

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І
АРХІТЕКТУРИ**

ОСНОВИ ДИНАМІЧНОГО ПРОГРАМУВАННЯ

**МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ КУРСОВИХ РОБІТ ДЛЯ
СТУДЕНТІВ, ЯКІ НАВЧАЮТЬСЯ ЗА НАПРЯМОМ ПІДГОТОВКИ 122
«ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ»**

Київ 2018

УДК 004.92
ББК 32.973
Б19

Укладач: Є.В. Бородавка, доктор технічних наук

Рецензент: В.М. Міхайленко, доктор технічних наук, професор

Відповідальний за випуск: В.М. Міхайленко, доктор технічних наук,
професор, завідувач кафедри прикладної математики

Затверджено на засіданні кафедри прикладної математики, протокол №14 від 6 червня 2018 року.

Б19 Основи динамічного програмування: методичні вказівки до виконання курсових робіт / Уклад. **Є.В. Бородавка**. – К.: КНУБА, 2018. – 24 с.

Методична розробка містить загальні положення, які визначають мету й завдання курсової роботи та основні вимоги щодо її виконання, рекомендації щодо вибору теми, формування завдань, складання плану, викладу матеріалу, подання курсової роботи викладачу, структуру курсової роботи та вимоги, щодо її технічного оформлення, рекомендації щодо захисту, теми курсових робіт, а також список рекомендованої літератури.

Призначено для студентів, які навчаються за напрямом підготовки 122 «Інформаційні системи і технології».

УДК 004.92
ББК 32.973

© Бородавка Є.В. 2018

© КНУБА, 2018

Загальні положення

Курсове проектування є логічним продовженням лекційного курсу і лабораторних занять з дисципліни «Основи динамічного програмування» та сполучною ланкою для переходу від виконання навчальних завдань до проведення самостійної роботи за реальною тематикою.

Метою курсового проектування є формування в майбутніх спеціалістів професійних навичок та вмінь по розробці програм автоматизованого проектування і з використанням алгоритмів динамічного програмування.

В процесі виконання курсових робіт студенти повинні продемонструвати вміння застосувати на практиці теоретичні знання, отримані під час вивчення дисципліни.

Тематика курсових робіт

Тематика і зміст курсових робіт обумовлені основними розділами робочої навчальної програми дисципліни та орієнтовані на програмну реалізацію алгоритмів динамічного програмування.

Конкретну тему та опис початкових даних для розробки програми вказують у завданні на курсову роботу.

Порядок виконання курсової роботи

У відповідності з навчальним планом курсова робота передбачена на 5-му курсі у 10-му семестрі. На виконання роботи відводиться десять тижнів.

Тему курсової роботи студент вибирає самостійно чи за вказівкою викладача з урахуванням рівня підготовки студента. У випадку практичної участі студента в науково-дослідних роботах, що ведуться на кафедрі чи в університеті і мають безпосереднє відношення до дисципліни, студент може запропонувати власне формулювання теми курсової роботи, узгодивши її з викладачем.

Кожен студент виконує роботу *індивідуально*. В окремих випадках з дозволу викладача допускається об'єднання студентів у групи (2-3 студенти) для роботи над складними чи комплексними темами.

Послідовність виконання курсової роботи включає такі етапи:

- аналіз та чітка математична постановка задачі;
- розробка та опис структури даних геометричних об'єктів;
- розробка та опис структури програми;
- розробка та опис алгоритмів складових програмних функцій;
- створення файлів вхідних та контрольних даних;
- розробка та відлагодження програми геометричного моделювання та графічного виводу результатів;
- оформлення пояснювальної записки.

В процесі розробки програми рекомендується використовувати створені авторами базові програмні класи для роботи з структурами геометричних даних (текст програми подано в додатку).

Кінцевими результатами курсового проектування є файли контрольних даних, працююча програма та пояснювальна записка.

Пояснювальна записка

Пояснювальна записка повинна включати:

- вступ;
- математичну постановку задачі;
- опис структури даних геометричних об'єктів;
- опис програми з розділами у відповідності до стандартів «Єдиної системи програмної документації (ЕСПД)»: загальні відомості, призначення та умови застосування, структура та функції програми, опис алгоритму, вхідні та вихідні дані, повідомлення оператору;
- додатки (тексти програм, графічні зображення екранних форм, файли контрольних даних);
- список використаної літератури.

Формальний опис алгоритмів програм рекомендується виконувати з використанням діаграм об'єктно-орієнтованих програмних компонентів, операторних схем чи обробних систем. В останньому випадку узагальнений алгоритм програми геометричного моделювання може бути поданий в вигляді наступної обробної системи:

$$S = (I, T, O, F, \delta),$$

де: I – множина вхідних даних; T – множина внутрішніх (проміжних) даних; O – множина вихідних даних; F – множина програмних функцій (об'єктів) обробки даних; δ – стратегія застосування функцій, яка визначає послідовність виклику окремих $f_i \in F$.

Залежно від конкретної теми курсової роботи, до складу множини F можуть входити, наприклад, функції, що реалізують перетворення кадрування, обчислення коефіцієнтів рівняння площини за трьома точками, побудову триангульованої моделі поверхні, визначення просторового відношення полігонів тощо.

Загальний обсяг пояснювальної записки повинен складати **10-20** аркушів формату **A4**.

Теми курсових робіт

Problem B-[Advanced] Максимальный приз

Максимальный приз

Time limit: 1 s

Memory limit: 64 M

Победителю курса «APS2» в качестве награды предложили денежный приз N грн ($10 \leq N \leq 999999$). Он может увеличить или уменьшить его совершив ровно K ($0 \leq K \leq 10^9$) перестановок цифр в этом числе (переставлять можно две любые цифры в числе). Найдите максимальное значение денежного приза после K перестановок.

Например при $N = 123$ и $K = 1$, после перестановки первой и последней цифры можно получить число 321. Если $K = 2$, то максимальный приз будет 312.

Input format

Одна строка, которая содержит два числа: число N ($10 \leq N \leq 999999$) и число перестановок K ($0 \leq K \leq 10^9$) разделенные пробелом.

Output format

Максимальное значение денежного приза после K перестановок.

Notes

Input	Output
32888 2	88832

Problem D-[Advanced] Роботы 2

Роботы 2

Time limit: 2 s

Memory limit: 64 M

На поле квадратной формы, которое состоит из $R \times R$ клеток, находятся N роботов. Каждую клетку может занимать только 1 робот, робот находится ровно в центре клетки и его размер существенно меньше размера клетки. Каждый робот имеет пушку, которая может стрелять в любом направлении. Когда один из роботов рассматривается как атакующий, остальные могут быть либо открыты либо закрыты. Если несколько роботов находятся на одной линии с атакующим, то только ближайшие к атакующему являются открытыми, а остальные закрыты. Для каждого атакующего робота необходимо найти количество открытых роботов и вывести их сумму.

Input format

Первая строка содержит два числа: размер поля R ($5 \leq R \leq 1000$) и количество роботов N ($1 \leq N \leq 5000$) разделенные пробелом. Далее следует N строк, каждая содержит координаты i -того робота, разделенные пробелом.

Output format

Сумма открытых роботов для всех атакующих роботов.

Notes

Пример:

```
..3..
.2...
145..
.67..
....8
```

Для робота 1 открытыми будут: 2, 4, 6, 7, робот 3 закрыт роботом 2, робот 5 закрыт роботом 4, робот 8 закрыт роботом 7. Для робота 2: 1 3 4 5 7.

Input	Output
5 5 3 1 3 2 3 3 3 4 3 5	8
5 4 3 1 3 2 3 4 3 5	6
5 7 2 2 4 3 4 4 3 3 2 4 3 4 4 5	36

Problem F-[Advanced] Серийный номер 2

Серийный номер 2

Time limit: 1 s

Memory limit: 64 M

Компания, выпускающая мобильные телефоны, присваивает каждому изделию серийный номер, который вычисляется по следующей формуле:

$$(x_0 \cdot A + x_1 \cdot A^2 + x_2 \cdot A^3 + x_3 \cdot A^4 + x_4 \cdot A^5 + x_5 \cdot A^6) \bmod B$$

где x_0 – год выпуска ($2000 \leq x_0 \leq 3000$), x_1 – месяц выпуска ($1 \leq x_1 \leq 12$), x_2 – день выпуска ($1 \leq x_2 \leq 31$), x_3 – час выпуска ($0 \leq x_3 \leq 23$), x_4 – минута выпуска ($0 \leq x_4 \leq 59$), x_5 – секунда выпуска ($0 \leq x_5 \leq 59$), A – константа ($10 \leq A \leq 10^9$) $B = 10^9 + 7$

В связи с невнимательностью одного из сотрудников компании все телефоны, выпущенные 24.09.2015, оказались прошиты неправильной прошивкой. Ваша задача из списка серийных номеров телефонов отобрать все такие, которые, возможно, были выпущены 24.09.2015, и вывести количество этих телефонов.

Input format

Первая строка содержит два числа: константа A и количество серийных номеров, которые нужно проверить N ($1 \leq N \leq 1000000$) разделенные пробелом. Далее следует N серийных номеров, которые необходимо проверить, разделенные пробелом.

Output format

Число серийных номеров, возможная дата генерирования которых соответствуют 24.09.2015.

Notes

Пусть константа $A = 13$. Тогда телефон с серийным номером 183099332 возможно бракованный, так как телефон, выпущенный 24.09.2015 в 12:11:37 имел бы именно такой серийный номер.

$$183099332 = (2015 * 13 + 9 * 13^2 + 24 * 13^3 + 12 * 13^4 + 11 * 13^5 + 37 * 13^6) \% (10^9 + 7)$$

Input	Output
13 2 183099332 183099333	1

Problem H-[Advanced] Последовательность задач

Последовательность задач

Time limit: 1 s

Memory limit: 64 M

У вас есть список задач, которые необходимо сделать. Некоторые из задач можно сделать только когда определенные другие задачи уже сделаны. Такая зависимость задается определенным количеством пар задач, где вторая задача должна быть сделана только после первой. Напишите программу, которая будет сортировать задачи так, чтобы все условия выполнялась. По условию такая последовательность всегда существует (циклов нет).

Input format

Первая строка содержит два числа разделенные пробелом – число задач N ($1 \leq N \leq 100000$) и K - число пар, которые описывают последовательность выполнения задач ($1 \leq K \leq 100000$). В следующей строке идет K пар чисел, описывающих связи между задачами. Вторая задача в паре должна быть выполнена после первой задачи (индексы задач представлены числами от 1 до N включительно).

Output format

Индексы задач в таком порядке, в каком они должны выполняться, чтобы все условия были соблюдены. Если есть несколько правильных ответов можно вывести любой из них.

Notes

Input	Output
9 9 4 1 1 2 2 3 2 7 5 6 7 6 1 5 8 5 8 9	8 9 4 1 5 2 3 7 6

Правильными последовательностями будут: 8, 9, 4, 1, 5, 2, 3, 7, 6 или 4, 1, 2, 3, 8, 5, 7, 6, 9

Но такая последовательность будет неправильной: 4, 1, 5, 2, 3, 7, 6, 8, 9 т.к. задача 5 не может быть выполнена раньше, чем задача 8.

Problem J-[Advanced] Расчет максимальной скорости передачи данных

Расчет максимальной скорости передачи данных

Time limit: 1 s

Memory limit: 64 M

Для заданной компьютерной сети, состоящей из N узлов и K связей, найти максимальную скорость передачи данных из узла A в узел B , при условии, что каждая связь характеризуется своей максимальной скоростью передачи данных. Скорость передачи данных через узлы не ограничена и всегда (кроме узлов A и B) выполняется условие, что скорость, с которой информация поступает в узел, равна скорости, с которой информация выходит из него.

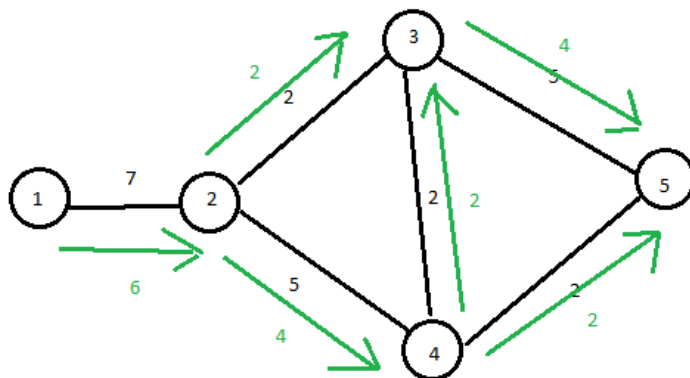
Input format

Первая строка содержит четыре числа разделенных пробелами – количество узлов N ($1 \leq N \leq 100$), количество связей K ($1 \leq K \leq 1000$), которые соединяют эти узлы, индексы передающего и принимающего узлов A, B ($A \neq B$). Далее идет K строк, где каждая строка содержит индексы двух соединенных узлов и скорость передачи информации между ними (индексы представлены числами от 1 до N включительно, скорость – число от 1 до 1000 включительно). Если узлы A и B соединены, то направление передачи данных между ними может быть произвольным, как от A к B , так и от B к A .

Output format

Максимальная скорость передачи данных из узла A в узел B .

Notes



Если задана такая сеть, то максимальная скорость передачи данных из узла 1 в узел 5 будет 6. В узел 2, например, информация поступает со скоростью 6 из узла 1, и выходит со скоростью 2 и 4 ($2 + 4 = 6$) в узлы 3 и 4 соответственно.

Input	Output
5 6 1 5 1 2 7 2 3 2 2 4 5 3 4 2 3 5 5 4 5 2	6

Problem L-[Advanced] Кратчайший путь

Кратчайший путь

Time limit: 1 s

Memory limit: 64 M

Вам задан неориентированный взвешенный граф, вершины которого пронумерованы от 1 до N. Ваша задача найти кратчайший путь от вершины 1 до вершины N и вывести его стоимость.

Кратчайший путь в графе – это такой набор ребер в графе, который соединяет стартовую и конечную вершины и имеет наименьшую сумму весов ребер среди всех других возможных путей между этими вершинами.

Input format

В первой строке задано числа N ($2 \leq N \leq 10^5$) и M ($1 \leq M \leq 10^5$) – количество вершин и ребер в графе соответственно.

Далее следует M строк, которые описывают каждое ребро в графе. Каждая строка содержит три целых числа u ($1 \leq u \leq N$), v ($1 \leq v \leq N$) и w ($1 \leq w \leq 10^7$) – номера двух вершин, которые соединяет ребро и вес этого ребра соответственно.

Output format

В единственной строке выведите одно число – сумму весов ребер на кратчайшем пути в графе от вершины с номером 1 до вершины с номером N.

Гарантируется, что между вершинами под номерами 1 и N всегда существует хотя бы один путь.

Notes

Input	Output
4 5 1 2 10 1 3 30 2 3 5 2 4 30 3 4 10	25
5 7 1 2 3 2 3 4 3 5 4 1 4 4 4 5 5 2 4 2 3 4 2	9

Пояснение к первому примеру

В заданном графе существует 4 простых (непересекающихся) пути:

1)	1	->	2	->	4	стоимостью	10	+	30	=	40				
2)	1	->	2	->	3	->	4	стоимостью	10	+	5	+	10	=	25
3)	1	->	3	->	4	стоимостью	30	+	10	=	40				
4)	1	->	3	->	2	->	4	стоимостью	30	+	5	+	30	=	65

Кратчайшим путем будет путь под номером (2), у которого наименьшая стоимость.

Problem N-[Advanced] Hanaro

Hanaro

Time limit: 1 s
Memory limit: 64 M

Вы принимаете участие в проекте Hanaro, задача которого объединить группу островов подводными туннелями. Ваша задача найти такую схему соединения островов, при которой все острова будут соединены между собой (непосредственно туннелем или через другие острова) и загрязнение окружающей среды будет минимальным. Все туннели прямые и всегда идут от одного острова к другому. Даже если туннели пересекаются, это происходит на разной глубине, и связи между ними нет. Загрязнение рассчитывается по формуле $L^2 \cdot E$, где L – длина туннеля, E – заданный коэффициент.

Input format

Первая строка содержит количество островов N ($1 \leq N \leq 1000$).

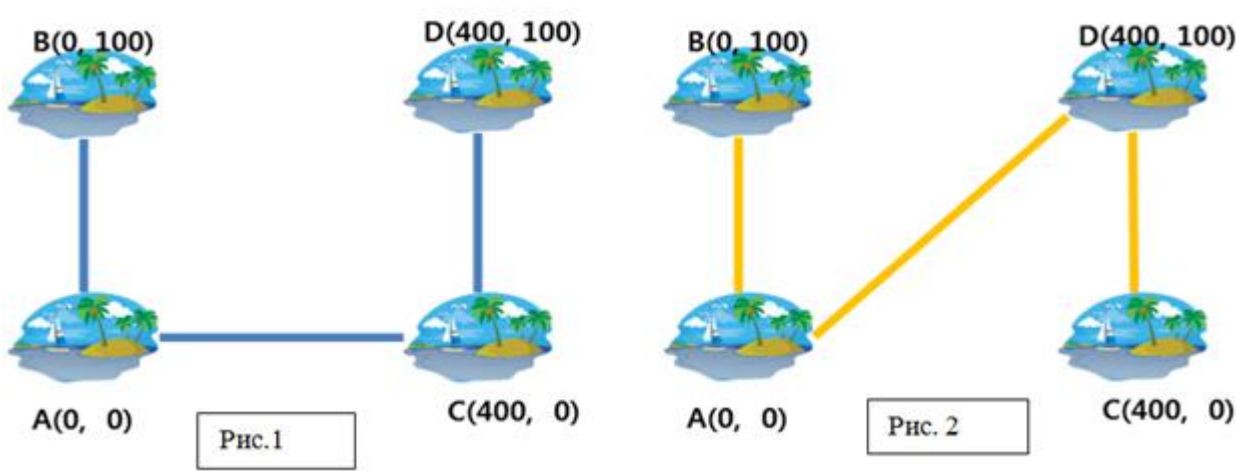
Вторая строка содержит N чисел, разделенных пробелами – x координаты всех островов ($0 \leq X \leq 1000000$).

Третья строка содержит N чисел, разделенных пробелами – y координаты всех островов ($0 \leq Y \leq 1000000$).

Output format

Суммарный уровень загрязнения оптимальной системы туннелей, округленный до ближайшего целого.

Notes



На рисунках показано два варианта соединения четырех островов. Видно, что первый вариант является оптимальным, так как при таком расположении туннелей загрязнение будет минимальным.

Input	Output
9 567 5 45674 24 797 29 0 0 0 345352 5464 145346 54764 5875 0 3453 4545 123 0.0005	27365366

Problem P-[Advanced] Симуляция падения блоков

Симуляция падения блоков

Time limit: 1 s

Memory limit: 64 M

Вы моделируете работу похожей на тетрис игры, в которой прямоугольные блоки падают сверху в колодец. Каждый блок опускается как одно целое до тех пор, пока все клетки, которые находятся под ним, свободны или пока он не достигает нижнего ряда. Ваша задача найти высоту конструкции, которая получится после падения заданного количества блоков. Высота всех блоков – 1, положение левого и правого конца блока задаются двумя числами. Обратите внимание, что полностью заполненные ряды не исчезают.

Input format

Первая строка содержит два числа разделенных пробелом – количество блоков N ($1 \leq N \leq 100000$), и ширина колодца K ($10 \leq K \leq 100000$). В следующей строке идет N пар чисел – каждая пара чисел описывает положение одного блока, его левого x_0 и правого x_1 конца ($0 \leq x_0, x_1 < K$).

Output format

Одно число – высота конструкции после падения всех блоков.

Notes

После падения такой последовательности блоков (1,2), (5,7), (2,2), (2,8), (0,9), (6,6), расположение блоков будет следующим

```
|           | 8
|           | 7
|           | 6
|      #    | 5
|#####| 4
|  #####  | 3
|   #      | 2
|  ##   ### | 1
|0123456789| 0
```

и соответственно высота конструкции – 5.

Input	Output
6 10 1 2 5 7 2 2 2 8 0 9 6 6	5

Problem R-[Advanced] Поиск подстроки по хешу

Поиск подстроки по хешу

Time limit: 1 s

Memory limit: 64 M

Вам дана последовательность чисел, где

$$x_0 = 1,$$

$$x_n = (x_{n-1} * A + B) \% 2^{32},$$

где $A = 1103515245$, $B = 12345$

И строка длиной N , полученная из этой последовательности следующим образом $c_n = 'a' + (x_n \gg 16) \% 26$, где c_n – это ASCII код n -ого символа в строке.

Начало этой строки выглядит так «aquzzptirwetbkelbhbdqmuhpfybxseirkcdronremebj».

Ваша задача найти в этой строке подстроку длиной K , чтобы хеш этой подстроки соответствовал заданному. Если таких подстрок несколько, вывести ближайшую к началу.

Хеш рассчитывается следующим образом. $\text{Hash} = (c_0 * C^0 + c_1 * C^1 + c_2 * C^2 + \dots + c_{K-1} * C^{K-1}) \% D$,

где $C = 1009$, $D = 1000000007$

Input format

Первая строка содержит три числа разделенных пробелами – количество символов в строке N ($1 \leq N \leq 1000000$), количество символов в подстроке, которая ищется, K ($1 \leq K \leq N$), значение хеша искомой подстроки H ($0 \leq H < D$).

Output format

Подстрока длиной K , хеш которой соответствует заданному или 0, если такая строка не найдена.

Notes

Input	Output
1000 10 536637247	segxxmemfx

Problem T-[Advanced] Поиск наибольшей общей подпоследовательности

Поиск наибольшей общей подпоследовательности

Time limit: 1 s

Memory limit: 64 M

Для заданных двух строк, состоящих из букв английского алфавита, найти длину наибольшей общей подпоследовательности и вывести ее значение. Наибольшая общая подпоследовательность – это такая строка, которая является подпоследовательностью обеих строк, при этом ее размер наибольший. Подпоследовательность – это такая строка, которую можно получить из некоторой заданной строки, если удалить из последней некоторое множество её элементов (возможно пустое). Например, ACDA является подпоследовательностью строки ABCDBAB, полученной после удаления всех букв B.

Input format

Две строки разделенные пробелом длиной от 1 до 1000 включительно.

Output format

Длина наибольшей общей подпоследовательности.

Notes

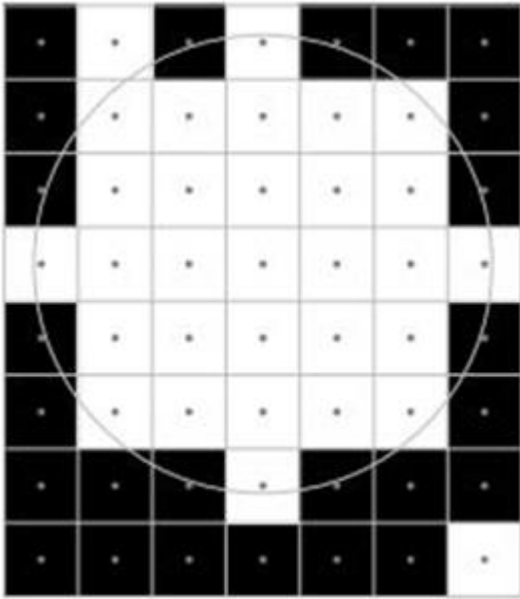
Для строк «RBKBGRBGGG» и «BGKRBRKBGB» наибольшая общая подпоследовательность будет «BKBRBG» и, соответственно, ее длина будет 6.

Input	Output
RBKBGRBGGG BGKRBRKBGB	6

Луна

Memory limit: 64 M

Напишите программу, которая по изображению найдет наибольший возможный радиус круга, который соответствует Луне.



Notes

Input	Output
7 8 . * . * * * * * * . . * * * * * . * * * * * * * . * * * * * . . * * * * * * *	3
5 4 * . . . * . . .	0

Problem BG-[Advanced] ColoredMutants

Разноцветные мутанты

Time limit: 1 s

Memory limit: 64 M

Уже долгое время в Институте Искусств, Мутантов и Информационных Технологий разводят милых разноцветных зверюшек. Для удобства каждый цвет обозначен своим номером, всего цветов не более 10^9+7 . В один из прекрасных дней в питомнике случилось чудо: все зверюшки выстроились в ряд. Пользуясь случаем, лаборанты решили посчитать, сколько зверюшек разных цветов живёт в питомнике, и, по закону жанра, попросили вас написать программу, которая поможет им в решении этой нелёгкой задачи.

Input format

В первой строке входного файла содержится единственное число N ($0 \leq N \leq 10^5$) - количество зверюшек в Институте. В следующей строке находится N неотрицательных целых чисел, строго меньше 10^9+7 и разделённых пробелами - их цвета. В третьей строке файла записано число M ($1 \leq M \leq 100000$) - количество запросов вашей программе, в следующей строке через пробел записаны M целых неотрицательных чисел (строго меньше 10^9+7).

Output format

Выходной файл должен содержать M чисел, разделённых пробелами. Для каждого запроса выведите число зверюшек заданного цвета в питомнике.

Notes

Input	Output
10 18 1 3 5 7 9 18 57 1 3 5 57 3 9 1 179	1 2 1 2 0

Problem BZ-Медиана двух последовательностей

Медиана двух последовательностей

Time limit: 1 s

Memory limit: 64 M

Программист Алёша играет с линейными конгруэнтными генераторами. Операция «mod» слишком сложна для него, поэтому каждое i -ое число (нумерация с 0) он генерирует простой формулой:

$$X[i] = A * i + C$$

где A и C – некоторые константы.

Алёша заметил, что если взять неотрицательные целые числа A и C для генерации N чисел, то последовательность X получится отсортированной.

Однако такая последовательность слишком однообразна, поэтому Алёша выбрал два разных генератора, каждым сгенерировал некоторую последовательность, после чего склеил полученные числа в один массив и отсортировал его.

Теперь его интересует значение медианы в полученном отсортированном массиве, однако массив настолько большой, что сам он не может её найти. Алёша не скрывает своих генераторов и просит Вас найти такую необходимую ему медиану. Помогите Алёше с этой задачей.

Медиана – это значение, которое находится строго в центре отсортированной последовательности. Если в последовательности чётное количество элементов, то это среднее арифметическое двух центральных элементов (см. примеры).

Детальнее про медиану: [https://ru.wikipedia.org/wiki/Медиана_\(статистика\)](https://ru.wikipedia.org/wiki/Медиана_(статистика))

Input format

В первой строке заданы целые числа A_1, C_1, N_1 ($0 \leq A_1, C_1 \leq 10^9, 1 \leq N_1 \leq 10^9$) – константы A и C для генератора первой последовательности и количество сгенерированных чисел первым генератором соответственно.

Во второй строке заданы целые числа A_2, C_2, N_2 ($0 \leq A_2, C_2 \leq 10^9, 1 \leq N_2 \leq 10^9$) – константы A и C для генератора второй последовательности и количество сгенерированных чисел вторым генератором соответственно.

Output format

Одно число – медиана полученной последовательности.

Обратите внимание, что ответ может быть не целым числом, поэтому выведите его с двумя знаками после десятичной точки.

Notes

Input	Output
1 1 2 1 3 2	2.50
7 13 7 1 22 6	26.00
123 20 1000 970 31 2000	484546.00

Пояснение к первому примеру:

Последовательность 1: $[1 * 0 + 1, 1 * 1 + 1] = [1, 2]$

Последовательность 2: $[1 * 0 + 3, 1 * 1 + 4] = [3, 4]$

Полученный массив: $[1, 2] + [3, 4] = [1, 2, 3, 4].\text{sort}() = [1, 2, 3, 4]$

Медиана: $(2 + 3) / 2 = 2.50$

Пояснение к второму примеру:

Последовательность 1: $[7 * 0 + 13, 7 * 1 + 13, \dots] = [13, 20, 27, 34, 41, 48, 55]$

Последовательность 2: $[1 * 0 + 22, 1 * 1 + 22, \dots] = [22, 23, 24, 25, 26, 27]$

Полученный массив: $[22, 23, 24, 25, 26, 27] + [13, 20, 27, 34, 41, 48, 55]$

Полученный массив: $[22, 23, 24, 25, 26, 27, 13, 20, 27, 34, 41, 48, 55].\text{sort}()$

Полученный массив: $[13, 20, 22, 23, 24, 25, 26, 27, 27, 34, 41, 48, 55]$

Медиана: 26.00