции будет равна d_k
- Для крайней слева и крайней справа чашки считается, что плотность популяции для соседней слева и соседней справа соответственно чашек равна 0.

Проверяющий ничего не понял из объяснений, а также из таблиц, и потребовал «просто, как на диаграмме и по ходам» нарисовать график состояния чашек с бактериями «по клеточкам». «А то дыхну — и все они у вас помрут» — полушутя-полусерьёзно заметил уже еле держащийся на ногах проверяющий.

Напишите программу, которая моделировала бы рост культуры бактерий, выращиваемых в течение 30 дней в ряду из 40 чашек, пронумерованных с 0 до 39, если изначально в чашке с номером 20 плотность популяции равна 1, а в остальных чашках — 0.

Формат входного файла

В первой строке входного файла задано натуральное число d ($1 \leq d \leq 100$) — количество тестовых примеров.

Каждый тестовый пример содержит 10 целых чисел d_i ($0 \leq d_i \leq 3$) — значения параметров информации ДНК.

Формат выходного файла

Для каждого тестового примера выведите схему распределения плотностей популяций в каждой из 40 чашек для каждого из 30 дней, начиная с первого дня. Вывод для каждого дня должен состоять из 40 символов. Каждая чашка представляется одним символом в соответствующем столбце. Нулевая популяция в чашке обозначается знаком '.', популяция 1 — знаком '!', популяция 2 — знаком 'x', популяция 3 — знаком 'W'. Разделяющую пустую строку между ответами к разным тестовым примерам выводить не требуется.

Пример

l.in
1 0 1 2 0 1 3 3 2 3 0
l.out
.....!!!!!x,x!!...!!!!!!!!!!x.....x!!..x.....x..!!!!xxx...xxx!!!!x!WW!x,x!WW!x!!...!xxW!.!Wxx!...!!!!..!WxW!!!WxW!.!!!!!x,xx!WWWW.WWWW!xx,x!!...!W!Wx.WWW..xW!W!...!!!!xWwXWwX.W.W.xWwXWwX!!!!!x..!WWWWWWW.W.WWWWWW!..x!!..x!x.....WW.WW.....x!x..!!!!x..!x.....WWWW.....x..!x!!! ...!x!..xx,xx...W..W...xx,xx..!x! ..!...!x!!!!x.....x!!!!x!!...! ..!!!x!!!...!x.....x!...!!!x!!!! !x,xx..!x!..!xx.....xx..!x!..xx,x ..!!!xx...xxx..!x.....x!..xxx...xx..!x ..!x,x..!x,x!W!.x!..x..x!x!W!x,x!..x! !..!x,x!..WW!...xx,xx...!WW!..!x,x!! !!!...!x!Wx!!..x!!!!x..!xW!x..!x,xx x,x!!!!x!WWW!x!x!..!x!x!WWW!x!...!!! x!..!...!x,xW!W..!...!..W!Wx,x!...!x,x ..x!!!!!!..!WwX!.!xxxxx!.!xWW!...!x!..!x ..x!...x!!!xWW.x.WWWW.x.WWx!!!!x!..x.. xx..!..x!..!WWwXWw..WwXWwW!..!...!xx.