

Задача А. Долгое путешествие в Одинцово

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Демид собирается в долгое путешествие в Одинцово на автобусах. Его маршрут состоит из n автобусных рейсов, пронумерованных от 1 до n в том порядке, в котором он должен их совершить. Автобусы ходят редко: i -й рейс курсирует только каждые x_i дней.

Более формально: в i -й автобус можно сесть только в дни $x_i, 2x_i, 3x_i, \dots$ (то есть в положительные целые кратные x_i дни). Поскольку автобусы быстрые, можно уехать по нескольким рейсам в один и тот же день.

Демид должен закончить своё путешествие не позже дня d , но он хочет как можно позже начать. Определите, в какой самый поздний день он может сесть на первый автобус (рейс 1), чтобы гарантированно завершить все рейсы не позднее дня d .

Гарантируется, что завершить путешествие к дню d возможно.

Формат входных данных

Первая строка содержит два целых числа n и d ($1 \leq n \leq 1000$, $1 \leq d \leq 10^{12}$) — количество рейсов и последний допустимый день завершения путешествия. В следующей строке находятся n целых чисел x_1, x_2, \dots, x_n ($1 \leq x_i$) — периодичности рейсов в днях.

Формат выходных данных

Выведите максимальный номер дня, в который Демид может сесть на первый автобус и всё же успеть завершить путешествие не позднее дня d .

Примеры

стандартный ввод	стандартный вывод
3 10 3 7 2	6
3 10 3 6 2	6

Задача В. Ломать Дейкстру

Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Каждый специалист в области информатики должен быть знаком с алгоритмом Дейкстры: базовым алгоритмом для вычисления кратчайших путей от одной вершины в заданном графе. При выполнении алгоритма мы поддерживаем множество активных вершин. На каждой итерации мы выбираем и обрабатываем активную вершину с наименьшим текущим расстоянием. При обработке вершины мы исследуем исходящие ребра. Всякий раз, когда одно из них можно использовать для улучшения расстояния до смежной вершины, мы делаем это и помечаем эту смежную вершину как активную.

Известно, что алгоритм эффективен, когда все длины ребер неотрицательны. Эталонная реализация, представленная в этой задаче, работает за время $O(m \log n)$ для таких графов (где m — количество ребер, а n — количество вершин).

Ситуация усложняется, когда мы разрешаем ребра с отрицательной длиной. Нахождение кратчайшего простого пути (т.е. пути без повторяющихся вершин) в таком графе фактически является NP-трудной задачей. Некоторые версии алгоритма Дейкстры быстро завершаются для таких графов, но иногда они выдают некорректные результаты. Это не относится к нашей эталонной реализации. Наша эталонная реализация фактически решает немного другую проблему: для каждой вершины v мы ищем длину кратчайшей цепочки из начальной вершины в v . (Цепочка — это путь, который может содержать повторяющиеся вершины и ребра. Заметим, что если все длины ребер положительны, кратчайшая цепочка должна быть простым путём.)

Иногда не существует кратчайшей цепочки из начальной вершины в некоторую вершину v , потому что для каждой цепочки можно найти ещё более короткую. Для таких входных данных наша эталонная реализация никогда не завершается.

Мы хотим показать, что существуют графы, для которых алгоритм Дейкстры завершается, но его временная сложность экспоненциальна от количества вершин.

Эталонная реализация содержит переменную с именем `PROCESSED_VERTICES`. Для данного p постройте любой корректный график, для которого наша реализация завершится за конечное число шагов, и значение `PROCESSED_VERTICES` в конце будет в точности равно p .

Псевдокод алгоритма.

```
distance[v] := INF для всех вершин v
distance[0] := 0
PQ := приоритетная очередь пар (расстояние, вершина)
поместить (0, 0) в PQ
PROCESSED_VERTICES := 0

while PQ не пустая:
    (d, v) := извлечь вершину с минимальным расстоянием
    если d <> distance[v]:
        продолжить (это устаревшая запись)

    PROCESSED_VERTICES := PROCESSED_VERTICES + 1
```

```
    для каждого исходящего ребра (v -> u) длины w:
        если d + w < distance[u]:
            distance[u] := d + w
            поместить (distance[u], u) в PQ
```

вывести PROCESSED_VERTICES

Формат входных данных

Дано одно целое число p ($1 \leq p \leq 10^7$).

Формат выходных данных

Граф должен удовлетворять условиям:

- $1 \leq n \leq 60$,
- веса рёбер — целые числа, помещающиеся в 32-битный знаковый тип,
- граф ориентированный, без петель и кратных рёбер.

Описание графа должно состоять из $m + 2$ строк:

- в первой строке записано число n — количество вершин,
- во второй строке записано число m — количество рёбер,
- далее m строк, каждая из которых содержит три числа a_i, b_i, w_i — ребро из a_i в b_i весом w_i ($0 \leq a_i, b_i < n, a_i \neq b_i$).

Пример

стандартный ввод	стандартный вывод
6	11 7 1 3 10 0 7 12 7 10 -4 0 3 9 3 7 1 10 1 12 0 10 9

Задача С. Побег по коридору

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Ты — студент ФПМИ, который опоздал на пару и теперь пытается незаметно проскочить по главному коридору корпуса, чтобы не попасться на глаза преподавателям. Коридор можно представить как сетку: ты начинаешь движение из точки $(0, 0)$ и должен за n шагов добраться до конца коридора (координата $y = n$), двигаясь только вперёд.

Каждую секунду ты можешь сделать шаг:

- влево-вперёд — из (x, y) в $(x - 1, y + 1)$,
- прямо — в $(x, y + 1)$,
- вправо-вперёд — в $(x + 1, y + 1)$.

На некоторых уровнях y стоят преподаватели — каждый занимает целый отрезок от x_1 до x_2 (включительно) на своей линии обороны. Если ты окажешься на уровне y в позиции x , где $x_1 \leq x \leq x_2$, — ты попадёшься, и тебя отправят на пересдачу.

Твоя цель — пройти все n шагов, не попав ни в один отрезок. Если получится — выведи маршрут. Если нет — смирись и напиши «impossible».

Формат входных данных

В первой строке даны два целых числа n и m ($2 \leq n \leq 500\,000$ и $0 \leq m \leq 200\,000$) — количество шагов и количество преподавательских засад.

Следующие m строк содержат по три целых числа x_1, x_2, y ($-n \leq x_1 \leq x_2 \leq n$ и $1 \leq y < n$) — координаты отрезка, который занимает преподаватель на уровне y .

Гарантируется, что отрезки не пересекаются и даже не касаются друг друга.

Формат выходных данных

Если можно пройти незамеченным — выведи строку длины n , состоящую из символов:

- — шаг влево-вперёд,
- 0 — прямо,
- + — вправо-вперёд.

Если решений несколько — подойдёт любое.

Если пути нет — выведи «impossible».

Примеры

стандартный ввод	стандартный вывод
3 2 1 2 1 -2 0 2	0+-
3 2 1 2 1 -2 1 2	impossible

Задача D. Побег из общежития

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

На ФПМИ любят придумывать необычные головоломки. Рассмотрим прямоугольную сетку из r строк и c столбцов, которая задает общежитие, где каждая клетка соответствует одной комнате.

У каждой комнаты есть четыре двери, расположенные по сторонам света: север, юг, запад и восток.

- Двери на границе сетки ведут наружу.
- Все остальные двери соединяют соседние комнаты.

В игре участвует ровно $r \times c$ студентов, и каждый начинает в своей уникальной комнате. Когда все встанут на места, двери автоматически закрываются, и срабатывает механизм: в каждой комнате **ровно одна** дверь может открываться изнутри, остальные три заблокированы. Для каждой комнаты это фиксировано на всё время игры. Обратите внимание: дверь, соединяющая две комнаты, может открываться только с одной стороны или с обеих сторон — это зависит от конфигурации.

Каждый студент действует независимо:

- Он может проходить только через ту дверь, которая открывается в его текущей комнате.
- После прохождения двери он обязан закрыть её за собой.
- Студент идёт до тех пор, пока не выйдет наружу, либо пока не совершил $r \times c$ шагов. Если после $r \times c$ шагов студент не выбрался, он остаётся заперт.

Ваша задача — выбрать, какая дверь открывается в каждой комнате, так, чтобы **ровно k студентов смогли выбраться наружу**. Определите, возможно ли это. Если возможно, постройте корректный план.

Формат входных данных

Дана одна строка, содержащая три целых неотрицательных числа r, c, k ($0 \leq k \leq r \cdot c \leq 100$, $1 \leq r, c$).

Формат выходных данных

В первой строке выведите **Yes**, если решение существует, или **No**, если нет.

Если решение существует, далее выведите r строк по c символов — описание плана.

- j -й символ i -й строки соответствует комнате в i -й строке и j -м столбце сетки.
- Каждый символ — одна из букв **N, S, E, W**, обозначающая дверь, которую можно открыть (север, юг, восток или запад соответственно).

Если существует несколько правильных решений, можно вывести любое.

Примеры

стандартный ввод	стандартный вывод
2 3 2	Yes SES SNW
1 1 0	No

Задача Е. Первомайская

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

На бесконечной улице Первомайской, идущей с запада на восток, два студента, Рома и Вова, тренируются для забега на длинные дистанции. Они стартуют одновременно из одной точки и бегут на восток по следующему правилу:

- Рома сначала бежит t_1 минут со скоростью a_1 метров в минуту, затем t_2 минуты со скоростью a_2 метров в минуту и далее циклично повторяет этот режим.
- Вова сначала бежит t_1 минут со скоростью b_1 метров в минуту, затем t_2 минуты со скоростью b_2 метров в минуту и также повторяет этот цикл бесконечно.

Вам нужно определить, сколько раз Рома и Вова окажутся в одной точке одновременно (кроме начального момента времени). Если это произойдёт бесконечно много раз, выведите `infinity`.

Формат входных данных

Даны шесть целых чисел, по два числа на строке, $t_1, t_2, a_1, a_2, b_1, b_2$ — параметры забега ($1 \leq t_i \leq 10^5, 1 \leq a_i, b_i \leq 10^{10}$). Гарантируется, что $a_1 \neq b_1$ и $a_2 \neq b_2$.

Формат выходных данных

Выполните одно число — количество встреч. Если встреч будет бесконечно много, выведите слово `infinity`.

Примеры

стандартный ввод	стандартный вывод
1 2 5 5 6 2	1
10 1 11 11 12 1	<code>infinity</code>

Замечание

В первом тестовом случае Рома и Вова ровно один раз окажутся в одной точке одновременно. Произойдёт это в момент времени $1\frac{1}{3}$ в точке $6\frac{2}{3}$.

Задача F. Кампусы МФТИ

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

В МФТИ есть n кампусов, пронумерованных от 1 до n , и m дорог, соединяющих их попарно. Каждая дорога i ($1 \leq i \leq m$) соединяет кампусы u_i и v_i и является двусторонней.

Все кампусы распределены по k физтех-школам, каждый кампус j ($1 \leq j \leq n$) принадлежит физтех-школе s_j . Гарантируется, что каждая физтех-школа содержит хотя бы один кампус.

Ректор хочет провести q перевозок оборудования. k -я перевозка ($1 \leq k \leq q$) заключается в том, чтобы перевезти оборудование из кампуса a_k в кампус b_k по дорогам. Однако участвуют только физтех-школы s_{a_k} и s_{b_k} (если $s_{a_k} = s_{b_k}$, то только эта физтех-школа). Если во время маршрута груз проходит через кампус, принадлежащий другой физтех-школе, оборудование может быть повреждено.

Для каждой перевозки необходимо определить, существует ли путь между a_k и b_k , проходящий только через кампусы, принадлежащие школам s_{a_k} и s_{b_k} .

Формат входных данных

Первая строка содержит три целых числа n , m и k ($2 \leq n \leq 400\,000$, $1 \leq m \leq 400\,000$, $1 \leq k \leq n$).

Следующие m строк содержат по два целых числа u_i , v_i ($1 \leq u_i < v_i \leq n$), описывающих дороги между кампусами. Гарантируется, что каждая пара (u_i, v_i) встречается не более одного раза.

Следующая строка содержит n целых чисел s_1, s_2, \dots, s_n ($1 \leq s_j \leq k$) — номера физтех-школ, которым принадлежат кампусы. Гарантируется, что каждая физтех-школа содержит хотя бы один кампус.

Следующая строка содержит одно целое число q ($1 \leq q \leq 400\,000$) — количество перевозок.

Следующие q строк содержат по два целых числа a_k и b_k ($1 \leq a_k, b_k \leq n$, $a_k \neq b_k$) — физтех-школы, между которыми необходимо перевезти оборудование.

Формат выходных данных

Выполните q строк. Для каждой перевозки выведите **Yes**, если существует путь, проходящий только через кампусы физтех-школ s_{a_k} и s_{b_k} , и **No** в противном случае.

Пример

стандартный ввод	стандартный вывод
4 3 2	Yes
1 2	No
2 3	Yes
3 4	
1 2 1 2	
3	
1 2	
1 3	
1 4	

Задача G. Стена из полимино

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дана прямоугольная стена размером $r \times c$, составленная из n полимино (каждый полимино состоит из одной или нескольких клеток, соединённых по стороне). Каждая клетка подписана заглавной буквой от A до Z, и все клетки одного полимино имеют одинаковую букву. Разные полимино имеют разные буквы.

Стена называется устойчивой, если существует такой порядок добавления полимино по одному, что при добавлении каждого полимино каждая его клетка:

- либо стоит на земле (находится в последней строке),
- либо имеет под собой клетку, принадлежащую какому-либо уже добавленному полимино.

Требуется проверить, является ли стена устойчивой. Если да, выведите любой корректный порядок добавления полимино (строку из n различных букв).

Формат входных данных

Первая строка содержит два целых числа r и c ($1 \leq r, c \leq 30$) — количество строк и столбцов в стене; Следующие r строк содержат по c символов каждая, описывающих стену сверху вниз. Каждый символ — заглавная буква латинского алфавита (A–Z), обозначающая полимино.

Гарантируется, что из каждой буквы можно достичь любую клетку с этой буквой, перемещаясь только к соседним по стороне клеткам.

Формат выходных данных

Выполните строку из n различных букв, описывающую порядок добавления полимино, при котором стена получается устойчивой. Если такого порядка не существует, выведите -1. Если корректных порядков несколько, можно вывести любой.

Примеры

стандартный ввод	стандартный вывод
3 8 FFFPPPMII FPPPMMII FPPMMMMI	MPFI
5 3 XXX XAX XXX XBX XXX	-1

Задача Н. Торт

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В честь успешной сдачи экзамена по дискретной математике студентка ФПМИ Катя получила огромный кусок торта в форме прямоугольника с размерами $w \times h$. Торт лежит на плоскости так, что его вершины расположены в точках $(0, 0)$, $(w, 0)$, (w, h) и $(0, h)$.

Но её однокурсник Михаил решил немного повеселиться и начал постепенно откусывать торт. Его челюсть можно описать кругом радиусом r . В течение q минут происходят события двух типов:

1. $x_i \ y_i$: Михаил ставит свою челюсть так, чтобы центр круга находился в точке (x_i, y_i) и откусывает всё, что попадает внутрь круга или выше его. Соответствующая часть торта исчезает. Более формально, точка (x, y) будет съедена Михаилом, если существует $y' \leq y$, такая что точка (x, y') попадает внутрь или на границу круга.
2. x_i : Катя берёт линейку, ставит её вертикально в точке $x = x_i$ и измеряет длину оставшегося сегмента торта, который пересекает эту линию. Если торта не осталось в этой координате, она измерит 0.

Ваша задача — предсказать результаты всех измерений.

Формат входных данных

Первая строка содержит четыре целых числа w, h, r, q ($1 \leq w, h, r \leq 10^5$, $1 \leq q \leq 10^5$).

Следующие q строк описывают события. Событие типа 1 задаётся тремя целыми числами $1 \ x_i \ y_i$ — координаты центра укуса. Событие типа 2 задаётся двумя целыми числами $2 \ x_i$ — координата вертикальной линии, вдоль которой производится измерение.

Гарантируется, что $0 \leq x_i, y_i \leq 10^6$.

Формат выходных данных

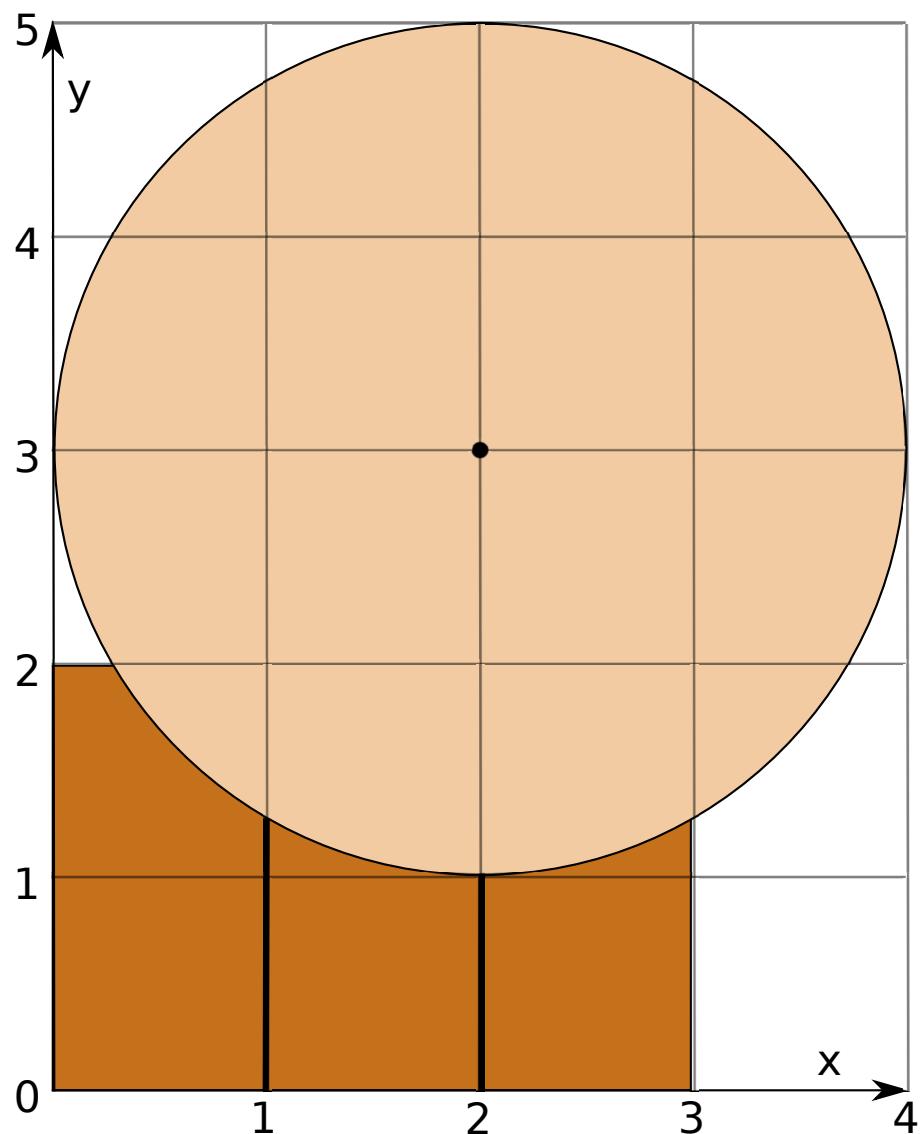
Для каждого события второго типа выведите одно действительное число — результат измерения. Ваш ответ будет принят, если абсолютная или относительная ошибка не превышает 10^{-6} .

Примеры

стандартный ввод	стандартный вывод
3 2 2 5 2 1 1 2 3 2 0 2 1 2 2	2.000000000000 2.000000000000 1.267949192431 1.000000000000
4 5 2 5 2 1 1 2 3 2 0 2 1 2 2	5.000000000000 3.000000000000 1.267949192431 1.000000000000

Замечание

Перед тем как Михаил откусил кусок торта, измеренная длина при любой x -координате была равна $h = 2$. После того как он откусил кусок, длина отрезка при $x = 0$ остаётся равной 2, при $x = 1$ она равна $3 - \sqrt{3} \approx 1.268$, а при $x = 2$ она равна 1 (см. рисунок).



Задача I. Нечетный подграф

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

В республике есть n городов, соединённых m двунаправленными дорогами. Каждая дорога соединяет два разных города a_i и b_i . Гарантируется, что нет повторяющихся дорог. Республика может быть не связной, то есть некоторые города могут быть недоступны друг из друга.

Администрация республики хочет устроить проверки инфраструктуры так, чтобы каждый город имел нечётное количество активных дорог, но при этом не обязательно использовать все дороги. Для этого они готовы временно закрывать некоторые дороги.

Ваша задача — определить, можно ли выбрать подмножество дорог так, чтобы в каждом городе оставалось нечётное число соединённых с ним дорог. Если это возможно, нужно вывести список дорог, которые останутся открытыми.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 10^5$) — количество городов и дорог.

Следующие m строк содержат по два целых числа a_i и b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$) — описание дорог.

Формат выходных данных

Если решения не существует, выведите одно число -1 .

Если решение существует, выведите число k — количество дорог, которые остаются активными, а затем k строк с номерами этих дорог (в порядке их появления во входных данных).

Примеры

стандартный ввод	стандартный вывод
6 5 1 2 2 3 3 4 4 1 5 6	3 3 1 5
5 6 1 2 2 3 3 1 3 4 4 5 5 3	-1

Задача J. Игра на буквенных строках

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Два игрока играют на строке. В начале перед ними пустая строка. По очереди игроки добавляют одну букву: либо в начало, либо в конец строки. После каждого хода, если в строке появляется одно из слов списка t , игрок, сделавший ход, проиграл. Также игрок проигрывает, если после его хода слово не является подстрокой ни одного из слов списка t .

Ваша задача определить, кто выигрывает при оптимальной игре обоих игроков.

Формат входных данных

Первая строка содержит число n ($1 \leq n \leq 200$) — количество слов в списке.

Следующие n строк содержат слова t_1, t_2, \dots, t_n . Каждое слово состоит из символов ASCII с кодами от 33 до 126. Длина каждого слова не превышает 50 символов. Большие и маленькие буквы следует различать.

Формат выходных данных

Выведите First, если первый игрок может выиграть при оптимальной игре, или Second, если выигрывает второй игрок.

Если первый игрок выигрывает, также нужно вывести все символы, с которых первый игрок может начать игру и выиграть. Символы выводятся в порядке возрастания их ASCII-кодов.

Пример

стандартный ввод	стандартный вывод
4 aa bbb ab cc	First ac

Задача К. Равномерное распределение задач

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

На ФПМИ готовят n задач для серии внутренних тренировочных контестов. Задачи пронумерованы целыми числами от 1 до n , при этом сложность i -й задачи равна ровно i условных единиц.

Организаторы хотят распределить все задачи между k контестами так, чтобы:

- все k контестов получили хотя бы по одной задаче;
- суммарная сложность задач в каждом контесте была одинаковой.

Ваша задача — определить, как распределить все задачи по k контестам, чтобы эти условия были выполнены.

Формат входных данных

Единственная строка содержит два целых числа n и k ($1 \leq n \leq 10^6$, $1 \leq k \leq n$) — количество задач и количество контестов.

Формат выходных данных

Если распределение невозможно, выведите единственную строку **No**. Иначе выведите в первой строке **Yes**, а затем k блоков, описывающих задачи каждого контеста. Каждый контест состоит из двух строк:

- первая строка содержит число c_i ($c_i \geq 1$) — количество задач в i -м контесте;
- вторая строка содержит c_i различных чисел $a_{i,1}, a_{i,2}, \dots, a_{i,c_i}$ — номера задач, выданных этой группе.

Если существует несколько корректных распределений, выведите любое из них.

Примеры

стандартный ввод	стандартный вывод
5 3	Yes 1 5 2 4 1 2 3 2
9 4	No

Задача L. Джекпот

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Недавно студент ФПМИ выиграл большой джекпот, но он не хочет тратить деньги на себя. Вместо этого он решил раздать их своим друзьям. Он знает, сколько денег у каждого из его друзей, и будет распределять деньги по следующему правилу, пока у него остаются деньги:

1. Вычисляется среднее арифметическое a — среднее количество денег у друзей.
2. Выбирается самый бедный друг. Если таких несколько, выбирается любой из них. Пусть p — текущее количество денег у выбранного друга.
3. Определяется x — минимальное целое число, такое что $p + x > a$.
4. Если у студента осталось хотя бы x рублей, он даёт x выбранному другу. Иначе он отдаёт этому другу все оставшиеся деньги, и процесс завершается.

Требуется определить, сколько денег будет у каждого друга после окончания раздачи.

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 47$) — количество друзей.

Вторая строка содержит n целых чисел m_1, m_2, \dots, m_n ($1 \leq m_i \leq 10^{18}$) — начальные суммы денег у каждого друга.

Третья строка содержит целое число j ($1 \leq j \leq 10^{18}$) — размер джекпота.

Формат выходных данных

Выполните n чисел — количество денег у каждого друга после завершения распределения, в порядке **неубывания**.

Пример

стандартный ввод	стандартный вывод
4 4 44 7 77 47	24 34 44 77

Замечание

Изначально среднее значение равно 33, поэтому студент передаст 30 рублей первому другу, и у него останется 17 рублей.

Затем среднее значение станет равным 40.5, и студент передаст оставшиеся деньги третьему другу.

Задача М. Шкафы

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

На кафедре АТП стоит n больших шкафов, расставленных вдоль длинного коридора. Шкаф с номером i имеет высоту s_i и стоит в позиции p_i .

Студент, оказавшийся в коридоре, может перемещаться влево или вправо, но не может пройти сквозь шкаф — он останавливается перед ним. Однако, если он пробежал в выбранном направлении расстояние d , то разогнался достаточно, чтобы снести любой шкаф, высота которого строго меньше d . После этого он может продолжать движение и сносить другие шкафы по тому же правилу. В процессе студент может разворачиваться и отбегать в другую сторону, чтобы разогнаться.

Студент считается выбравшимся, если в конце концов он сносит самый левый или самый правый шкаф в коридоре и покидает здание. Ваша задача — вычислить суммарную длину тех участков коридора, из которых студент не сможет выбраться, если будет действовать оптимально.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 10^5$) — количество шкафов. Каждая из следующих n строк содержит два целых числа s_i и p_i ($1 \leq s_i, p_i \leq 10^9$) — высоту шкафа и его позицию. Все позиции p_i различны.

Формат выходных данных

Выведите одно целое число — суммарную длину участков коридора, из которых студент не сможет выбраться, если изначально находится там.

Примеры

стандартный ввод	стандартный вывод
5 6 1 1 3 8 7 7 15 4 20	6
4 100 1 1 2 1 4 2 5	2

Замечание

Во втором примере студент не может выбраться, если его стартовая позиция между 1 и 2, или между 4 и 5.

Задача N. Фрукты

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вы хотите заработать на фруктах. Каждый фрукт можно продать за 100 рублей или посадить в качестве фруктового дерева. Фруктовое дерево через каждые три дня (начиная с третьего дня после посадки) приносит три новых фрукта.

Также существуют экзотические фрукты, которые можно купить в соседней деревне. Один раз в день можно потратить 400 рублей и купить один экзотический фрукт. Экзотический фрукт можно продать за 500 рублей или посадить в качестве экзотического фруктового дерева. Экзотическое дерево приносит три экзотических фрукта каждые три дня аналогично обычным фруктовым деревьям.

Любое количество фруктов или экзотических фруктов можно собирать, продавать и сажать в течение одного дня, но экзотический фрукт можно купить только один раз в день. Действия можно выполнять в любом порядке: например, собрать фрукты, продать их, купить экзотический фрукт и сразу посадить или продать его.

Ваша задача — определить максимальное количество рублей, которое вы можете заработать за оставшиеся d дней. Сейчас утро первого дня, у вас уже имеется r рублей, f фруктов, а также посажены некоторые деревья.

Формат входных данных

Одна строка содержит шесть целых чисел: d ($1 \leq d \leq 40$) — количество оставшихся дней, r ($0 \leq r \leq 500$) — текущее число рублей у вас, f ($0 \leq f \leq 100$) — текущее количество фруктов, t_0 — количество фруктовых деревьев, которые принесут урожай сегодня, t_1 — количество фруктовых деревьев, которые принесут урожай через один день, t_2 — количество фруктовых деревьев, которые принесут урожай через два дня ($0 \leq t_i \leq 100$).

Формат выходных данных

Выведите одно целое число — максимальное количество рублей, которое вы можете иметь после d дней.

Примеры

стандартный ввод	стандартный вывод
4 0 1 0 0 0	300
3 1 1 1 1 1	1301
2 400 0 0 0 0	600
40 399 0 0 0 0	399
5 0 1 0 1 0	1900