1. **Прямоугольник**

Ограничение по времени: **2 секунды**

Ограничение по памяти: **65 мегабайт**

Николай любит, когда всё по полочкам и когда всё компактно. Для него надо написать программу, которая покажет, как ровным прямоугольником записывать набор знаков (букв). При этом необходимо, чтобы прямоугольник имел соотношение сторон как можно более близкое к квадрату, то есть 1:1.

Входные данные

Дан набор букв латинского алфавита. Количество букв не более 10000.

Выходные данные

Выведите требуемый прямоугольник символов.

Примеры

|  |  |
| --- | --- |
| Ввод | Вывод |
| abcd | ab  cd |
| abcdefg | abcdefg |
| abc | a  b  c |
| abcdtfghhaaa | abcd  tfgh  haaa |

1. **Сумма 3‑х чисел**

Ограничение по времени: **2 секунды**

Ограничение по памяти: **65 мегабайт**

Николай знает, что в математике от перестановки мест слагаемых сумма не меняется. Он решил проверить, так ли это в информатике.

Входные данные

Дано три действительных числа, по одному на каждой строчке. Гарантируется, что в записи каждого из них присутствует не более 11 цифр.

Выходные данные

Выведите результат сложения 3‑х чисел.

Примеры

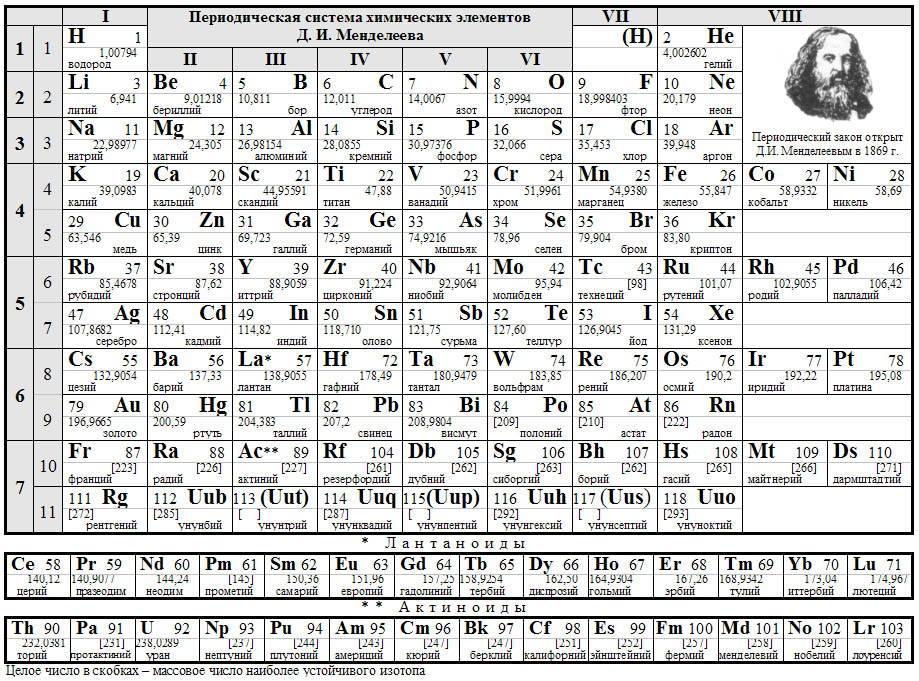
|  |  |
| --- | --- |
| Ввод | Вывод |
| 123456789  -0.000002  -123456789 | -0.000002 |

1. **Химия**

Ограничение по времени: **2 секунды**

Ограничение по памяти: **65 мегабайт**

Николай начал увлекаться химией. Ему сейчас очень нужна программа, которая бы по символическому обозначению химического элемента выдавала бы его номер.



Входные данные

Символическое обозначение химического элемента.

Выходные данные

Порядковый номер элемента в химической таблице Менделеева.

Пример

|  |  |
| --- | --- |
| Ввод | Вывод |
| Ac | 89 |
| Uuo | 118 |
| H | 1 |
| Uut | 113 |

1. **Циклы**

Ограничение по времени: **2 секунды**

Ограничение по памяти: **65 мегабайт**

Николай не скучает в свободное время, он изучает программирование. Когда количество циклов и длина циклов известны заранее, то код у Николая получается вполне короткий.

Входные данные

Одно целое положительное чётное число N (2<N<60).

Выходные данные

Выведите все возможные суммы положительных чётных чисел, которые в сумме дают число N. Порядок вывода значения не имеет.

Пример

|  |  |
| --- | --- |
| Ввод | Вывод |
| 6 | 4+2=6  2+2+2=6 |
| 8 | 2+2+2+2=8  6+2=8  4+4=8  4+2+2=8 |

1. **Деление**

Ограничение по времени: **2 секунды**

Ограничение по памяти: **65 мегабайт**

Николай знает, что из двух последовательных целых чисел одно число всегда четное, а одно из трёх последовательных чисел всегда кратно трём.

Входные данные

В первой строке дано одно число N (0<N<40) – количество чисел во второй строке. Во второй строке через пробел находятся N целых положительных чисел ai (0<ai<40). В третьей строке находится одно положительное целое число Z (0<Z<1018).

Выходные данные

Выведите минимальное число большее Z, которое будет делиться без остатка на все числа указанные во второй строке.

Примеры

|  |  |
| --- | --- |
| Ввод | Вывод |
| 2  2 3  6 | 12 |

1. **Теги**

Ограничение по времени: **2 секунды**

Ограничение по памяти: **65 мегабайт**

Николай знает язык разметки, в котором используются теги. При этом важно, что тег является элементом документа, а текст между начальным и конечным тегами является содержанием документа.

Любой начальный тег имеет вид: <a>, а конечный тег имеет вид: </a>, где a – название тега (любая последовательность букв латинского алфавита).

Есть небольшой нюанс, дело в том, что в языке, о котором идёт речь, элементы должны быть строго вложены друг в друга, то есть «перехлёст» не допускается.

Пример правильного расположения тегов: <b> <a> <c> </c> </a> </b>.

Пример неправильного расположения тегов: <a> <b> <c> </a> </c> </b>.

У Николая имеется программа, которая позволяет редактировать текст, но в ней (во внутреннем представлении документа) перехлёст тегов не только допускается, но и является нормой.

Пример: <b>Тёплые<i> горячие</b> булочки</i>

Результат: **Тёплые *горячие*** *булочки*

Обратите внимание на пробелы, их именно столько, сколько в примере! В тегах пробелов нет (гарантируется).

Входные данные

Дан документ, созданный при помощи программы, в которой «перехлёст» тегов является нормой. Документ состоит из элементов и содержания. Содержание – состоит только из букв латинского алфавита и пробелов, имя элементов только из букв латинского алфавита. Переносов строк нет! В документе не более 70 знаков и не менее 1 знака содержания. Так же гарантируется, что количество начальных и конечных тегов равно по каждому имени.

Выходные данные

Строка, соответствующая языку разметки, который знает Николай. При этом количество знаков не должно превышать 5000.

Примеры

|  |  |
| --- | --- |
| Ввод | Вывод |
| <b>P<i>G</b>U</i> | <b>P</b><i><b>G</b>U</i> |
| <b>P<i>G</b>U</i> | <b>P<i>G</i></b><i>U</i> |
| <b>P<i>G</b>U</i> | <b><b>P<i>G</i></b></b><i>U</i> |
| <PgU><pgu> </pgu> Olymp</PgU> | <PgU><pgu> </pgu> Olymp</PgU> |
| <alfa></alfa><a>t</a> | <a>t</a> |
| e | e |
| t | t |
| <a>T<a>t<\a><\a> | <a>Tt<\a> |