

Лабораторная работа №1

Задачи приведены практически без изменений в том виде, в котором они были взяты из онлайн-источников, включая язык. Все права на тексты и тестовые наборы задач `cards` и `diploma` принадлежат их авторам, на текст и тестовый набор задачи `polynomial` – организации TopCoder, Inc.

Проверка работ происходит автоматически. Придерживайтесь условий задач и ограничений ввода-вывода.

Формальные требования: файл программы называется так же, как и задача, как и входные и выходные файлы к задаче. К примеру, файл программы к задаче `cards` – это `cards.cpp/c/pas/dpr`. Первая строчка файла программы содержит комментарий следующего вида:

- `/* GCC */` – для C++.
- `{ Free Pascal }` – для Pascal.
- `{ Turbo Delphi }` – для Delphi.

Задача «Дипломы»

Входной файл: `diploma.in`

Выходной файл: `diploma.out`

Ограничение по времени: 4 с

Ограничение по памяти: 64 МБ

Когда Петя учился в школе, он часто участвовал в олимпиадах по информатике, математике и физике. Так как он был достаточно способным мальчиком и усердно учился, то на многих из этих олимпиад он получал дипломы. К окончанию школы у него накопилось n дипломов, причём, как оказалось, все они имели одинаковые размеры: w – в ширину и h – в высоту.

Сейчас Петя учится в одном из лучших российских университетов и живёт в общежитии со своими одногруппниками. Он решил украсить свою комнату, повесив на одну из стен свои дипломы за школьные олимпиады. Так как к бетонной стене прикрепить дипломы достаточно трудно, то он решил купить специальную доску из пробкового дерева, чтобы прикрепить её к стене, а к ней – дипломы. Для того чтобы эта конструкция выглядела более красиво, Петя хочет, чтобы доска была квадратной и занимала как можно меньше места на стене. Каждый диплом должен быть размещён строго в прямоугольнике размером w на h . Дипломы запрещается поворачивать на 90 градусов. Прямоугольники, соответствующие различным дипломам, не должны иметь общих внутренних точек.

Требуется написать программу, которая вычислит минимальный размер стороны доски, которая потребуется Пете для размещения всех своих дипломов.

Формат входного файла

Входной файл содержит три целых числа: w, h, n ($1 \leq w, h, n \leq 10^9$).

Формат выходного файла

В выходной файл необходимо вывести ответ на поставленную задачу.

Пример

diploma.in	diploma.out
2 3 10	9

Задача «Карты»**Вхідний файл:** cards.in**Вихідний файл:** cards.out**Обмеження за часом:** 1 с**Обмеження за пам'яттю:** 32 МБ

Колода складається з парної кількості карт. Карти занумеровано послідовними натуральними числами від 1 до $2N$ (для певного натурального N) включно. З колодою роблять таке:

1. Спочатку колоду ділять на дві рівні частини. В першу частину потрапляють карти, що стояли на непарних місцях, в другу — ті карти, що були на парних місцях:

$$(1, 2, \dots, N) \rightarrow (1, 3, 5, \dots, 2N - 1) + (2, 4, \dots, 2N)$$

2. З обох утворених частин викидають карту, що розташована у відповідній частині на місці $1 + \lfloor aN/b \rfloor$ при $0 \leq a < b \leq 1000$. Наприклад, якщо $a = 0, b = 1$, то буде викинуто першу карту:

$$(1, 3, 5, \dots, 2N - 1) \rightarrow (3, 5, 7, \dots, 2N - 1)$$

$$(2, 4, 6, \dots, 2N) \rightarrow (4, 6, 8, \dots, 2N)$$

3. Другу частину кладуть поверх першої:

$$(3, 5, 7, \dots, 2N - 1) + (4, 6, 8, \dots, 2N) \rightarrow (3, 5, 7, \dots, 2N - 1, 4, 6, 8, \dots, 2N)$$

Дії 1–3 повторюють доти, поки в колоді не залишиться лише 2 карти. Наприклад, якщо $a = 1, b = 2$, а початкова кількість карт – 5, маємо:

$$\begin{aligned} (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) &\rightarrow (1, 3, 5, 7, 9) + (2, 4, 6, 8, 10) \rightarrow \\ &(1, 3, 7, 9) + (2, 4, 8, 10) \rightarrow (1, 3, 7, 9, 2, 4, 8, 10) \rightarrow (1, 7, 2, 8) + (3, 9, 4, 10) \rightarrow \\ &(1, 7, 8) + (3, 9, 10) \rightarrow (1, 7, 8, 3, 9, 10) \rightarrow (1, 8, 9) + (7, 3, 10) \rightarrow \\ &(1, 9) + (7, 10) \rightarrow (1, 9, 7, 10) \rightarrow (1, 7) + (9, 10) \rightarrow (1) + (9) \rightarrow (1, 9) \end{aligned}$$

бо $1 + \lfloor 5 \cdot 1/2 \rfloor = 3, 1 + \lfloor 4 \cdot 1/2 \rfloor = 3, 1 + \lfloor 3 \cdot 1/2 \rfloor = 2, 1 + \lfloor 2 \cdot 1/2 \rfloor = 2$.

Завдання: визначте кінцевий стан колоди.

Вхідні дані

Вхідний файл містить 3 невід'ємні цілі числа N, a, b . $2 \leq N \leq 500000$.

Вихідні дані

Вихідний файл має містити два натуральних числа: номери карт, що залишаються. Номер нижньої карти потрібно вказати першим.

Приклад

cards.in	cards.out
5 1 2	1 9

Задача “Solve polynomial”**Input file:** polynomial.in**Output file:** polynomial.out**Time limit:** 2 s**Memory limit:** 64 MB

An integer polynomial of degree n is a function of the form $a_0 + a_1x^1 + a_2x^2 + \dots + a_nx^n$, where each a_i is a constant integer and x is a variable. An integer **root** of an integer polynomial is an integer value of x for which the expression equals zero.

You will be given the coefficients of an integer polynomial, and must return all the integer roots in increasing order. Roots must appear only once in the output (see example 1 for clarification).

Since the degree may be quite large, the coefficients are presented indirectly. Use the following pseudo-code to generate the coefficients $a[0]$ to $a[n]$:

```
lX = length(X)
lY = length(Y)
for i = 0, 1, ..., n:
    p = i mod lX
    q = (i + Y[i mod lY]) mod lX
    a[i] = X[p]
    X[p] = X[q]
    X[q] = a[i]
```

The array indices are all 0-based and $a \bmod b$ is the remainder when a is divided by b . Here arrays X , Y and iteration count n are given to you in the input file.

n will be between 0 and 10000, inclusive.

X will contain between 1 and 50 elements, inclusive.

Y will contain between 1 and 50 elements, inclusive.

Each element of X will be between -10^9 and 10^9 , inclusive.

Each element of Y will be between 0 and 50, inclusive.

At least one element of a will be non-zero.

Input format

First line contains lX — number of elements in X . Second line contains all the elements of X . Third and fourth lines contain the same information for Y . Finally, fifth line contains only number n .

Output format

Output all the integer roots in increasing order, separated by spaces. Each root should appear only once.

Examples

polynomial.in	polynomial.out
3 -4 2 2 1 0 2	-2 1
3 1 2 0 4 2 0 0 0 3	-1
3 1 4 4 1 0 2	
4 -15 -10 2 1 1 0 3	3
21 735134400 1383 4121 18875 10463 13512 19603 28679 13483 9509 1701 13383 24425 7923 7978 21702 30989 20676 18547 28130 10944 14 34 23 6 5 3 5 4 34 37 5 6 21 17 9 10000	