

### Задача А. X+Y

Имя входного файла: `xy.in`  
Имя выходного файла: `xy.out`  
Ограничение по времени: 1 second  
Ограничение по памяти: 64 megabytes

Вам заданы  $x$  и  $y$ , выведите  $x + y$ !

#### Формат входного файла

Заданы  $x$  и  $y$  ( $1 \leq x, y \leq 1000$ ).

#### Формат выходного файла

Выведите  $x + y$ .

#### Примеры

<code>xy.in</code>	<code>xy.out</code>
20 10	30
10 20	30

### Задача В. Спички

Имя входного файла: `matches.in`  
Имя выходного файла: `matches.out`  
Ограничение по времени: 0.5 секунда  
Ограничение по памяти: 64 мегабайта

Есть  $N$  спичек. Необходимо найти максимально возможную площадь прямоугольника, который можно сложить из этих спичек. При построении прямоугольника нельзя ломать спички и не обязательно использовать все спички. Длины всех спичек одинаковы и равны 1 см.

#### Формат входного файла

В первой строке находится одно целое число —  $N$  ( $4 \leq N \leq 10^9$ ).

#### Формат выходного файла

Выведите максимальную площадь прямоугольника в квадратных сантиметрах.

#### Примеры

<code>matches.in</code>	<code>matches.out</code>
4	1

#### Замечания

Подсказка: используйте `int64` в *Borland Delphi* и *Free Pascal*, `long long` в *GNU C/C++* (для вывода таких чисел используется `%lld`) или `long` в *Java*.

### Задача С. Сортировка слов

Имя входного файла: `stringsort.in`  
Имя выходного файла: `stringsort.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Недавно Сергею Сергеевичу в руки попал список учеников параллели С. Посмотрев на него внимательно, Сергей Сергеевич понял, что имена школьников в списке указаны в произвольном порядке, и это показалось ему крайне некрасивым. Поэтому, он решил убедиться, что все его ученики умеют лексикографически сортировать слова.

Одно слово лексикографически меньше другого, если оно идет раньше по алфавиту или является началом второго слова.

#### Формат входного файла

В первой строке входного файла записано целое число  $N$  — количество слов ( $1 \leq N \leq 100$ ). Далее следуют сами слова, по одному в строке. Все слова состоят только из строчных букв английского алфавита, а их длина не превышает 10.

#### Формат выходного файла

Выведите лексикографически отсортированные слова, по одному в строке.

#### Примеры

<code>stringsort.in</code>	<code>stringsort.out</code>
5	a
a	aa
abacaba	abacaba
bac	bac
aa	zxc
zxc	

### Задача D. Сумма цифр

Имя входного файла: `digitssum.in`  
Имя выходного файла: `digitssum.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Дан набор больших чисел, требуется посчитать сумму цифр каждого числа.

#### Формат входного файла

Первая строка содержит одно число  $N$  ( $1 \leq N \leq 1000$ ) — количество чисел, для которых требуется посчитать сумму цифр. На следующих  $N$  строках даны сами числа  $A_i$  ( $1 \leq i \leq N, 0 \leq A_i \leq 10^{18}$ ).

#### Формат выходного файла

Для каждого числа вывести номер числа и сумму его цифр. Придерживайтесь формата вывода из примера.

### Примеры

digitssum.in	digitssum.out
3	Case #1: 1
10	Case #2: 14
77	Case #3: 22
17239	

### Задача Е. Проверка простоты

Имя входного файла: isprime.in  
Имя выходного файла: isprime.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Целое число  $p \geq 2$  является простым, если у него нет делителей кроме 1 и  $p$ . Необходимо для всех чисел во входном файле проверить простые они или нет.

### Формат входного файла

Во входном файле от 1 до 1000 целых чисел, каждое от 1 до  $10^6$ .

### Формат выходного файла

Для каждого числа во входном файле выведите на отдельной строке YES или NO (в зависимости от того, простое оно или нет).

### Примеры

isprime.in	isprime.out
5	YES
6	NO
7	YES
8	NO

### Замечания

Число 1 не является простым по определению.

### Задача F. Бонусный уровень

Имя входного файла: snake.in  
Имя выходного файла: snake.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Сережа играл в змейку и почти успевал на поезд в ЛКШ. Но он не знал, что за последним уровнем следует дополнительный "бонусный".

Он представляет собой клетчатое поле  $N \times N$ , в каждой клетке которого лежит яблочко. Сережина змея последовательно ползет по диагоналям данного поля, поедая фрукты. Чтобы совершать такие поползновения, ей приходится после каждой диагонали менять направление движения (подробнее см. примеры).

### Формат входного файла

Во входном файле одно целое число  $1 \leq N \leq 100$ .

### Формат выходного файла

Выведите ровно  $N$  строк по  $N$  чисел в каждой: для каждой клетки поля ее номер в последовательности прохождения поля змеей.

### Примеры

snake.in	snake.out
4	1 3 4 10 2 5 9 11 6 8 12 15 7 13 14 16
5	1 3 4 10 11 2 5 9 12 19 6 8 13 18 20 7 14 17 21 24 15 16 22 23 25

### Задача G. Удалить гласные

Имя входного файла: vowels.in  
Имя выходного файла: vowels.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Дана строчка  $s$ , состоящая из строчных букв английского алфавита. Требуется удалить из нее все гласные буквы. Напомним, что в английском алфавите 6 гласных букв: a, e, i, o, u, y.

### Формат входного файла

В единственной строке входного файла содержится строчка  $s$ , состоящая из строчных английских букв. Длина строчки не превосходит  $10^5$  символов.

### Формат выходного файла

Выведите строчку, получившуюся из исходной удалением всех строчных букв.

### Примеры

vowels.in	vowels.out
abcd	bcd
aaaaa	
abcdefghijklmnopqrstuvwxy	bcdfghjklmnpqrstvwxyz

## Задача Н. Ханойские башни

Имя входного файла: `hanoi.in`  
Имя выходного файла: `hanoi.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Даны три колышка. На первом колышке находятся  $N$  дисков сверху вниз по возрастанию размера диска. Два другие пустые. Требуется перенести все диски с первого колышка на второй. Переносить диски разрешается только по одному. Не разрешается класть больший диск на меньший. Для предотвращения заикливания, как это произошло с буддийскими монахами, не используйте циклы в программе.

### Формат входного файла

Вводится одно число  $N$ ,  $1 \leq N \leq 19$ .

### Формат выходного файла

Выведите по два числа в строке – номер колышка, откуда и куда переносится диск. Выведите кратчайшее решение.

### Примеры

<code>hanoi.in</code>	<code>hanoi.out</code>
3	1 2 1 3 2 3 1 2 3 1 3 2 1 2

## Задача I. Инопланетная переписка

Имя входного файла: `numbers.in`  
Имя выходного файла: `numbers.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

На заре  $XXX$  века ученые планеты Глюк отправили сообщение в космос. Сообщение состояло из одного целого числа  $a$ , записанного в системе счисления с основанием  $b$ . Вчера ученым удалось получить сообщение с далекой планеты Баг. Ученые установили, что сообщение — целое число  $c$ , записанное в системе счисления с основанием  $d$ . Чтобы проверить, что сообщение является ответом на послание ученых планеты Глюк, надо проверить, что эти числа совпадают. Ваша задача, как главного программиста планеты Глюк состоит в том, чтобы помочь ученым проверить равенство этих двух чисел.

### Формат входного файла

В первой строке входного файла находятся два целых числа —  $a(0 \leq a \leq 10^9)$  и  $b(2 \leq b \leq 10)$ . Число  $a$  записано в системе счисления с основанием  $b$ . Во второй стро-

ке — числа  $c(0 \leq c \leq 10^9)$  и  $d(2 \leq d \leq 10)$ . Число  $c$  записано в системе счисления с основанием  $d$ .

### Формат выходного файла

В выходной файл выведите "YES"(без кавычек), если ученые планеты Глюк получили ответ на свое сообщение. Иначе выведите "NO"(без кавычек)

### Примеры

<code>numbers.in</code>	<code>numbers.out</code>
1 2 1 10	YES
1 10 2 10	NO
111 2 21 3	YES