

## Задача 1. Последние цифры

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

В каждом целом числе от  $M$  до  $N$  включительно берётся последняя (младшая) цифра, и все такие цифры складываются. Найдите получившуюся сумму.

### Формат входных данных

В первой строке записано целое число  $M$ , во второй — целое число  $N$  ( $1 \leq M \leq N \leq 10^{18}$ ).

### Формат выходных данных

Выведите одно целое число — ответ.

### Система оценки

Подзадача 1 (до 50 баллов):  $1 \leq M \leq N \leq 10^6$ .

Подзадача 2 (до 50 баллов):  $1 \leq M \leq N \leq 10^{18}$ .

Каждый тест оценивается независимо. Участнику сообщаются результаты проверки на каждом тесте.

### Пример

стандартный ввод	стандартный вывод
7	27
12	

### Замечание

Обратите внимание, что входные данные во второй подзадаче могут быть достаточно большими и не помещаться в 32-битные типы данных. Рекомендуется использовать 64-битный тип данных, например, тип *long long* в языке C++, тип *int64* в языке Pascal, тип *long* в языках Java и C#. Язык Python автоматически работает с целыми числами любой длины.

## Задача 2. Парные носки

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Под кроватью у Васи лежат  $N$  носков. Каждому носку присвоено целое число, обозначающее цвет (точнее, оттенок чёрного). Вася считает, что два носка могут образовать пару, если модуль разности их цветов не превосходит  $K$ .

Помогите Васе составить из этих носков максимальное количество пар. При этом, если существует несколько решений, то нужно найти такое, в котором сумма цветов носков, не вошедших в пары, будет минимальной.

### Формат входных данных

В первой строке входных данных записано целое число  $N$  ( $1 \leq N \leq 10^5$ ). Во второй строке записано целое число  $K$  ( $0 \leq K \leq 1000$ ). В следующих  $N$  строках записано по одному целому числу из диапазона от 1 до 1000 — цвет каждого носка.

### Формат выходных данных

В первой строке выведите максимальное количество пар, которое можно составить. Во второй строке выведите минимальную сумму цветов всех непарных носков, которую при этом можно получить.

### Система оценки

Подзадача 1 (до 60 баллов):  $N \leq 100$ .

Подзадача 2 (до 40 баллов):  $N \leq 10^5$ .

Каждый тест оценивается независимо. Участнику сообщаются результаты проверки на каждом тесте.

### Пример

стандартный ввод	стандартный вывод
5	2
1	3
3	
4	
7	
3	
8	

### Замечание

В примере можно составить две пары —  $\langle 3, 4 \rangle$  и  $\langle 7, 8 \rangle$ , при этом ещё один носок цвета 3 останется непарным.

## Задача 3. Параллельные вычисления

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Существует несколько способов записи арифметических выражений. В привычной нам (инфиксной) записи знак операции ставится между аргументами — например:  $(10 - 6) * (3 + 5)$ . В постфиксной (или обратной польской) записи знак операции пишется после аргументов. Например, это же выражение в постфиксной записи будет выглядеть так:  $10 6 - 3 5 + *$ .

Как видно из примера, в постфиксной записи не используются скобки, поскольку операнды для каждой операции всегда определяются однозначно.

Вычисление выражения в постфиксной записи выполняется очень просто. Идём слева направо. Встретив знак операции, берём два операнда перед ним, выполняем операцию, стираем её и оба операнда, а на это место записываем результат. Возьмём для примера наше выражение  $10 6 - 3 5 + *$ . На первом шаге выполним вычитание — получится  $4 3 5 + *$ . Теперь выполним сложение — получаем  $4 8 *$ . На последнем шаге выполняем умножение, и получаем ответ 32.

Представим теперь, что операнды у нас — не короткие числа, а очень длинные (содержащие миллионы цифр). Тогда операции с ними будут выполняться медленно. Пусть наш компьютер выполняет одну операцию за одну секунду. Тогда, чтобы вычислить выражение из примера (но с длинными операндами), ему потребуется три секунды.

Однако, современные компьютеры умеют производить вычисления параллельно! Если в нашем примере операции вычитания и сложения выполнять одновременно, то вычисление выражения займёт всего две секунды вместо трёх. Более формально: любую операцию можно начинать выполнять сразу, как только оба её операнда уже вычислены.

Напишите программу, решающую следующую задачу. На вход программы подаётся выражение в постфиксной записи. Определите, за какое минимальное время можно вычислить это выражение.

### Формат входных данных

На вход программе подаётся строка, содержащая корректную запись выражения в постфиксной записи. Операнды обозначаются строчными английскими буквами, все буквы в строке различны. Знаками операций могут быть символы '+', '-', '\*', '/'. Элементы выражения разделяются пробелом. Длина строки не превышает 101 символ.

### Формат выходных данных

Выведите одно целое число — минимальное число секунд, за которое можно вычислить выражение.

### Система оценки

Подзадача 1 (до 40 баллов): выражение содержит не более трёх знаков операций.

Подзадача 2 (до 30 баллов): выражение содержит не более шести знаков операций.

Подзадача 3 (до 30 баллов): дополнительные ограничения отсутствуют.

Каждый тест оценивается независимо. Участнику сообщается результат проверки каждого теста.

### Примеры

стандартный ввод	стандартный вывод
$a b - c d + *$	2
$a b + c - d +$	3

### Замечание

Возможно, вы заметили, что можно улучшить распараллеливание с помощью эквивалентных преобразований выражений — например,  $a b + c - d +$  заменить на равносильное выражение  $a b + c d - -'$ . Однако, в данной задаче преобразования выражений не используются.

## Задача 4. Количество троек

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1.5 секунд  
Ограничение по памяти: 256 мегабайт

Дана последовательность из  $N$  целых положительных чисел  $a_1, a_2, \dots, a_N$ . Найдите количество таких троек индексов  $i < j < k$ , что  $a_i$  делится на  $a_j$ , и  $a_j$  делится на  $a_k$ .

### Формат входных данных

В первой строке записано целое число  $N$  ( $3 \leq N \leq 10^5$ ). В следующих  $N$  строках записаны  $N$  целых чисел  $a_i$  ( $1 \leq a_i \leq 10^5$ ).

### Формат выходных данных

Выведите одно целое число – ответ.

### Система оценки

Подзадача 1 (до 50 баллов):  $N \leq 300$ .

Подзадача 2 (до 30 баллов):  $N \leq 3000$ .

Подзадача 3 (до 20 баллов):  $N \leq 10^5$ .

Каждый тест оценивается независимо. Участнику сообщаются результаты проверки на каждом тесте.

### Пример

стандартный ввод	стандартный вывод
5	5
12	
6	
12	
4	
2	

### Замечание

В третьей подзадаче ответ может получаться достаточно большим и не помещаться в 32-битный тип данных. Рекомендуется использовать 64-битный тип данных, например, тип `long long` в языке C++, тип `int64` в языке Pascal, тип `long` в языках Java и C#. Язык Python автоматически работает с целыми числами любой длины.

## Задача 5. Полигон для роботов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

На кружке по робототехнике, куда ходит Вася, имеется полигон для испытания собираемых роботов. Полигон представляет собой квадратный стол, на котором расчерчено клетчатое поле размером  $N \times N$  клеток. В некоторых клетках устанавливаются препятствия, и роботы вынуждены их объезжать.

Недавно Вася собрал нового робота, и теперь хочет проверить его работу. Вася помещает робота в клетку с координатами  $(1, 1)$  и проверяет, сможет ли он добраться до клетки  $(N, N)$ . За один шаг робот может переместиться в соседнюю по стороне клетку, если в ней нет препятствия. Робот не может выходить за пределы полигона.

Вася хочет разместить препятствия на полигоне так, чтобы кратчайший путь из клетки  $(1, 1)$  в клетку  $(N, N)$  содержал ровно  $K$  клеток. Помогите ему это сделать.

### Формат входных данных

В первой строке входных данных записано целое число  $N$  ( $3 \leq N \leq 100$ ).

Во второй строке записано целое число  $K$  ( $K \geq 2N - 1$ ;  $2K \leq N^2 + 1$ ).

### Формат выходных данных

Выведите схему полигона —  $N$  строк по  $N$  символов '.' или '#' в каждой строке. Символ '.' означает свободную клетку, символ '#' — клетку с препятствием. Верхняя левая клетка на схеме имеет координаты  $(1, 1)$ , правая нижняя — координаты  $(N, N)$ .

Вы можете вывести любой верный ответ. Если решения нет, выведите "No solution" (без кавычек).

### Система оценки

Подзадача 1 (до 60 баллов):  $3 \leq N \leq 7$ .

Подзадача 2 (до 40 баллов):  $3 \leq N \leq 100$ .

Каждый тест оценивается независимо. Участнику сообщаются результаты проверки на каждом тесте.

### Примеры

стандартный ввод	стандартный вывод
5 11	..... ###.. ..... .#.## .....
4 8	No solution