

## Содержание

<b>Must have</b>	<b>2</b>
<b>Задача 24А. Happy Three Friends [0.1, 256]</b>	<b>2</b>
<b>Задачи здорового человека</b>	<b>3</b>
<b>Задача 24В. Matrix Multiplication [1.5, 256]</b>	<b>3</b>
<b>Задача 24С. Альфа Дерево [1.5, 256]</b>	<b>4</b>
<b>Для искателей острых ощущений</b>	<b>5</b>
<b>Задача 24D. Арифметическая прогрессия [0.2, 256]</b>	<b>5</b>
<b>Задача 24Е. Длинная дорога [0.4, 256]</b>	<b>6</b>

---

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 24А. Happy Three Friends [0.1, 256]

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

#### Подсказка по решению

Разобрана и на лекции, и на практике.

## Задачи здорового человека

### Задача 24В. Matrix Multiplication [1.5, 256]

Маленький Джошуа учится умножать матрицы. Он тренируется, умножая большие бинарные матрицы в  $\mathbb{F}_2$  (все арифметические операции производятся по модулю два). Недавно он перемножил две матрицы  $A$  и  $B$  размера  $n \times n$  и получил результат  $C$ .

Дженни не верит ему и говорит, что Джошуа ошибся. Помогите детям понять, кто прав. Даны матрицы  $A, B, C$ , проверьте, что  $AB = C$ .

#### Формат входных данных

На первой строке число  $n$  — размер матриц ( $1 \leq n \leq 4000$ ).

Следующие три строки содержат описания матриц  $A, B$  и  $C$ .

Каждая матрица описывается строкой, содержащей  $n$  блоков размера  $\lceil n/4 \rceil$  16-ричных цифр. Если записать цифры в двоичной записи в данном порядке от старших цифр к младшим и обрезать лишние цифры в конце строки, получится очередная строка матрицы. Например, матрица

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

описывается как 28, A8, 68, 78, D0, 88.

#### Формат выходных данных

Выведите “YES” если  $AB = C$ , иначе “NO”.

#### Пример

stdin	stdout
6 28 A8 68 78 D0 88 80 40 20 10 08 04 28 A8 68 78 D0 88	YES
6 28 A8 68 78 D0 88 80 40 20 10 08 04 28 B8 68 78 D0 88	NO

#### Подсказка по решению

Умножать матрицы за куб — это точно не то, что вам нужно.

А вот умножить матрицу на вектор можно за квадрат.

На какой же вектор умножать?

### Задача 24С. Альфа Дерево [1.5, 256]

У вас есть полное бинарное дерево глубины  $n$  ( $0 \leq n \leq 30$ ).

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

#### Подсказка по решению

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

Разобрана на практике.

## Для искателей острых ощущений

### Задача 24D. Арифметическая прогрессия [0.2, 256]

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	1 3
8 7 1 5 4 3	

#### Подсказка по решению

Ткните в 3 элемента. Подумайте о хорошем. Что вы видите?

**Задача 24Е. Длинная дорога [0.4, 256]**

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.

**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

**Подсказка по решению**

Что-то на тему Random Walk.