

## Задача А. Снова про строки

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

*Неоднородностью* строки  $s$  назовем разность между числом вхождений в нее самого часто встречающегося символа и самого редко встречающегося символа. Например, для строки `aaabbc` неоднородность равна двум, так как наиболее часто встречающийся символ `a` входит в нее три раза, а наиболее редко встречающийся символ `c` — один раз.

Напишите программу, вычисляющую неоднородность заданной строки.

### Формат входного файла

Первая строка входного файла содержит непустую строку  $s$  длиной не более 100 символов. Строка  $s$  состоит только из строчных букв латинского алфавита.

### Формат выходного файла

В выходной файл выведите ответ на задачу.

### Примеры

| <code>again.in</code> | <code>again.out</code> |
|-----------------------|------------------------|
| <code>aaabbc</code>   | 2                      |
| <code>abba</code>     | 0                      |

## Задача В. Размножение апельсинов

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

Одному студенту одного университета по имени Саша преподаватель задал изучить и рассказать своей группе о одном красивом математическом парадоксе. Суть парадокса<sup>1</sup> заключается в том, что любой шар в пространстве можно разрезать на 5 частей, и с помощью поворотов и параллельных переносов составить из них два абсолютно таких же шара. Более того, шар можно разрезать на 5 частей и собрать из них два шара любых радиусов.

Ребята из группы, в принципе, поняли, как собрать два одинаковых шара, однако, их воображение и математическая подкованность не позволила понять им, что можно собирать шары разных радиусов, поэтому подобная операция ассоциируется у них с каким-то неведомым волшебством или обманом зрения. Докладчик решил, что наилучшим убеждением будет просто привести пример. Для этого он достал из сумки апельсин, представляющий собой шар радиуса  $R$  и спросил каждого из студентов, апельсин какого размера он хочет получить. Получив набор чисел  $\{R_i\}$  он взял нож и начал резать...

За одну операцию докладчик брал со стола один апельсин радиуса  $r$ , и после некоторых операций клал на стол два апельсина радиусов  $r'$  и  $r''$ . Однако разрезать шар на шары размеров отличных от исходного тяжело, поэтому Саша хочет минимизировать количество операций, в которых  $r \neq r'$  или  $r \neq r''$ . Также он понял, что если он вдруг нарежет больше апельсинов чем нужно, то скорее всего ребята подумают, что он незаметно достает их из рукава, и тоже будут считать его обманщиком.

Помогите Саше выбраться из столь затруднительной для него ситуации.

### Формат входного файла

В первой строке входного файла записано два целых числа  $n$  ( $2 \leq n \leq 10^3$ ) — количество студентов в группе (не включая Сашу) и  $R$  ( $1 \leq R \leq 10^9$ ). В следующей строке записано  $n$  целых чисел  $R_1, \dots, R_n$  ( $1 \leq R_i \leq 10^9$ ).

### Формат выходного файла

В первой строке запишите одно число — минимальное количество раз, которое Саше придется разрезать апельсин на два, радиус хотя бы одного из которых отличен от радиуса апельсина. В следующих  $n - 1$  строке (а именно столько будет операций) запишите описание операций в формате:  $r \ r' \ r''$ .

### Примеры

| <code>btp.in</code> | <code>btp.out</code> |
|---------------------|----------------------|
| 2 1<br>1 1          | 0<br>1 1 1           |
| 2 1<br>1 2          | 1<br>1 2 1           |

---

<sup>1</sup>Подробнее о парадоксе Банаха-Тарского можно прочитать в статье Яценко И.В. и в Википедии

## Задача С. Ферма

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

На ферме живут  $x$  коров и  $y$  овец. Недавно фермер Петя увлекся математикой и теперь хочет приобрести  $z$  коз, чтобы выполнялось соотношение  $x^n + y^n = z^n$ . После некоторого размышления он понял, что такое  $z$  найти непросто, поэтому он решил упростить себе задачу — теперь он хочет купить  $z$  коз так, чтобы  $z^n$  было как можно ближе к  $x^n + y^n$  (то есть число  $|x^n + y^n - z^n|$  было как можно меньше).

Помогите фермеру Пете — напишите программу, которая по числам  $x$ ,  $y$  и  $n$  найдет нужное ему число  $z$ .

### Формат входного файла

Входной файл содержит три целых числа:  $x$ ,  $y$  и  $n$  ( $1 \leq x, y \leq 100$ ,  $3 \leq n \leq 10$ ),  $x^n + y^n \leq 10^9$ .

### Формат выходного файла

В первой строке выходного файла выведите минимальное значение  $k$  величины  $|x^n + y^n - z^n|$ . Во второй строке выходного файла выведите  $z$ , на котором это значение достигается. Кроме этого, если  $k = 0$ , то в третьей строке выходного файла выведите без кавычек фразу «*Cuius rei demonstrationem mirabilem sane detexi. Hanc marginis exiguitas non caperet.*».

### Примеры

| <code>fermat.in</code> | <code>fermat.out</code> |
|------------------------|-------------------------|
| 1 1 3                  | 1<br>1                  |
| 2 3 4                  | 16<br>3                 |

## Задача D. Футбольная статистика

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

Вам надо написать программу которая будет вести статистику матча по игрокам для чемпионата мира по футболу 2010 года.

Как известно в футбол играют две команды по  $n$  игроков в каждой. Игроки пасуют друг другу мяч и иногда бьют по воротам и иногда забивают голы. Игра проходит довольно быстро, поэтому единственная членораздельная информация произносится спортивными комментаторами. Именно на нее мы и будем полагаться.

Спортивные комментаторы называют имена игроков, которые владеют мячом. А когда забивается гол, кричат «ГОЛ!!!». Вам задана эта последовательность имен игроков в которой они владели мячом, в которой иногда встречается слово «GOAL!», говорящее что был забит гол. Вам нужно определить сколько голов забил каждый игрок, сколько результативных передач он дал и сколько совершил ошибок.

- Считается, что игрок забил гол, если он владел мячом, после чего был забит гол.
- Считается, что игрок  $A$  дал результативную передачу игроку  $B$ , если  $B$  тут же забил гол (без других промежуточных пасов) и игроки  $A$  и  $B$  играют в одной команде. В случае, если игрок забивает мяч непосредственно в начале игры (никто не владел им до него) или непосредственно после предыдущего гола, очко за результативную передачу не начисляется.
- Считается, что игрок  $A$  совершил ошибку, если он дал пас игроку  $B$ , после чего тот тут же забил гол (без других промежуточных пасов) и игроки  $A$  и  $B$  играют в разных командах.

### Формат входного файла

В первой строке входного файла записано число  $n$  ( $1 \leq n \leq 100$ ) — количество игроков в одной команде. В следующих  $2n$  строках перечислены имена игроков. Первые  $n$  играют за первую команду, вторые  $n$  за вторую. Имена представляют собой непустые строчки из маленьких латинских букв, не длиннее 10 символов.

Далее следует не более 100000 имен игроков владевших мячом и слов «GOAL!» разделенных пробелами или переводами строки.

Гарантируется, что имена игроков не совпадают.

### Формат выходного файла

В выходной запишите сначала статистику по выигравшей команде, затем пустую строку, затем статистику по проигравшей (в случае ничьей порядок команд остается как во входном файле).

Статистика по команде представляет собой  $n$  строк, в каждой из которых, через пробел, записано имя игрока, количество забитых им голов, отданных результативных передач и совершенных ошибок. Порядок игроков должен совпадать с порядком их перечисления во входном файле.

## Примеры

| football.in                      | football.out   |
|----------------------------------|----------------|
| 4                                | pasha: 0 1 0   |
| pasha                            | serega: 0 0 0  |
| serega                           | dima: 1 0 0    |
| dima                             | sasha: 1 0 0   |
| sasha                            |                |
| ronaldo                          | ronaldo: 0 0 1 |
| figo                             | figo: 0 1 0    |
| rooney                           | rooney: 1 0 0  |
| zidane                           | zidane: 0 0 0  |
| zidane figo pasha sasha GOAL!    |                |
| serega figo rooney GOAL! ronaldo |                |
| dima GOAL! zidane                |                |

## Задача Е. Палиндромность — 3

Имя входного файла: `palin3.in`  
Имя выходного файла: `palin3.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Напомним, что *палиндромом* называется строка, которая читается одинаково как слева направо, так и справа налево. Например, палиндромами являются строки «abba» и «madam».

Для произвольной строки  $s$  введем операцию *деления пополам*, обозначаемую  $half(s)$ . Значение  $half(s)$  определяется следующими правилами:

- Если  $s$  не является палиндромом, то значение  $half(s)$  не определено;
- Если  $s$  имеет длину 1, то значение  $half(s)$  также не определено;
- Если  $s$  является палиндромом четной длины  $2m$ , то  $half(s)$  — это строка, состоящая из первых  $m$  символов строки  $s$ ;
- Если  $s$  является палиндромом нечетной длины  $2m + 1$ , большей 1, то  $half(s)$  — это строка, состоящая из первых  $m + 1$  символов строки  $s$ .

Например, значения  $half(\text{informatics})$  и  $half(i)$  не определены,  $half(\text{abba}) = \text{ab}$ ,  $half(\text{madam}) = \text{mad}$ .

*Палиндромностью* строки  $s$  будем называть максимальное число раз, которое можно применить к строке  $s$  операцию деления пополам, чтобы результат был определен.

Например, палиндромность строк «informatics» и «i» равна 0, так как к ним нельзя применить операцию деления пополам даже один раз. Палиндромность строк «abba» и «madam» равна 1, а палиндромность строки «totottotot» равна 3, поскольку операция деления пополам применима к ней три раза: «totottotot» → «totot» → «tot» → «to».

Задана некоторая строка  $s$  и число  $k$ . Необходимо вычислить палиндромность заданной строки.

### Формат входного файла

Первая строка входного файла содержит непустую строку  $s$ , состоящую из строчных букв латинского алфавита. Ее длина не превосходит  $10^5$  символов.

### Формат выходного файла

В выходной файл выведите ответ палиндромность строки, заданной во входном файле.

### Примеры

| <code>palin3.in</code>   | <code>palin3.out</code> |
|--------------------------|-------------------------|
| <code>informatics</code> | 0                       |
| <code>madam</code>       | 1                       |
| <code>totottotot</code>  | 3                       |



## Задача G. Сумма? Это просто!

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

Сумма всех простых чисел от 2 до 1000000 равна 37550402023 и действительно является простым числом.

---

из блога <http://blog.sjinks.org.ua>

Вася занимается в математическом кружке и недавно узнал о *простых* числах. Напомним, что целое число  $x$  называется простым, если  $x \geq 2$  и  $x$  не делится ни на одно число от 2 до  $x - 1$ .

Вася — очень любознательный мальчик, поэтому он решил найти как можно больше информации о простых числах. Для этого он решил воспользоваться интернетом и ввел в строку поиска одной популярной поисковой системы словосочетание **простые числа**. На одной из найденных страниц он прочитал, что сумма всех простых чисел, не превосходящих миллион, является простым числом, и заинтересовался, насколько часто случаются подобные ситуации.

Помогите Васе — напишите программу, которая по числу  $n$ , вычислит сумму всех простых чисел, не превосходящих  $n$  и определит, является ли эта сумма простым числом.

### Формат входного файла

Первая строка входного файла содержит число  $n$  ( $2 \leq n \leq 10^4$ ).

### Формат выходного файла

В первой строке выходного файла выведите сумму  $S$  всех простых чисел, не превосходящих  $n$ . Во второй строке выходного файла выведите слово YES, если  $S$  является простым числом, и слово NO иначе.

### Примеры

| <code>prime.in</code> | <code>prime.out</code> |
|-----------------------|------------------------|
| 3                     | 5<br>YES               |
| 6                     | 10<br>NO               |

## Задача Н. Тройки чисел

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

Найдите число троек  $0 \leq i, j, k \leq n$ , для которых выполняются условия  $i = j \pmod 2$  и  $j = k \pmod 3$ .

### Формат входного файла

Первая строка входного файла содержит целое число  $n$  ( $1 \leq n \leq 100000$ ).

### Формат выходного файла

В выходной файл выведите ответ на задачу.

### Примеры

| <code>triples.in</code> | <code>triples.out</code> |
|-------------------------|--------------------------|
| 4                       | 5                        |

### Примечание

Для  $n = 4$  соответствующими тройками являются  $(0\ 0\ 0)$ ,  $(0\ 0\ 3)$ ,  $(0\ 2\ 2)$ ,  $(1\ 1\ 1)$  и  $(1\ 1\ 4)$ .