

## Содержание

<b>Must have</b>	2
<b>Задача 18А. Happy Three Friends [0.1 sec, 256 mb]</b>	2
<b>Обязательные задачи</b>	3
<b>Задача 18В. Арифметическая прогрессия [0.2 sec, 256 mb]</b>	3
<b>Задача 18С. Альфа Дерево [1.5 sec, 256 mb]</b>	4
<b>Для искателей острых ощущений</b>	5
<b>Задача 18D. Первообразный корень [0.1 sec, 256 mb]</b>	5
<b>Задача 18Е. Корни [0.2 sec, 256 mb]</b>	6
<b>Задача 18F. Длинная дорога [0.4 sec, 256 mb]</b>	7
<b>Задача 18G. Необычный случай [3.0 sec, 256 mb]</b>	8

---

У вас не получается читать/выводить данные?  
Воспользуйтесь примерами (c++) (python).

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

Обратите внимание на GNU C++ компиляторы с суффиксом inc.

Подни можно пользоваться **дополнительной библиотекой** (optimization.h).

То есть, использовать быстрый ввод-вывод: **пример про числа и строки**.

И быструю аллокацию памяти (ускоряет vector-set-map-весь-STL): **пример**.

Для тех, кто хочет разобраться, как всё это работает.

Короткая версия быстрого ввода-вывода (**тык**) и короткая версия аллокатора (**тык**).

## Must have

### Задача 18А. Happy Three Friends [0.1 sec, 256 mb]

Три весёлых друга расплющены в лепёшки. Каждая лепёшка имеет форму идеального круга. Координаты центра круга от 0 до 1, радиус круга от 0 до 1. Ваша задача – найти площадь части плоскости, покрытой всеми тремя друзьями.

#### Формат входных данных

Три строки, каждая содержит по три вещественных числа  $x_i$   $y_i$   $r_i$ .

#### Формат выходных данных

Выведите ответ с абсолютной погрешностью не более 0.01.

#### Примеры

stdin	stdout
0.0 0.0 1.0 1.0 1.0 1.0 0.0 1.0 1.0	0.442960

## Обязательные задачи

### Задача 18В. Арифметическая прогрессия [0.2 sec, 256 mb]

Однажды Петя узнал очень важную последовательность из  $n$  чисел. Тщательно проанализировав ее, он обнаружил, что она является арифметической прогрессией. Чтобы не забыть он записал ее элементы на  $n$  карточках.

Но затем случилась неприятность. Не зная всю важность этой последовательности, его брат Вовочка взял еще  $n$  карточек и написал на них произвольные числа, а потом перемешал все  $2n$  карточек.

Теперь Петя хочет восстановить исходную последовательность по этим карточкам. К сожалению возможно, что это можно сделать несколькими способами, но Петю устроят любые  $n$  чисел, образующие арифметическую прогрессию.

Петя не может сделать это вручную, поэтому обратился к вам за помощью.

Напомним что последовательность  $a_1, a_2, \dots, a_n$  называется арифметической прогрессией, если  $a_i = a_{i-1} + d$  для всех  $i$  от 2 до  $n$  и некоторого  $d$ . Число  $d$  называется *разностью* арифметической прогрессии.

#### Формат входных данных

В первой строке входного файла находится целое число  $n$  ( $1 \leq n \leq 100\,000$ ). В следующей строке находится  $2n$  целых чисел по модулю не превосходящих  $10^9$  — числа, написанные на карточках, перечисленные в произвольном порядке. Гарантируется, что можно выбрать  $n$  из них так, чтобы они образовывали арифметическую прогрессию.

#### Формат выходных данных

В первой строке выходного файла выведите  $a_1$  и  $d$  — первый элемент и разность найденной арифметической прогрессии. Если  $d = 0$ , число  $a_1$  должно встречаться среди заданных чисел  $n$  раз.

Если существует несколько решений, выведите любое.

#### Примеры

stdin	stdout
3 8 7 1 5 4 3	1 3

**Задача 18С. Альфа Дерево [1.5 sec, 256 mb]**

У вас есть полное бинарное дерево глубины  $n$  ( $0 \leq n \leq 32$ ).

В дереве  $2^n$  листьев, они пронумерованы слева направо числами от 0 до  $2^n - 1$ .

В  $i$ -м листе записано число  $x_i = (ai^2 + bi + c) \bmod m$ .

Есть фишка, которая изначально находится в корне дерева. Двое играют в игру, двигая фишку вниз по дереву. Когда фишка достигает листа дерева, игра заканчивается. Цель первого игрока – максимизировать число в листе, цель второго – минимизировать.

**Формат входных данных**

Числа  $n, a, b, c, m$ . При этом  $10 \leq m \leq 10^9$ .

Все  $a, b, c$  сгенерированы равномерным распределением на  $[0, m)$ .

**Формат выходных данных**

Выведите результат игры при оптимальной игре обоих.

**Примеры**

stdin	stdout
3 10 7 9 20	11

**Замечание**

Взятие остатка по модулю – небыстрая операция. Чем их меньше, тем лучше.

## Для искателей острых ощущений

### Задача 18D. Первообразный корень [0.1 сек, 256 mb]

Дано простое число  $p$ , найти первообразный корень  $g$ .

Первообразный корень – число, порождающее мультипликативную группу кольца вычетов по модулю  $p$ :  $\langle 1, g, g^2, \dots, g^{p-2} \rangle = (\mathbb{Z}/p\mathbb{Z})^*$ . Известно, что для любого  $p$  такое  $g$  существует.

#### Формат входных данных

Мультитест. Тестов не более 20.

Каждый тест – число  $2 \leq p \leq 10^9$ ,  $p$  простое.

#### Формат выходных данных

Для каждого теста выведите любое подходящее  $g$ .

#### Примеры

stdin	stdout
7	3
2	1

**Задача 18Е. Корни [0.2 sec, 256 mb]**

Дано целое число  $n \geq 1$ . Нужно найти такое  $g$ , что для любого  $a$ :  $\gcd(a, n) = 1, 1 \leq a < n \implies \exists$  целое  $x: g^x = a \pmod n$ . Напомним, что  $\gcd(a, b)$  — наибольший общий делитель чисел  $a$  и  $b$ .

**Формат входных данных**

Внимание, мультитест!

На каждой строке число  $n$  ( $2 \leq n \leq 10^{12}$ ).

Сколько тестов, мы вам не скажем, но все в рамках приличия.

**Формат выходных данных**

Для каждого  $n$  на отдельной строке выведите  $g$  ( $1 \leq g < n$ ) или  $-1$ , если такого  $g$  не существует.

**Примеры**

stdin	stdout
5	2
10	3
9	2
15	-1

**Задача 18F. Длинная дорога [0.4 сек, 256 mb]**

Дорога, дорога, осталось немного...

---

Дан случайный неориентированный граф  $G$  из  $n$  вершин и  $m$  ребер. Ваша задача — найти гамильтонов путь. Гарантируется, что гамильтонов путь в графе есть.

**Формат входных данных**

На первой строке число вершин  $n \geq 2$  и число ребер  $m \geq 1$ .

Следующие  $m$  строк содержат пары чисел от 1 до  $n$  — ребра графа.

В графе нет ни петель, ни кратных ребер.

Поскольку почти полный граф — совсем не интересный тест,  $m \leq 500$ .

**Формат выходных данных**

На первой строке выведите  $n$  различных чисел от 1 до  $n$  — вершины гамильтонового пути в порядке прохода по ним. Начинать и заканчивать можно в любой вершине. Если гамильтоновых путей несколько, выведите любой.

**Система оценки**

Подзадача 1 (20 баллов)  $n \leq 26$ .

Подзадача 2 (20 баллов)  $n \leq 35$ .

Подзадача 3 (20 баллов)  $n \leq 50$ .

Подзадача 4 (20 баллов)  $n \leq 70$ .

Подзадача 5 (20 баллов)  $n \leq 100$ .

**Примеры**

stdin	stdout
5 8 3 1 2 5 5 4 3 4 1 4 3 5 3 2 1 2	1 4 3 5 2

**Задача 18G. Необычный случай [3.0 сек, 256 mb]**

Сэр Гамильтон любит длинные прогулки...

---

Дан случайный неориентированный граф из 10 000 вершин, 200 000 рёбер и число  $k$ . Ваша задача — найти в данном графе  $k$  непересекающихся гамильтоновых путей.

Путь называется гамильтоновым, если проходит по всем вершинам графа ровно по одному разу.

**Формат входных данных**

На первой строке число вершин  $n$ , число рёбер  $m$ , количество путей, которое нужно найти  $k$ . В данной задаче  $n = 10\,000$ ,  $m = 200\,000$ ,  $1 \leq k \leq 200\,000$ . Далее  $m$  строк содержат описание рёбер графа. Ребро описывается парой чисел от 1 до  $n$  — номера вершин, которые соединяет ребро. В графе нет ни петель, ни кратных рёбер.

В этой задаче кроме примера имеется ровно 30 тестов.

**Формат выходных данных**

Выведите  $k$  строк. На каждой строке выведите гамильтонов путь — последовательность из  $n$  вершин в порядке прохода. Каждое ребро графа должно быть использовано не более чем в одном из выведенных путей.

**Примеры**

stdin	stdout
5 9 2	1 3 5 2 4
1 3	5 1 4 3 2
1 4	
1 5	
2 3	
2 4	
2 5	
3 5	
4 3	
5 4	

**Замечание**

Пример в условии — единственный тест, где  $n \neq 10\,000$ ,  $m \neq 200\,000$ ,  $k \neq 200\,000$ .

Он нужен, чтобы продемонстрировать формат данных.

В тестирующей системе тест из примера есть.