

## Задача A. $A+B$

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

Вычислите сумму  $A+B$ .

### Формат входного файла

Первая строка входного файла содержит целое число  $A$ , вторая - целое число  $B$  ( $0 \leq A, B \leq 10^{3000}$ ).

### Формат выходного файла

Первая строка выходного файла должна содержать одно целое число -  $A+B$ .

### Примеры

<code>sum.in</code>	<code>sum.out</code>
6 4	10
467499778956485 563337	467499779519822

## Задача B. $A-B$

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

Вычислите разность  $A-B$ .

### Формат входного файла

Первая строка входного файла содержит целое число  $A$ , вторая - целое число  $B$  ( $0 \leq A, B \leq 10^{3000}$ ).

### Формат выходного файла

Первая строка выходного файла должна содержать целое число -  $A-B$ . Ответ может быть отрицательным!

### Примеры

<code>sub.in</code>	<code>sub.out</code>
6 495	-489
812100766 641888	811458878

## Задача C. $A*B$

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

Вычислите сумму  $A*B$ .

### Формат входного файла

Первая строка входного файла содержит целое число  $A$ , вторая - целое число  $B$  ( $0 \leq A, B \leq 10^{3000}$ ).

### Формат выходного файла

Первая строка выходного файла должна содержать целое число -  $A*B$ .

### Примеры

<code>mul.in</code>	<code>mul.out</code>
6 4	24
506155469 2124996	1075578347003124

## Задача D. $A \div B$

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

Вычислите частное (целую часть) при делении  $A$  на  $B$ .

### Формат входного файла

Первая строка входного файла содержит целое число  $A$ , вторая - целое число  $B$  ( $0 \leq A, B \leq 10^{3000}$ ).

### Формат выходного файла

Первая строка выходного файла должна содержать одно целое число -  $A \div B$ .

### Примеры

<code>div.in</code>	<code>div.out</code>
66350 896	74
12 13	0
66151802969158216 63962	1034235999017

## Задача E. $A \bmod B$

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

Вычислите остаток при целочисленном делении  $A$  на  $B$ .

### Формат входного файла

Первая строка входного файла содержит целое число  $A$ , вторая - целое число  $B$  ( $0 \leq A, B \leq 10^{3000}$ ).

### Формат выходного файла

Первая строка выходного файла должна содержать одно целое число - остаток от деления.

### Примеры

<code>mod.in</code>	<code>mod.out</code>
6 3	0
6 4	2
673 1	0
66350 896	46
506155469 2124996	406421

## Задача F. Четный граф

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

Неориентированный граф называется четно-нечетным, если найдутся две его вершины, между которыми существует пути как из четного, так и из нечетного числа ребер. Напишите программу, которая:

а) определяет, является ли заданный граф четно-нечетным;

б) В случае отрицательного ответа на пункт а) находит максимальное подмножество  $X$  вершин графа такое, что для любых двух вершин  $i$  и  $j$  из  $X$  выполняется следующее условие: все пути между  $i$  и  $j$  состоят из четного числа ребер.

### Формат входного файла

Первая строка входного файла содержит число вершин графа  $N$  ( $1 \leq N \leq 100$ ), а каждая последующая - пару чисел  $(i, j)$ , означающих, что в графе присутствует ребро, соединяющее вершины с номерами  $i$  и  $j$ .

### Формат выходного файла

Первая строка выходного файла должна содержать ответ на пункт А в форме YES/NO. В случае отрицательного ответа на пункт А вторая строка должна содержать количество вершин в множестве  $X$ , а третья - номера вершин из этого множества в порядке возрастания, записанные через пробел. Если вариантов решений несколько, то достаточно вывести любое из них.

### Примеры

graph.in	graph.out
3 1 2	NO 2 2 3
11 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10 11 11 1	YES
4 1 2 2 3 3 4 4 1	NO 2 1 3

## Задача G. Игра в города

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

Всем известны правила игры «в города»: первый игрок называет произвольный город, следующий - город, название которого начинается на ту же букву, на которую заканчивается название предыдущего города, и т.д. Аналогичным образом можно играть не в названия городов, а, например, в названия животных. Задан список допустимых для описанной игры слов, слова в нем могут повторяться.

Напишите программу, определяющую, в каком порядке в процессе игры должны быть названы слова из списка, чтобы каждое слово было использовано ровно столько раз, сколько оно в нем встречается.

### Формат входного файла

В первой строке входного файла записано целое число  $N$  - количество слов в списке ( $1 \leq N \leq 1000$ ), а в последующих  $N$  строках - сами слова. Каждое из них является последовательностью не более чем из 30 строчных английских букв.

### Формат выходного файла

Выведите в выходной файл слова в искомом порядке, либо сообщение «NO», если такого порядка не существует. Каждое слово должно быть выведено в отдельную строку выходного файла.

### Примеры

town.in	town.out
4 b ab bc bb	ab bb b bc