

Задача А. Absolute Prime

Имя входного файла: `absolute.in`
Имя выходного файла: `absolute.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Назовём простое число p *абсолютно простым*, если p остаётся простым после произвольной перестановки цифр в его десятичной записи.

По заданному числу определите, является ли оно абсолютно простым.

Формат входного файла

Первая и единственная строка входного файла содержит целое число p ($1 \leq p \leq 2 \cdot 10^{18}$).

Формат выходного файла

Выведите “YES”, если заданное число абсолютно просто и “NO” в противном случае.

Примеры

<code>absolute.in</code>	<code>absolute.out</code>
2	YES
51	NO

Задача В. Figure

Имя входного файла: `figure.in`
Имя выходного файла: `figure.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Два треугольника заданы координатами своих вершин. Известно, что треугольники пересекаются, что их основания лежат на одной линии, сами они лежат по одну сторону этой линии и являются равнобедренными:

$$y_{11} = y_{13} = y_{21} = y_{23}, y_{12} > y_{11}, y_{22} > y_{21}, x_{12} - x_{11} = x_{13} - x_{12}, x_{22} - x_{21} = x_{23} - x_{22}.$$

Требуется найти периметр многоугольника, получающегося в результате объединения данных треугольников.

Формат входного файла

Во входном файле заданы шесть строк по два целых числа в каждом. В первых трёх строках заданы координаты вершин первого треугольника $x_{11} y_{11}, x_{12} y_{12}, x_{13} y_{13}$, в следующих трёх строках — координаты вершин второго треугольника $x_{21} y_{21}, x_{22} y_{22}, x_{23} y_{23}$.

При этом

$$y_{11} = y_{13} = y_{21} = y_{23}, y_{12} > y_{11}, y_{22} > y_{21}, x_{12} - x_{11} = x_{13} - x_{12}, x_{22} - x_{21} = x_{23} - x_{22}.$$

Все x_{ij} и y_{ij} не превосходят 1000 по абсолютной величине.

Формат выходного файла

Требуется вывести периметр многоугольника, получающегося в результате объединения данных треугольников с точностью 10^{-3} .

Пример

<code>figure.in</code>	<code>figure.out</code>
0 0	11.479
2 1	
4 0	
1 0	
3 2	
5 0	

Задача C. Chess

Имя входного файла: `chess.in`
Имя выходного файла: `chess.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

На пустой шахматной доске (8 на 8) стоит одинокая шахматная фигура. Необходимо определить число клеток, которых фигура может достичь за один ход.

Формат входного файла

В единственной строке входного файла находится название шахматной фигуры по-английски — одно из следующих слов: «king» для короля, «rook» для ладьи, «bishop» для слона, «knight» для коня, «queen» для ферзя. Далее через пробел следует положение фигуры в обычной шахматной нотации (строчная латинская буква и цифра).

Формат выходного файла

В выходной файл необходимо вывести число возможных ходов фигуры.

Пример

<code>chess.in</code>	<code>chess.out</code>
king d4	8

Задача D. Генератор строки

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

Генератором строки назовём наименьший префикс, который нужно повторить (возможно, нецелое число раз), чтобы получить эту строку. Например, генератор строки “`ababab`” — это строка “`ab`”, генератор строки “`abcabca`” — это строка “`abc`”, генератором строки “`abcd`” является она сама.

По данной строке найдите её генератор.

Формат входного файла

В первой строке входного файла задана непустая строка, состоящая из строчных букв латинского алфавита. Длина строки не превышает одного миллиона символов.

Формат выходного файла

В первой строке выходного файла выведите генератор данной строки.

Примеры

<code>genstr.in</code>	<code>genstr.out</code>
<code>ababab</code>	<code>ab</code>
<code>abcabca</code>	<code>abc</code>
<code>abcd</code>	<code>abcd</code>

Задача E. Minimal distance

Имя входного файла: `distance.in`
Имя выходного файла: `distance.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Вам заданы прямоугольник, стороны которого параллельны осям координат, и эллипс. Ваша задача — найти минимальное расстояние между ними. Если они пересекаются или один из них находится внутри другого, считается, что расстояние равно нулю.

Формат входного файла

Первая строка входного файла описывает эллипс и содержит 4 целых числа — x , y , a и b , где:

- (x, y) - координаты центра эллипса
- a — длина полуоси эллипса, параллельной оси абсцисс.
- b — длина полуоси эллипса, параллельной оси ординат.

Вторая строка входного файла также содержит 4 целых числа — описание прямоугольника: первые два числа — координаты x и y левой нижней вершины прямоугольника, следующие два — координаты x и y правой верхней вершины прямоугольника.

Гарантируется, что все числа во входном файле не превосходят по абсолютной величине 1000.

Формат выходного файла

В выходной файл выведите требуемое расстояние с точностью до 10^{-6} .

Примеры

<code>distance.in</code>	<code>distance.out</code>
10 0 5 5 0 0 2 2	3.000000

Задача F. Bad Solution

Имя входного файла: badsol.out
Имя выходного файла: badsol.in
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 mebibytes

По тексту неверной программы и приведённым примерам разберитесь, решением какой задачи является приведённая программа и исправьте ошибку.

```
#include <cassert>
#include <cmath>
#include <cstdio>

const int MinD = 2, MaxD = 10, MinK = 1,
        MaxK = 20, MaxL = 1 << MaxK, MaxV = MaxL;

int label [MaxV];
int d, k, n;

void recur (int v)
{
    int start, out;
    start = (v * d) % n;
    while (label[v])
    {
        label[v]--;
        out = label[v];
        recur (start + label[v]);
        putchar (out + '0');
    }
}

int main (void)
{
    int i;

    scanf ("%d %d", &d, &k);
    assert (MinD <= d && d <= MaxD);
    assert (MinK <= k && k <= MaxK);
    assert (pow (d, k) + k - 1 <= (double) MaxL);

    n = (int) pow (d, k - 1);
    for (i = 0; i < n; i++)
        label[i] = d;
    recur (0);
    for (i = 1; i < k; i++)
        putchar ('0');
    putchar ('\n');

    return 0;
}
```

Пример

	badsol.out	badsol.in
2 3		0001011100