

## Задача А. Без условия

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.25 секунд  
Ограничение по памяти: 1024 мегабайта

Это развлекательный контекст. Все задачи необычные и вряд-ли встретятся на каких-нибудь олимпиадах. Однако это не отменяет того, что для решения каждой задачи надо хоть чуть-чуть подумать, поэтому рекомендуется всё таки порешать эти задачи.

В этой задаче нет условия. Если вы знаете её решение, пожалуйста, не спойлерите другим.

### Формат входных данных

Дано одно целое число  $n$  ( $0 \leq n \leq 10^9$ ).

### Формат выходных данных

Выведите единственное число, ответ на задачу.

### Примеры

стандартный ввод	стандартный вывод
0	1
1	1
2	2
3	3
4	5
5	8
6	13
7	21
8	34

### Замечание

Даже не пытайтесь спрашивать что тут нужно.

## Задача В. Удачные дни

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Боб и Алиса часто участвует в различных соревнованиях по программированию. Как и у многих спортивных программистов, у Алисы и Боба есть удачные и неудачные дни. Они заметили, что их удачные и неудачные дни повторяются с некоторым периодом. Например, у Алисы дни  $[l_a, r_a]$  являются удачными, далее следуют неудачные дни:  $[r_a + 1, l_a + t_a - 1]$ , далее опять удачные: с  $[l_a + t_a, r_a + t_a]$  и так далее. Иначе говоря, удачными для Алисы являются дни, попадающие в отрезок  $[l_a + t_a \cdot k, r_a + t_a \cdot k]$ , для некоторого целого неотрицательного значения  $k$ .

Аналогично устроены удачные дни у Боба, только параметры последовательности у него другие:  $l_b, r_b, t_b$ . То есть для него удачными являются дни попадающие в отрезок  $[l_b + t_b \cdot k, r_b + t_b \cdot k]$ , для некоторого целого неотрицательного значения  $k$ .

Алиса и Боб хотят вместе выступить в командных соревнованиях, и для этого они хотят выяснить, какое максимально количество дней подряд у них обоих могут быть удачные дни.

Нам стало лень делать вам тесты, поэтому в этой задаче вам придётся не только написать решение, но и протестировать его самому и отправить в тестирующую систему архив с тестами. При этом тесты должны быть достаточно хорошие и неправильные решения должны работать на них некорректно, а правильные должны проходить. Ниже предоставлен формат входных и выходных данных.

### Формат входных данных

Первая строка содержит три целых числа  $l_a, r_a, t_a$  ( $0 \leq l_a \leq r_a \leq t_a - 1, 2 \leq t_a \leq 10^9$ ) и задаёт удачные дни для Алисы.

Вторая строка содержит три целых числа  $l_b, r_b, t_b$  ( $0 \leq l_b \leq r_b \leq t_b - 1, 2 \leq t_b \leq 10^9$ ) и задаёт удачные дни для Боба.

### Формат выходных данных

Требуется вывести одно целое число: максимальное количество подряд идущих дней, которые и для Алисы, и для Боба являются удачными.

### Примеры

стандартный ввод	стандартный вывод
0 2 5 1 3 5	2
0 1 3 2 3 6	1

### Замечание

Формат отправки теста такой: вы должны отправить один zip или tar архив, содержащий единственную папку «tests». В этой папке должно лежать не более 200 файлов, описывающих тесты. Все тесты должны быть пронумерованы от 1 до  $k$ , где  $k$  — число тестов, которые вы предлагаете. Для каждого теста должен быть передан сам тест и ответ на него. Например для теста 15 тест должен называться «015.dat» — сам тест, на котором программа работает неверно и «015.ans» — правильный ответ на этот тест. Соответственно у вас должны быть файлы 001.dat, 001.ans, 002.dat, 002.ans и т.д. Более подробно вы можете прочитать тут: [https://ejudge.ru/wiki/index.php/Структура\\_архива\\_с\\_тестами](https://ejudge.ru/wiki/index.php/Структура_архива_с_тестами). При этом наличие файла «README» не обязательно. Файлы с «.» в начале названия будут игнорироваться. Если у вас есть какие-то вопросы на счёт формата или вам ejudge не принимает тесты или чекер падает, пишите мне в telegram. Ваши тесты будут проверяться валидатором, доступным в общем архиве.

## Задача C. Brainfuck

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Мы начинаем курс программирования на языке brainfuck <https://ru.wikipedia.org/wiki/Brainfuck>. Язык состоит всего из 8 команд, описание которых есть на странице википедии. Несмотря на ограниченное число команд, на этом языке можно написать любые по сложности программы. Отличием нашей версии от обычного языка будет только то, что количество ячеек памяти неограниченно. Обратите внимание, что при работе ваша программа не должна делать больше  $5 \cdot 10^7$  действий, а так же длина исходной программы не должна превышать  $5 \cdot 10^5$  символов.

В нашем первом уроке языка brainfuck мы научимся базовым вещам — вводу чисел, выводу чисел, сложению, вычитанию, умножению, делению и взятию по модулю.

От вас требуется написать программу, **генерирующую корректную программу на языке brainfuck**, и выводящую её в стандартный поток вывода. Формат ввода и вывода, с которым должна работать ваша сгенерированная программа, описан ниже. Обратите внимание, что ваша сгенерированная программа не должна содержать символов перевода строки.

Для удобства вот ссылка на тестировщик программ на языке brainfuck <http://algotcode.ru/docs/docs/bff.cpp>.

### Формат входных данных

Сгенерированной вами программе на стандартный поток ввода через знак переноса строки будут передаваться 2 числа:  $a$  и  $b$  ( $0 \leq a < 256$ ,  $1 \leq b < 256$ ). Эти же числа, но в зашифрованном неизвестном вам образом, будут передаваться вашему генератору программы на brainfuck.

### Формат выходных данных

Сгенерированная вами программа на стандартный поток вывода через пробел должна вывести 5 чисел:  $a + b$ ,  $a - b$ ,  $a \cdot b$ ,  $a/b$ ,  $a\%b$ . Операции сложения, вычитания и умножения следует выполнять по модулю 256. Деление и взятие по модулю надо выполнить как с обычными числами, при этом деление надо округлять вниз.

### Пример

стандартный ввод	стандартный вывод
1	3 255 2 0 1
2	

### Замечание

Конечно вы можете зауглеть, как реализовывать все возможные операции на brainfuck, но я рекомендую вам самим попробовать всё это придумать.

Чтобы не умереть во время написания генератора советую все повторяющиеся процедуры заводить в отдельные функции. Так например можно создать функцию *go* для перемещения, функцию *add* для прибавления текущего значения к какой-то ячейке памяти и так со всем. Вообще почти все части пишутся довольно просто, но вот деление может вызвать у вас вопросы. Подумайте, как писать *if else* конструкцию.

## Задача D. Brainfuck

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Мы продолжаем курс программирования на языке brainfuck <https://ru.wikipedia.org/wiki/Brainfuck>. Напоминаем, что отличием нашей версии от обычного языка будет только то, что количество ячеек памяти неограниченно. Обратите внимание, что при работе ваша программа не должна делать больше  $5 \cdot 10^7$  действий, а так же длина исходной программы не должна превышать  $5 \cdot 10^5$  символов.

В нашем втором уроке языка brainfuck мы научимся циклам и массивам, а именно, вам будет предложено отсортировать массив.

От вас требуется написать программу, **генерирующую корректную программу на языке brainfuck**, и выводящую её в стандартный поток вывода. Формат ввода и вывода, с которым должна работать ваша сгенерированная программа, описан ниже. Обратите внимание, что ваша сгенерированная программа не должна содержать символов перевода строки.

Для удобства вот ссылка на тестировщик программ на языке brainfuck <http://algocode.ru/docs/docs/bff.cpp>.

### Формат входных данных

Сгенерированной вами программе на стандартный поток ввода в первой строке будет передаваться число  $n$  ( $2 \leq n \leq 15$ ) — размер массива.

В следующих  $n$  строках вам будет передано по одному числу  $a_i$  ( $0 \leq a_i < 256$ ) — значения элементов массива. Все числа разделены знаком переноса строки.

### Формат выходных данных

Сгенерированная вами программа через пробел должна вывести  $n$  чисел — значения элементов массива в отсортированном по возрастанию порядке.

### Пример

стандартный ввод	стандартный вывод
3	1 2 3
3	
1	
2	

### Замечание

Подумайте, как удобнее всего хранить массив, чтобы по нему можно было удобно итерироваться. Для сортировки я рекомендую сортировку пузырьком, её проще всего писать. У вас наверное возникнут трудности в сравнении двух чисел. Подумайте, как использовать для этого предыдущую задачу.