

## Задача А. Встреченные ранее числа

Имя входного файла: `metbefore.in`  
Имя выходного файла: `metbefore.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

### Формат входных данных

Во входной строке записана последовательность из не более, чем 100 000 чисел через пробел. Каждое из чисел не превышает  $10^9$  по абсолютному значению.

### Формат выходных данных

Для каждого числа выведите слово **YES** (в отдельной строке), если это число ранее встречалось в последовательности или **NO**, если не встречалось.

### Примеры

<code>metbefore.in</code>	<code>metbefore.out</code>
1 2 3 2 3 4	NO NO NO YES YES NO

## Задача В. Частотный анализ

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

Вам дан текст. Мы не спрашиваем вас, что хотел сказать автор; в течение отведенного вам времени выпишите все слова из текста в порядке убывания их частоты.

### Формат входных данных

Во входном файле содержится исходный текст. Текст состоит не более чем из 100 000 слов, разделённых пробелами и переводами строк. Все слова состоят из строчных латинских букв. Соседние слова разделены ровно одним пробельным символом. Длина любого слова не превышает 20 символов.

### Формат выходных данных

Выведите все слова, встречающиеся в тексте, по одному на каждую строку. Слова должны быть отсортированы по убыванию их количества в тексте, а при равенстве — по алфавиту.

### Примеры

frequency.in	frequency.out
hi hi what is your name my name is bond james bond my name is damme van damme claudio van damme jean claudio van damme	damme is name van bond claudio hi my james jean what your
oh you touch my tralala mmm my ding ding dong	ding my dong mmm oh touch tralala you

## Задача С. Англо-латинский словарь

Имя входного файла: dictionary.in  
Имя выходного файла: dictionary.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Однажды, разбирая старые книги на чердаке, школьник Вася нашёл англо-латинский словарь. Английский он к тому времени знал в совершенстве, и его мечтой было изучить латынь. Поэтому попавшийся словарь был как раз кстати.

К сожалению, для полноценного изучения языка недостаточно только одного словаря: кроме англо-латинского необходим латинско-английский. За неимением лучшего он решил сделать второй словарь из первого.

Как известно, словарь состоит из переводимых слов, к каждому из которых приводится несколько слов-переводов. Для каждого латинского слова, встречающегося где-либо в словаре, Вася предлагает найти все его переводы (то есть все английские слова, для которых наше латинское встречалось в его списке переводов), и считать их и только их переводами этого латинского слова.

Помогите Васе выполнить работу по созданию латинско-английского словаря из англо-латинского.

### Формат входных данных

Во входном файле содержатся несколько описаний английских слов. Каждое описание содержится в отдельной строке, в которой записано сначала английское слово, затем отделённый пробелами дефис (символ номер 45), затем разделённые запятыми с пробелами переводы этого английского слова на латинский. Переводы отсортированы в лексикографическом порядке. Порядок следования английских слов в словаре также лексикографический.

Все слова состоят только из маленьких латинских букв, длина каждого слова не превосходит 15 символов. Общее количество слов на входе не превышает 100 000.

### Формат выходных данных

Программа должна вывести количество латинских слов в словаре  $k$ . В следующих  $k$  строках программа должна вывести латинско-английский словарь, соответствующий входному словарю, в точности соблюдая формат входных данных. В частности, первым должен идти перевод лексикографически минимального латинского слова, далее — второго в этом порядке и т. д. Внутри перевода английские слова должны быть также отсортированы лексикографически.

### Примеры

dictionary.in	dictionary.out
apple - malum, pomum, popula	7
fruit - baca, bacca, popum	baca - fruit
punishment - malum, multa	bacca - fruit
	malum - apple, punishment
	multa - punishment
	pomum - apple
	popula - apple
	popum - fruit

## Задача D. Расстояние между вершинами

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

Коль Дейкстру́ писать без кучи,  
То тайм-лимит ты получишь...  
А в совсем крутой задаче  
Юзай кучу Фибоначчи!

---

Спектакль преподавателей ЛКШ.июль-2007

Дан неориентированный взвешенный граф. Требуется найти вес минимального пути между двумя вершинами.

### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количества вершин и рёбер графа соответственно ( $1 \leq n \leq 100\,000$ ,  $1 \leq m \leq 200\,000$ ). Вторая строка входного файла содержит натуральные числа  $s$  и  $t$  — номера вершин, длину пути между которыми требуется найти ( $1 \leq s, t \leq n$ ,  $s \neq t$ ).

Следующие  $m$  строк содержат описание рёбер по одному на строке. Ребро номер  $i$  описывается тремя натуральными числами  $b_i$ ,  $e_i$  и  $w_i$  — номерами концов ребра и его вес соответственно ( $1 \leq b_i, e_i \leq n$ ,  $0 \leq w_i \leq 10000$ ).

### Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число — вес минимального пути между вершинами  $s$  и  $t$ , или -1, если такого пути нет.

### Примеры

dist.in	dist.out
4 4 1 3 1 2 1 2 3 2 3 4 5 4 1 4	3

## Задача Е. Коммерческий калькулятор

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

Фирма OISAC выпустила новую версию калькулятора. Этот калькулятор берет с пользователя деньги за совершаемые арифметические операции. Стоимость каждой операции в долларах равна 5% от числа, которое является результатом операции.

На этом калькуляторе требуется вычислить сумму  $N$  натуральных чисел (числа известны). Нетрудно заметить, что от того, в каком порядке мы будем складывать эти числа, иногда зависит, в какую сумму денег нам обойдется вычисление суммы чисел (тем самым, оказывается нарушен классический принцип "от перестановки мест слагаемых сумма не меняется":-) ).

Например, пусть нам нужно сложить числа 10, 11, 12 и 13. Тогда если мы сначала сложим 10 и 11 (это обойдется нам в \$1.05), потом результат - с 12 (\$1.65), и затем - с 13 (\$2.3), то всего мы заплатим \$ 5, если же сначала отдельно сложить 10 и 11 (\$1.05), потом - 12 и 13 (\$1.25) и, наконец, сложить между собой два полученных числа (\$2.3), то в итоге мы заплатим лишь \$4.6.

Напишите программу, которая будет определять, за какую минимальную сумму денег можно найти сумму данных  $N$  чисел.

### Формат входных данных

Во входном файле записано число  $N$  ( $2 \leq N \leq 1000$ ). Далее идет  $N$  натуральных чисел, которые нужно сложить, каждое из них не превышает 10000.

### Формат выходных данных

В выходной файл выведите, сколько денег нам потребуется на нахождение суммы этих  $N$  чисел. Результат должен быть выведен с двумя знаками после десятичной точки.

### Примеры

<code>calc.in</code>	<code>calc.out</code>
4 10 11 12 13	4.60
2 1 1	0.10

## Задача F. Равные подстроки

Имя входного файла: `eqsubstr.in`  
Имя выходного файла: `eqsubstr.out`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 256 мегабайт

Дана строка  $S = s_1s_2 \dots s_n$  и множество запросов вида  $(l_1, r_1, l_2, r_2)$ . Для каждого запроса требуется ответить, равны ли подстроки  $s_{l_1} \dots s_{r_1}$  и  $s_{l_2} \dots s_{r_2}$ .

### Формат входных данных

В первой строке записана строка  $S$ , состоящая из строчных латинских букв. Эта строка непустая и имеет длину не более 100 000 символов. Во второй строке записано целое число  $q$  ( $1 \leq q \leq 100\,000$ ) — количество запросов. В каждой из следующих  $q$  строк записаны числа  $l_1, r_1, l_2, r_2$  ( $1 \leq l_1 \leq r_1 \leq |S|$ ;  $1 \leq l_2 \leq r_2 \leq |S|$ ).

### Формат выходных данных

Для каждого запроса выведите «+», если соответствующие подстроки равны, и «-», в противном случае.

### Примеры

eqsubstr.in	eqsubstr.out
abacaba 4 1 1 7 7 1 3 5 7 3 4 4 5 1 7 1 7	++-+
qa 3 1 1 1 1 2 2 2 2 1 1 2 2	++-

## Задача G. Поиск общей подстроки

Имя входного файла: `commonsubstr.in`  
Имя выходного файла: `commonsubstr.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дана строка  $A = a_1a_2 \dots a_n$  и строка  $B = b_1b_2 \dots b_m$ . Так же дано число  $L$ . Нужно узнать, есть ли у строк  $A$  и  $B$  общая подстрока длиной  $L$ .

### Формат входных данных

В первых двух строках записаны строки  $A$  и  $B$ , состоящие из строчных латинских букв. Эти строки непустые и имеют длину не более 100 000 символов. В третьей строке записано целое число  $L$  ( $0 \leq L \leq 100\,000$ ) — длина общей подстроки.

### Формат выходных данных

В выходной файл выведите YES, если существует общая подстрока такой длины. В противном случае выведите NO.

### Примеры

<code>commonsubstr.in</code>	<code>commonsubstr.out</code>
saaa baaa 3	YES
raabc taaac 3	NO