

Задача А. Ультраотрезок

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

В Юпитерском Государственном Университете ученые решили заняться наукой. Они долго сёрфили интернет в поисках того, что ещё не открыто. Завести к британским ученым не было предела: казалось, что они уже открыли всё. Тогда один из ведущих ученых решил, что они сами придумают то, что они собираются открыть. Таким образом к вечеру из-под его пера появился он — ультраотрезок.

Ультраотрезок — это такой отрезок целочисленной прямой, который включает ультраделитель. В свою очередь, *ультраделитель* — это целое число, которое способно нацело поделить любое другое целое число больше себя.

В данный момент ученые смогли точно определить, что $[1..1\,000\,000\,000]$ точно является ультраотрезком, а, например $[256\,000..300\,000]$ — не является.

Работа в университете движется медленно из-за бюрократических проволочек, а ведь наука не может ждать. Если и дальше медлить, то британские учёные могут забрать всю славу себе.

Поэтому ученые Юпитерского Государственного Университета просят вас написать программу, которая по левому концу отрезка l и правому концу отрезка r будет определять, является ли данный отрезок ультраотрезком.

Формат входных данных

Вводятся натуральные числа l и r ($1 \leq l \leq r \leq 10^9$) — граница отрезка.

Формат выходных данных

Выведите «YES», если отрезок является ультраотрезком, и «NO» в противном случае.

Система оценки

В этой задаче 10 тестов, каждый из которых оценивается в 10 баллов.

Примеры

стандартный ввод	стандартный вывод
1 1000000000	YES
256000 300000	NO

Замечание

Целое число a нацело делит целое число b , если при делении не образуется остаток. Например 3 нацело делит 6 ($6/3 = 2(0)$), но при этом не делит нацело 7 ($7/3 = 2(1)$).

Задача В. Стипендия

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В Юпитерском Государственном Университете закончилась летняя сессия, и скоро нужно будет выплачивать стипендии за новый семестр.

Университет выплачивает два типа стипендий:

- *Повышенная* стипендия выплачивается, если студент закончил сессию только на «отлично»
- *Обычная* стипендия выплачивается, если студент закончил сессию на «хорошо» и «отлично», и не получает повышенную стипендию

Для мотивации студентов размер повышенной стипендии должен быть хотя бы в два раза больше обычной. При этом размеры стипендий не обязаны быть нецелыми числами.

В Университете на выплату стипендий в осеннем семестре заложено m дублей. Чтобы улучшить показатели годовых отчетов и привлечь больше абитуриентов, стипендиальная комиссия хочет назначить такие размеры стипендий, чтобы *средняя величина стипендии* оставалась как можно выше. Напоминаем, что среднее арифметическое набора чисел — это сумма чисел набора, деленная на их количество. Например, для набора чисел $\{70, 130, 4\,000\}$ среднее будет равно 1 400.

Известно, что по итогам летней сессии в Университете a студентов имеют оценки «хорошо» и «отлично», **из них** b студентов — только «отлично».

Помогите стипендиальной комиссии найти максимально возможный средний размер стипендии!

Формат входных данных

Вводятся натуральные числа m , a и b ($1 \leq m \leq 10\,000$, $1 \leq b \leq a \leq 10\,000$) — размер стипендиального фонда, число стипендиатов и число отличников в Университете.

Формат выходных данных

Выведите максимально возможный средний размер стипендии с точностью не менее шести знаков после десятичной точки.

Система оценки

В этой задаче 10 тестов, каждый из которых оценивается в 10 баллов.

Пример

стандартный ввод	стандартный вывод
100 5 2	20.000000000000

Замечание

В примере для достижения максимума можно установить обычную стипендию в 12 дублей, а повышенную — в 32 дубля. Тогда средний размер будет равен $\frac{12 \cdot 3 + 32 \cdot 2}{5} = 20$ дублей.

Задача С. Постулат Бер...на

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

После успешного исследования ультраотрезка учеными Юпитерского Государственного Университета государство выделило кучу инопланетной валюты на исследования, а значит исследования надо продолжать. Ведущий ученый Юпитерского Государственного Университета решил развеять мифы британских ученых.

Пока ученый думал, что он будет опровергать, краем глаза заметил, что на его подставке для кофе что-то написано. Недолго думая, ученый взял бумагу и всю испачканную разводами подставку.

Не всё наш ученый смог прочитать. После долгого восстановления текста у него получилось: *Постулат Бер...на: Для любого натурального n ... найдется простое число p , которое лежит в интервале $n < p < 2n$.* Никаких учёных с подобной фамилией он не помнил. «Значит, из Британии,» — заключил ведущий учёный. После этого он с яростью принялся проверять данный постулат. «Ха!» — крикнул учёный через несколько секунд. Он сразу нашёл натуральное число, которое не подходит под условие — единицу. Радости не было предела. Правда, проверка двойки снова опечалила учёного.

Рабочий день учёного подошёл к концу, и ему пора домой. Он просит вас о помощи, так как возможно одного числа, которое противоречит постулату, будет недостаточно научному миру.

Он просит вас написать программу, которая при введении натурального n будет проверять его на соответствие постулату. Учёный слишком мнительный, поэтому при положительном ответе, он просит вас вывести простое число, которое удалось вам обнаружить в следующей строке.

Формат входных данных

Вводится натуральное число n ($1 \leq n \leq 10^{15}$).

Формат выходных данных

Выведите «NO», если в интервале $(n, 2n)$ нет простых чисел.

Иначе выведите «YES», а в следующей строке одно из найденных простых чисел, лежащих в интервале.

Система оценки

Решения, работающие для $n \leq 10^6$, оцениваются в 55 баллов.

Примеры

стандартный ввод	стандартный вывод
2	YES 3
1	NO

Замечание

Простым числом называется число, имеющее ровно два различных делителя — себя и единицу.

Обратите внимание, что ограничения в этой задаче достаточно велики. В языках C++ и Pascal используйте 64-битные типы данных (`long long` в C++, `int64` в Pascal).

Задача D. Лестница

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В новом корпусе Юпитерского государственного университета построили лестницу. Поскольку денег на строительство лестницы выделено не было, все ступеньки получились разными — i -тая ступенька находится на высоте h_i сантиметров от уровня первого этажа.

Выяснилось, что по СанПиНам каждая ступенька должна быть выше не более, чем на x сантиметров от предыдущей, а значит, лестницу нужно успеть переделать до приезда проверки! Было решено осуществить ремонт следующим образом: вставить дополнительные ступеньки между уже имеющимися.

Помогите руководству университета найти **минимальное количество ступенек**, которые придется достроить.

Формат входных данных

В первой строке вводится два натуральных числа n и x ($1 \leq n \leq 10^5$, $1 \leq x \leq 10^9$) — количество уже построенных ступенек на лестнице и минимальная разница высот между соседними ступеньками.

Во второй строке вводятся n натуральных чисел a_i ($1 \leq a_i \leq 10^9$) — высоты уже построенных ступенек. Гарантируется, что высоты возрастают, то есть $a_i < a_{i+1}$.

Формат выходных данных

Выведите одно число — минимальное количество ступенек, которые необходимо достроить.

Система оценки

Решения, работающие для $n = 4$, $x, a_i \leq 800$, оцениваются в 19 баллов.

Решения, работающие для $n, x, a_i \leq 800$, оцениваются в 35 баллов.

Решения, работающие для $n \leq 1\,000$, $x, a_i \leq 10^5$, оцениваются в 60 баллов.

Примеры

стандартный ввод	стандартный вывод
4 1 1 2 5 6	2
4 3 1 2 5 6	0

Замечание

В первом примере необходимо добавить ступеньки на высоте 3 и 4.

Во втором примере ступенек уже достаточно.

Задача Е. Факториал

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

Человека, ответственного за написание данного условия, переманили на тёмную сторону силы (в Юпитерский Государственный Университет), поэтому условие этой задачи будет максимально сухим и сжатым.

Вам требуется найти последнюю ненулевую цифру факториала n . Напомним, что *факториалом* числа n называется число $n! = 1 \times 2 \times \dots \times (n - 1) \times n$. Например, $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$, а последняя ненулевая цифра будет равна 2.

Формат входных данных

Вводится натуральное число n ($1 \leq n \leq 100\,000$).

Формат выходных данных

Выведите последнюю ненулевую цифру $n!$.

Система оценки

Решения, работающие для $n \leq 20$, оцениваются в 25 баллов.

Решения, работающие для $n \leq 2\,000$, оцениваются в 40 баллов.

Пример

стандартный ввод	стандартный вывод
5	2

Задача F. Рунология

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

Ученые Юпитерского Государственного Университета заинтересовались такой наукой, как рунология.

В руки учёных попало древнее сообщение, которое предположительно состоит из последовательно записанных анаграмм рун. *Анаграмма* слова — это любое слово, которое состоит из тех же букв, но не обязательно в том же порядке. Например, для слова *дом* анаграммами будут *дом*, *дмо*, *мод*, *мдо*, *одм*, *омд*.

Древние люди не хотели, чтобы сообщение быстро расшифровали, поэтому добавили в произвольные места сообщения лишних букв. В секретных документах учёные обнаружили все используемые руны и их порядок в этом сообщении.

Вы, как истинный аутсорсер, ~~должны сделать всю работу за них~~ помочь им в этом разбив строку на несколько частей, состоящих из последовательно идущих символов, таким образом, чтобы в каждой части лежала только одна руна. Части должны следовать в том же порядке, что и содержащиеся в них руны во входных данных.

Формат входных данных

В первой строке вводится натуральное число n — количество рун в сообщении.

В следующей строке вводится непустая строка из строчных латинских букв s — само древнее сообщение. Длина строки не превышает 300 000 символов.

Далее следует n непустых строк t_i из строчных латинских букв — сами руны, из анаграмм которых состоит сообщение.

Гарантируется, что сообщение содержит анаграммы всех рун в порядке их появления во входных данных.

Формат выходных данных

Выведите n строк, в каждой из которых должна быть пара чисел l_i и r_i — номера символов левого и правого конца куска строки, содержащего анаграмму руны t_i .

l_1 должно быть равно 1, r_n должно быть равно длине строки s , а l_{i+1} должно быть равно $r_i + 1$ для всех i — то есть вы должны вывести разбиение строки в порядке от начала до конца.

Если существует несколько возможных ответов, выведите любой.

Система оценки

Решения, работающие для строк s длиной не более 125 символов без лишних букв, оцениваются в 11 баллов.

Решения, работающие для строк s длиной не более 1 500 символов, оцениваются в 42 балла.

Решения, работающие для строк s длиной не более 15 000 символов и строк t_i длиной не более 1 000, оцениваются в 90 баллов.

Примеры

стандартный ввод	стандартный вывод
4 ihistemmario	1 2 3 5 6 7 8 12
me mario	
2 zezrdazzierzzfnzd	1 6 7 17
dear friend	

Замечание

Рассмотрим второй пример. У древних людей было две руны: **dear** и **friend**. После перемешивания букв получились анаграммы **erda** и **ierfnd**, а после добавления лишних букв — **zezrdazzierzzfnzd** (в качестве лишних букв не обязательно используются **z**, на их месте могут быть и другие символы).

Чтобы в каждом куске строки было по одной анаграмме руны, разобьем её на кусок из 6 букв **zezrda** и кусок из 11 букв — **zzierzzfnzd**. Тогда ответ будет:

1 6

7 17

Также бывают и другие ответы, например,

1 8

9 17