

## Задача А. Задача для второклассника

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

Вам даны два числа. Необходимо найти их произведение.

### Формат входных данных

Входные данные состоят из двух строк, на каждой из которых находится целое одно **целое** число, длина которого не превосходит двухсот пятидесяти тысяч символов.

### Формат выходных данных

Выведите произведение данных чисел.

### Примеры

multiply.in	multiply.out
2	4
2	

## Задача В. HEX-Hell и сломанная строка

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 256 мегабайт

Серёжа потерял место, где в редакторе шестнадцатеричных кодов HEX-Hell находилась его строка. И, так как он большой молодец, саму строку он сломал.

Напоминаем, что один байт — две шестнадцатеричных цифры из диапазона `[0-9A-F]`. Последовательность кодов в редакторе в данный момент имеет длину не более 125 000 байт (т.е. в ней не более 250 000 символов, и она имеет четную длину).

А также, у Вас есть серёжина битовая строка из нулей и единиц, «поломанная» в некоторых местах (некоторые биты будут заменены на знаки ?).

Пожалуйста, для каждого возможного начала этой строки в редакторе выведите количество совпадающих нулей и единиц. При этом можно считать, что знаки вопроса совпадают с чем угодно! По этой информации горе-Серёжа разберется со своими строками сам. Только найдите количество совпадений!

### Формат входных данных

Во входном файле две непустых строки. Первая состоит из символов от 0 до 9 и от A до F. Количество таких символов чётно и не более 250 000.

Вторая строка состоит из нулей, единиц и знаков вопроса. Её длина кратна восьми (она тоже задаёт последовательность байт) и не превосходит четырех длин первой строки (она помещается в редактор).

### Формат выходных данных

Если длина (в символах) первой строки  $n$ , а второй  $k$ , то Вам необходимо вывести в первой строке выходного файла  $n/2 - k/8 + 1$  целых чисел — количество совпадающих бит при прикладывании второй строки к некоторому месту первой строки.

Прикладывания упорядочены естественным образом — слева направо.

### Примеры

<code>stdin</code>	<code>stdout</code>
B2D6 1011?010	8 5

### Замечание

B2D6 в двоичном виде это 10110010 11010110

При прикладывании шаблона 1011?010 к первому байту 10110010 получаем восемь совпадений, а ко второму 11010110 — пять.

## Задача С. Дуэль

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

Двое дуэлянтов решили выбрать в качестве места проведения поединка тёмную аллею. Вдоль этой аллеи растёт  $n$  деревьев и кустов. Расстояние между соседними объектами равно одному метру. Дуэль решили проводить по следующим правилам. Некоторое дерево выбирается в качестве стартовой точки. Затем два дерева, находящихся на одинаковом расстоянии от исходного, отмечаются как места для стрельбы. Дуэлянты начинают движение от стартовой точки в противоположных направлениях. Когда соперники достигают отмеченных деревьев, они разворачиваются и начинают стрелять друг в друга.

Дана схема расположения деревьев вдоль аллеи. Требуется определить количество способов выбрать стартовую точку и места для стрельбы согласно правилам дуэли.

### Формат входных данных

Во входном файле содержится одна строка, состоящая из символов '0' и '1' — схема аллеи. Деревья обозначаются символом '1', кусты — символом '0'. Длина строки не превосходит 100 000 символов.

### Формат выходных данных

Выведите количество способов выбрать стартовую точку и места для стрельбы согласно правилам дуэли.

### Примеры

<code>duel.in</code>	<code>duel.out</code>
101010101	4
101001	0

### Замечание

В первом примере возможны следующие конфигурации дуэли (стартовое дерево и деревья для стрельбы выделены жирным шрифтом): **101010101**, **101010101**, **101010101** и **101010101**.

## Задача D. Пути в дереве

Имя входного файла: `tree-paths.in`  
Имя выходного файла: `tree-paths.out`  
Ограничение по времени: 15 секунд  
Ограничение по памяти: 512 мегабайт

Дано дерево из  $n$  вершин. Найдите для каждого  $d$  от 1 до  $n - 1$  число путей длины  $d$ .

### Формат входных данных

Первая строка содержит  $n$  — число вершин дерева ( $1 \leq n \leq 50000$ ).

Следующие  $n - 1$  строк содержат ребра дерева. Каждое ребро описывается парой чисел  $v_i, u_i$  — концы ребра ( $1 \leq v_i, u_i \leq n$ ).

### Формат выходных данных

Выведите  $n - 1$  число:  $i$ -е число — число путей длины  $i$ .

### Примеры

<code>tree-paths.in</code>	<code>tree-paths.out</code>
3	2
1 2	1
2 3	
9	8
3 2	10
4 2	10
1 2	6
5 1	2
1 6	0
7 6	0
6 8	0
8 9	

## Задача Е. Системы счисления

Имя входного файла: `radix.in`  
Имя выходного файла: `radix.out`  
Ограничение по времени: 15 секунд  
Ограничение по памяти: 256 мегабайт

Переведите число из системы счисления  $a$  в систему счисления  $b$ .

### Формат входных данных

В первой строке задана начальная система счисления  $a$  ( $2 \leq a \leq 10$ ). Во второй строке задано само число, его длина не превышает 100 000 символов. В третьей строке записана система счисления  $b$ , в которую нужно перевести число ( $2 \leq b \leq 10$ ).

### Формат выходных данных

В единственной строке выведите заданное число в системе счисления  $b$ .

### Примеры

<code>radix.in</code>	<code>radix.out</code>
3	27
1000	
10	