

## Системы счисления.

### A. Перевод двоичной дроби в десятичную

Переведите число из двоичной системы счисления в десятичную.

Дано число, представленное в виде двоичной дроби: запись длиной не более 30 символов, содержащая цифры 0 и 1 и, возможно, одну точку.

Необходимо вывести данное число в виде десятичной дроби (тип переменной `double` с точностью не менее 12 знаков).

Input	Output
0.11	0.75
0.111	0.875

### B. Перевод десятичной дроби в двоичную

Переведите десятичное число в двоичную систему.

Дано действительное неотрицательное число, не превосходящее 100, записанное в десятичном виде. Целые числа при этом могут не содержать точку.

Необходимо представить число в виде двоичной дроби с фиксированной точкой и вывести это представление. Ответ должен отличаться от правильного не более, чем на  $2^{-32}$ , то есть необходимо вывести не менее 32 двоичных цифр после точки.

Input	Output
0.5	0.1
0.1	0.00011001100110011001100110011001100110011001100110

### C. Перевод двоичной периодической дроби в десятичную

Дана запись целого двоичного числа или двоичной периодической дроби, которая включает в себя:

- Необязательную целую часть.
- Обязательный символ точки, отделяющий целую часть от дробной.
- Необязательную дробную непериодическую часть.
- Необязательную периодическую дробную часть, записываемую в круглых скобках.

Необходимо определить значение этой дроби, сохранить его в переменной типа `double` и вывести на экран с точностью не менее 12 знаков. Общая длина входной строки не превосходит 30 символов.

Input	Output
0.1	0.5
0.(01)	0.33333333333333331483

### D. Перевод рационального числа в периодическую двоичную дробь

Дано рациональное число. Запишите его в виде двоичной периодической дроби.

На вход программа получает два натуральных числа  $N$  и  $M$ , каждое из которых не превосходит 1000.

Программа должна вывести значение  $N/M$ , записанное в виде двоичной периодической дроби, при этом длина непериодической дробной части и длина периода должны быть минимально возможными. Если данное число является конечной двоичной дробью, периодическую часть выводить не надо.

Input	Output
1 2	0.1
1 3	0.(01)

### E. Перевод двоичной периодической дроби в рациональное число

Дана запись двоичной дроби, как в задаче A, но в целых числах точки может не быть.

Необходимо представить ее в виде несократимой рациональной дроби  $N/M$ .

Программа должна вывести значения  $N$  и  $M$ .

Input	Output
0.1	1 2
0.(001)	1 7

### F. Перевод рационального числа в периодическую десятичную дробь

Напишите программу, которая переводит правильную дробь в десятичную, выделив (если нужно), период дроби.

Входная строка содержит два числа  $M$  и  $N$ , разделённых пробелами. Гарантируется, что  $M < N$ .

Программа должна вывести десятичную запись дроби, выделив (если нужно) её период. В качестве разделителя целой и дробной части используйте точку.

Input	Output
1 2	0.5
3 14	0.2(142857)

G. *Перевод из любой системы в любую*

Напишите программу, переводящую запись числа между двумя произвольными системами счисления.

На вход программа получает три величины:  $N, A, K$ , где  $N$  и  $K$  — натуральные числа, основания системы счисления ( $2 \leq N, K \leq 36$ ),  $A$  — целое число, записанное в системе счисления с основанием  $N$ ,  $0 \leq A < 2^{31}$ .

Необходимо вывести значение  $A$  в системе счисления с основанием  $K$  без лидирующих нулей.

Input	Output
2 101111 16	2F
10 35 36	Z

H. *Произвольная СС: инкремент*

Первая строка входных данных содержит последовательность символов '0', ..., '9', 'A', ..., 'Z', являющейся записью некоторого неотрицательного числа в системе счисления с основанием `base`.

Длина числа не превосходит 100000 символов. Вторая строка входных данных содержит основание системы счисления `base`, не превосходящее 36.

Увеличьте это число на 1 и выведите результат в той же системе счисления.

Input	Output
19A 11	1A0

I. *Произвольная СС: декремент*

Первая строка входных данных содержит последовательность символов '0', ..., '9', 'A', ..., 'Z', являющейся записью некоторого неотрицательного числа в системе счисления с основанием `base`.

Длина числа не превосходит 100000 символов. Вторая строка входных данных содержит основание системы счисления `base`, не превосходящее 36.

Уменьшите это число на 1 и выведите результат в той же системе счисления.

Input	Output
1A0 11	19A

J. *Произвольная СС: сложение*

Первая и вторая строки входных данных содержат последовательности символов '0', ..., '9', 'A', ..., 'Z', являющиеся записью некоторых неотрицательных чисел в системе счисления с основанием `base`. Длина чисел не превосходит 100000 символов. Третья строка входных данных содержит основание системы счисления `base`, не превосходящее 36. Гарантируется, что первые две строки являются корректной записью чисел в системе счисления с основанием `base`.

Выведите результат сложения данных чисел в той же системе счисления.

Input	Output
2 2 3	11
AAA 1 11	1000
245DSF 1010101Z 36	10346DUE

Рассмотрим последовательность Фибоначчи:  $F_1 = 1, F_2 = 2$  и  $F_n = F_{n-1} + F_{n-2}$  при  $n > 2$ . Первые элементы этой последовательности таковы:  $F_3 = 2, F_4 = 3, F_5 = 5, F_6 = 8$  и т.д.

Любое натуральное число можно представить в виде суммы нескольких членов последовательности Фибоначчи. Такое представление будет неоднозначным (приведите пример). Однако, если потребовать, чтобы в представлении не было двух соседних членов последовательности Фибоначчи, то представление становится единственным (докажите!). Будем говорить, что число  $A$  представимо в фибоначчиевой системе счисления (ФСС) в виде  $a_k a_{k-1} \dots a_1$ , где  $a_i \in \{0; 1\}$ , если  $A = a_k \cdot F_k + \dots + a_1 \cdot F_1$  и в записи  $a_k a_{k-1} \dots a_1$  нет двух единиц подряд.

Вот как записываются небольшие числа в фибоначчиевой системе счисления:

Десятичная	0	1	2	3	4	5	6	7	8	9	10	11	12
Фибоначчиева	0	1	10	100	101	1000	1001	1010	10000	10001	10010	10100	10101

**К. Перевод из ФСС (фибоначчиевой СС) в десятичную**

Дана запись числа в фибоначчиевой системе счисления. Запишите его в десятичной системе счисления.

Программа получает на вход строку из символов 0 и 1 и должна вывести одно целое число. Гарантируется, что результат принимает значения от 0 до  $2 \cdot 10^9$ .

Input	Output
10101	12

**L. Перевод из десятичной в ФСС**

Дано целое число от 0 до  $2 \cdot 10^9$ . Выведите его представление в фибоначчиевой системе счисления без лидирующих нулей.

Input	Output
12	10101

**M. ФСС: инкремент**

Дано целое неотрицательное число  $N$ , записанное в фибоначчиевой системе счисления, длина числа не превосходит 100000 символов. Выведите значение числа  $N + 1$  в фибоначчиевой системе счисления.

Input	Output
100101	101000

**N. ФСС: декремент**

Дано целое неотрицательное число  $N$ , записанное в фибоначчиевой системе счисления, длина числа не превосходит 100000 символов. Выведите значение числа  $N - 1$  в фибоначчиевой системе счисления.

Input	Output
101000	100101

**O. ФСС: сложение**

Даны два числа, записанные в фибоначчиевой системе счисления, длины чисел не превосходят 100000 символов. Выведите значение их суммы в фибоначчиевой системе счисления.

Input	Output
10010 10101	1000001

В троичной взвешенной системе счисления (ТВСС) используется основание 3 и три цифры: 0, 1 и  $-1$ . Цифру  $-1$  будем обозначать знаком  $\$$ . Достоинство уравновешенной троичной системы счисления: простота хранения отрицательных чисел и удобство нахождения числа, противоположного данному.

Вот как записываются небольшие числа в уравновешенной троичной системе счисления:

Десятичная	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9
ТВСС	\$00	\$01	\$1\$	\$10	\$11	\$\$	\$0	\$1	\$	0	1	1\$	10	11	1\$\$	1\$0	1\$1	10\$	100

**P. Перевод из ТВС в десятичную**

Дана запись числа в уравновешенной троичной системе счисления. Переведите его в десятичную. Гарантируется, что ответ не превосходит по модулю  $10^9$ .

Input	Output
\$01	-8

**Q. Перевод из десятичной в ТВС**

Дано целое число от  $-2 \cdot 10^9$  до  $2 \cdot 10^9$ . Выведите его представление в уравновешенной троичной системе счисления без лидирующих нулей.

Input	Output
-8	\$01

R. *TBC: инкремент*

Дана запись некоторого числа в уравновешенной троичной системе счисления, длина записи не превосходит 100000 символов.

Увеличьте это число на 1 и выведите его значение в той же системе.

Input	Output
\$01	\$1\$

S. *TBC: декремент*

Дана запись некоторого числа в уравновешенной троичной системе счисления, длина записи не превосходит 100000 символов.

Уменьшите это число на 1 и выведите его значение в той же системе.

Input	Output
\$1\$	\$01

T. *TBC: сложение*

Дано два числа, записанных в уравновешенной троичной системе счисления. Выведите их сумму без лидирующих нулей. Длины входных чисел не превосходят 100000 символов.

Input	Output
1\$\$\$ \$0\$	11

Пояснение: приведённый пример соответствует выражению  $14 + (-10) = 4$ .

U. *Красивые числа*

Требуется по заданным числам  $N$  и  $K$  найти такое  $D$ , чтобы число  $N$  в системе счисления с основанием  $D$  заканчивалось как можно большим количеством цифр  $K$ .

Вводятся два целых десятичных числа  $N$  и  $K$  ( $1 \leq N \leq 10^{11}; 0 \leq K \leq 9$ ).

Выведите два числа:  $d$  — искомое основание системы счисления и  $L$  — количество цифр  $K$ , которым заканчивается запись числа  $N$  в этой системе счисления. Если искомым  $D$  несколько, выведите любое из них, не превосходящее  $10^{12}$  (такое всегда существует).

Input	Output
49 1	3 2
7 5	3 0
4 4	5 1
9 9	10 1

V. *Круглые факториалы*

Требуется по заданным числам  $N$  и  $K$  найти количество нулей в конце записи в системе счисления с основанием  $K$  числа  $N! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (N-1) \cdot N$ .

В первой строке входных данных содержатся числа  $N$  и  $K$ , разделённые пробелом,  $1 \leq N \leq 10^9, 2 \leq K \leq 1000$ .

Выведите число  $X$  — количество нулей в конце записи числа  $N!$  в системе счисления с основанием  $K$ .

Input	Output
5 10	1
1 2	0
100 10	24
1000 10	249

W. *Дырявое множество*

Определим множества  $K[i]$  рекуррентно. Пусть  $K[0] = [0, 1]$ . Разделим сегмент  $[0, 1]$  на три части точками  $\frac{1}{3}$  и  $\frac{2}{3}$  и удалим из него интервал  $(\frac{1}{3}, \frac{2}{3})$ . Получим множество  $K[1]$ , состоящее из двух

оставшихся сегментов  $[0, \frac{1}{3}]$  и  $[\frac{2}{3}, 1]$ .

Каждый из них разделим на три части (точками  $\frac{1}{9}$  и  $\frac{2}{9}$  для первого сегмента, и точками  $\frac{7}{9}$  и  $\frac{8}{9}$  для второго) и удалим средние интервалы  $(\frac{1}{9}, \frac{2}{9})$  и  $(\frac{7}{9}, \frac{8}{9})$ . Таким образом получаем множество

$K[2]$ , и т.д.

Пусть мы построили множество  $K[i]$ . Поделим каждый оставшийся сегмент из  $K[i]$  на 3 части и удалим из этих сегментов средние интервалы. Получим, таким образом, из  $K[i]$  множество  $K[i+1]$ .

Вводятся 3 целых числа  $n, a, b$  ( $a \leq b$ ). Необходимо определить, принадлежит ли точка с координатой  $\frac{a}{b}$  множеству  $K[n]$ .

Input	Output
1 2 4	NO
2 13 18	YES

X. *Счастливые цифры*

Требуется написать программу, которая по заданным числам  $N$  и  $K$  найдет такое  $D$ , чтобы число  $N$  в системе счисления с основанием  $D$  заканчивалось как можно большим количеством цифр  $K$ .

Вводятся два целых десятичных числа  $N$  и  $K$  ( $1 \leq n \leq 10^{11}, 0 \leq k \leq 9$ ).

Выведите два числа:  $D$  — искомое основание системы счисления и  $L$  — количество цифр  $K$ , которым заканчивается запись числа  $N$  в этой системе счисления. Если искомым  $D$  несколько, выведите любое из них, не превосходящее  $10^{12}$  (такое всегда существует).

Input	Output
49 1	3 2
7 5	3 0