Задача A. Длина подпоследовательности

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Вам требуется написать программу, которая по заданной последовательности находит длину её максимальной невозрастающей подпоследовательности (то есть такой последовательности чисел $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$ ($i_1 < i_2 < \ldots < i_k$), что $a_{i_1} \geqslant a_{i_2} \geqslant \ldots \geqslant a_{i_k}$ и не существует последовательности с теми же свойствами длины $k + 1$).

Формат входных данных

В первой строке задано число n — количество элементов последовательности ($1 \leqslant n \leqslant \mathbf{239\,017}$). В последующих строках идут сами числа последовательности a_i, отделённые друг от друга произвольным количеством пробелов и переводов строки (все числа не превосходят по модулю $2^{31} - 2$).

Формат выходных данных

Вам необходимо выдать число k — длину максимальной невозрастающей подпоследовательности.

Пример

стандартный ввод	***стандартный вывод***

`5`
`  5 8`
`10     4  1`
	`3`

Задача B. Невозрастающая подпоследовательность

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Вам требуется написать программу, которая по заданной последовательности находит максимальную невозрастающую её подпоследовательность (то есть такую последовательность чисел $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$ ($i_1 < i_2 < \ldots < i_k$), что $a_{i_1} \geqslant a_{i_2} \geqslant \ldots \geqslant a_{i_k}$ и не существует последовательности с теми же свойствами длины $k+1$).

Формат входных данных

В первой строке задано число n — количество элементов последовательности ($1 \leqslant n \leqslant \mathbf{239\,017}$). В последующих строках идут сами числа последовательности a_i, отделённые друг от друга произвольным количеством пробелов и переводов строки (все числа не превосходят по модулю $2^{31} - 2$).

Формат выходных данных

Вам необходимо выдать в первой строке число k — длину максимальной невозрастающей подпоследовательности. В последующих строках должны быть выведены (по одному числу в каждой строке) все номера элементов исходной последовательности i_j, образующих искомую подпоследовательность. Номера выводятся в порядке возрастания. Если оптимальных решений несколько, разрешается выводить любое.

Пример

<table>
<tr><th>стандартный ввод</th><th>стандартный вывод</th></tr>
<tr><td><pre>5
  5 8
10   4 1
</pre></td><td><pre>3
2
4
5</pre></td></tr>
</table>

Задача C. Игра в деление

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

На доске записано целое число $n > 1$. Ивонна и Зара играют в игру, делая ходы по очереди, начинает Ивонна. За каждый ход игрок может либо поделить число на 2 и округлить вниз, либо поделить на 3 и округлить вверх. После этого старое число стирается, а новое записывается на доску.

Выигрывает та, кто первой запишет на доске число 1. Кто выиграет при правильной игре?

Формат входных данных

В первой строке задано целое число n ($1 < n \leqslant 10^{18}$).

Формат выходных данных

Если при правильной игре выигрывает Ивонна, выведите «`Yvonne`».

Если при правильной игре выигрывает Зара, выведите «`Zara`».

Буквы можно выводить как в верхнем, так и в нижнем регистре.

Примеры

стандартный ввод	***стандартный вывод***
`2`	`Yvonne`
`4`	`Zara`

Задача D. Скобочная запись

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

В этой задаче требуется найти кратчайшую правильную скобочную последовательность, сопоставленную заданному числу.

Как вы, возможно, знаете, множество *правильных скобочных последовательностей* $\mathcal{S}$ определяется рекурсивно следующим образом:

1. $\varepsilon \in \mathcal{S}$ (пустая строка)
2. $A \in \mathcal{S} \Rightarrow \texttt{(}A\texttt{)} \in \mathcal{S}$ (взятие в скобки)
3. $A, B \in \mathcal{S} \Rightarrow AB \in \mathcal{S}$ (конкатенация)

Например, «`(())`» $\in \mathcal{S}$ по правилу 2 и потому, что «`()`» $\in \mathcal{S}$; это, в свою очередь, верно по правилу 2 и потому, что «» $\in \mathcal{S}$; последнее верно по правилу 1. Аналогично, «`()()`» $\in \mathcal{S}$ по правилу 3 и потому, что «`()`» $\in \mathcal{S}$; последнее доказано выше.

Сопоставим правильным скобочным последовательностям $X \in \mathcal{S}$ числа $f(X)$ по следующим правилам:

1. $\varepsilon \longrightarrow 1$
2. $\texttt{(}A\texttt{)} \longrightarrow 1 + f(A)$
3. $AB \longrightarrow f(A) \cdot f(B)$

Можно показать, что сопоставление корректно, так как не зависит от порядка вычислений. Заметим, что различным последовательностям может быть сопоставлено одно и то же число.

По заданному числу найдите правильную скобочную последовательность, сопоставленную этому числу. Найденная вами последовательность должна иметь минимальную возможную длину. Если возможных ответов несколько, можно найти любой из них.

Каждый тест в этой задаче — это некоторый отрезок $[a; b]$ целых чисел, причём $1 \leqslant a \leqslant b \leqslant 10^9$, а количество целых чисел на этом отрезке лежит в пределах от 1 до 10.

Формат входных данных

В первой строке ввода содержатся два целых числа a и b ($1 \leqslant a \leqslant b \leqslant 10^9$). Задачу следует решить для всех целых чисел n от a до b включительно в порядке возрастания. Гарантируется, что в каждом тесте количество таких чисел n — от 1 до 10 включительно.

Формат выходных данных

Для каждого n выведите в отдельной строке скобочную последовательность минимальной возможной длины, соответствующую этому числу n. Если возможных ответов несколько, выведите любой из них.

Пример

стандартный ввод	*стандартный вывод*
`3 4`	`(())` `()()`

Задача E. Новая модель телефона

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Компания Gnusmas разработала новую модель мобильного телефона. Основное достоинство этой модели — ударопрочность: её корпус сделан из особого сплава, и телефон должен выдерживать падение с большой высоты.

Компания Gnusmas арендовала n-этажное здание и наняла экспертов, чтобы те при помощи серии экспериментов выяснили, с какой высоты бросать телефон можно, а с какой — нельзя. Один эксперимент заключается в том, чтобы бросить телефон с какого-то этажа и посмотреть, сломается он от этого или нет. Известно, что любой телефон этой модели ломается, если его сбросить с x-го этажа или выше, где x — некоторое целое число от 1 до n, включительно. Задача экспертов заключается в том, чтобы узнать число x и передать его рекламному отделу компании.

Задача осложняется тем, что экспертам предоставлено всего k образцов новой модели телефона. Каждый телефон можно бросать сколько угодно раз, пока он не сломается; после этого использовать его для экспериментов больше не удастся.

Подумав, эксперты решили действовать так, чтобы минимизировать максимально возможное количество экспериментов, которое может потребоваться произвести. Чему равно это количество?

Формат входных данных

В первой строке входного файла записаны через пробел два целых числа n и k — количество этажей в здании и количество образцов новой модели телефона ($1 \leqslant n \leqslant 100\,000$, $0 \leqslant k < n$).

Формат выходных данных

В выходной файл выведите единственное число — минимальное количество экспериментов, которое потребуется совершить, чтобы узнать число x и использовать не более k телефонов. Если решить задачу невозможно, выведите вместо этого -1.

Примеры

стандартный ввод	**стандартный вывод**
4 2	2
4 1	3

Пояснения к примерам

В первом примере сначала следует бросить телефон со второго этажа. Если он сломается, то второй бросок следует сделать с первого этажа. В случае поломки станет известно, что $x = 1$. Иначе мы узнаем, что $x = 2$.

Если же при броске со второго этажа телефон не сломался, бросим телефон с третьего этажа. При поломке будет ясно, что $x = 3$. Иначе из условия $1 \leqslant x \leqslant 4$ следует, что $x = 4$.

Всего будет сделано два эксперимента. В них будет использовано не более чем два телефона.

Во втором примере следует сначала бросить единственный данный нам телефон с первого этажа, если он не сломается, то со второго, а если опять не сломается, то с третьего. При первой же поломке мы узнаем точное значение x. Если после трёх бросков телефон так и не сломался, то $x = 4$.

Задача F. Аттракцион

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

В парке поставили экспериментальную модель аттракциона. Посетители рассаживаются по местам, затем аттракцион запускают, и посетители с головокружительной скоростью носятся по причудливым траекториям. К сожалению, аттракцион не всегда работает: иногда он ломается, и все, кто в этот момент был на аттракционе, уходят недовольными и больше в парк не приходят. Было установлено, что аттракцион ломается тогда и только тогда, когда на нём катается не меньше, чем x человек, для некоторого неизвестного заранее, но фиксированного целого числа x ($1 \leqslant x \leqslant n$, где n — общее число мест на аттракционе).

Хозяин парка аттракционов — циничный человек, и он не очень расстроится, если несколько посетителей будут недовольны поломками на аттракционе. Гораздо больше его заботит расход электроэнергии, требуемой для запуска аттракциона. Количество энергии, требуемое для однократного запуска аттракциона, постоянно и не зависит от количества занятых мест и от того, случится ли при работе аттракциона поломка.

Хозяин парка хочет сделать как можно меньше запусков аттракциона, прежде чем ему станет известно, чему в точности равно число x. В его власти количество посетителей при каждом новом запуске аттракциона. Однако хозяин парка знает, что, если в ходе его экспериментов недовольных окажется больше, чем k человек, они соберутся вместе и напишут жалобу в министерство. Конечно, для хозяина парка такой поворот событий совершенно неприемлем.

Примерьте на себя шкуру холодного расчётливого циника и посчитайте, какое минимальное количество запусков потребуется для решения поставленной задачи.

Формат входных данных

В первой строке входного файла записаны через пробел два целых числа n и k — количество мест на аттракционе и максимальное возможное количество недовольных ($1 \leqslant n \leqslant 600$, $0 \leqslant k \leqslant 10^9$).

Формат выходных данных

В выходной файл выведите единственное число — минимальное количество запусков аттракциона, которое потребуется совершить, чтобы узнать число x и сделать не более k человек недовольными. Если решить задачу невозможно, выведите вместо этого -1.

Примеры

стандартный ввод	**стандартный вывод**
`4 3`	`2`
`4 2`	`-1`

Пояснения к примерам

В первом примере сначала следует запустить аттракцион с двумя посетителями. Если он сломается, во второй запуск следует посадить на аттракцион одного человека. В случае поломки всего недовольных будет трое и станет известно, что $x = 1$. В случае успешного запуска недовольных будет двое и мы узнаем, что $x = 2$.

Если же с двумя посетителями аттракцион не сломался, запустим его для трёх человек. При поломке будет три недовольных и $x = 3$. При успешном запуске недовольных нет и из условия $1 \leqslant x \leqslant 4$ получаем, что $x = 4$.

Определить x быстрее, чем за два запуска, в первом примере не удастся.

Во втором примере невозможно посадить на аттракцион более двух человек, из-за чего оказывается, что случаи $x = 3$ и $x = 4$ неразличимы.

Задача G. Перестановки и различные порядки

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Рассмотрим множество всех перестановок из n чисел от 1 до n. *Тождественной* называется перестановка, в которой на первом месте стоит единица, на втором — двойка, и так далее.

Перестановки можно применять друг к другу. При применении перестановки (a_i) к перестановке (b_i) получается перестановка $a(b)$, в которой на i-е место становится элемент a_{b_i}. Это также называют *умножением* перестановок.

В частности, перестановки можно *возводить в степень*. Возвести перестановку a в степень n — значит, выполнить $x = a$, а потом $n - 1$ раз $x = a(x)$.

Известно, что для каждой перестановки существует такое число k, что если её возвести в степень k, получается тождественная перестановка. Минимальное положительное такое k называется *порядком* этой перестановки.

Необходимо сосчитать, сколько различных значений порядков получится для всех перестановок из n элементов. Например, для $n = 2$ есть перестановка порядка 1 $(1, 2)$ и перестановка порядка 2 $(2, 1)$. Для $n = 3$ бывают ещё перестановки порядка 3 (например, $(2, 3, 1)$).

Формат входных данных

Целое n ($1 \leqslant n \leqslant 1000$).

Формат выходных данных

Количество различных возможных порядков.

Примеры

стандартный ввод	**стандартный вывод**
2	2
3	3
4	4
10	16

Задача H. Максимальный порядок

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Перестановка длины n — это последовательность из n чисел, в которой каждое целое число от 1 до n встречается ровно один раз. Например, существует шесть перестановок длины 3: `1 2 3`, `1 3 2`, `2 1 3`, `2 3 1`, `3 1 2` и `3 2 1`.

Тождественная перестановка длины n — это последовательность `1 2 3 ... n`. Тождественная перестановка обозначается буквой e; её длина обычно понятна из контекста.

Произведением перестановок $p = p_1\ p_2\ \ldots\ p_n$ и $q = q_1\ q_2\ \ldots\ q_n$ называется перестановка $r = r_1\ r_2\ \ldots\ r_n$, в которой элементы получаются по формуле $r_i = p_{q_i}$. Произведение обозначается так: $p \cdot q$. Например, произведением перестановок $p =$ `2 4 1 3 5` и $q =$ `4 2 1 5 3` является перестановка $p \cdot q =$ `3 4 2 5 1`. Заметим, что порядок сомножителей важен: например, $q \cdot p =$ `2 5 4 1 3`.

Перестановки также можно возводить в степень. *Степень перестановки* определяется так: $p^0 = e$, а для любого целого $k > 0$ верно $p^k = p \cdot p^{k-1}$. Например, для перестановки $p =$ `2 1 4 5 3` первые несколько степеней выглядят так:

```
0  1 2 3 4 5
1  2 1 4 5 3
2  1 2 5 3 4
3  2 1 3 4 5
4  1 2 4 5 3
5  2 1 5 3 4
6  1 2 3 4 5
7  2 1 4 5 3
       ...
```

Порядок перестановки p — это минимальное целое $k > 0$ такое, что $p^k = e$. Например, порядок перестановки `1 2 3` равен единице, а порядок перестановки `2 1 4 5 3` равен шести.

По данной длине n найдите перестановку, которая имеет максимальный порядок среди всех перестановок длины n.

Формат входных данных

В первой строке задано целое число n — требуемая длина перестановки ($1 \leqslant n \leqslant 100$).

Формат выходных данных

В первой строке выведите максимальный порядок перестановки длины n. Во второй строке выведите любую перестановку длины n, имеющую такой порядок. Соседние числа в перестановке следует разделять пробелами.

Примеры

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

Задача I. Мирные множества

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Группа математиков проводит бои между натуральными числами. Результаты боя между двумя натуральными числами, вообще говоря, случайны, однако подчиняются следующему правилу: если одно из чисел не менее чем в два раза превосходит другое, то большее число всегда побеждает; в противном случае победить может как одно, так и другое число.

Бой называется *неинтересным*, если его результат предопределён. Множество натуральных чисел называется *мирным*, если бой любой пары различных чисел из этого множества неинтересен. *Силой* множества называется сумма чисел в нём. Сколько существует мирных множеств натуральных чисел силы n?

Формат входных данных

В первой строке задано число n ($1 \leqslant n \leqslant 2000$).

Формат выходных данных

В первой строке выведите одно число — количество мирных множеств натуральных чисел силы n.

Примеры

стандартный ввод	**стандартный вывод**
2	1
5	2

Задача J. Требушет Джон

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	8 секунд
Ограничение по памяти:	512 мебибайт

Требушет Джон — основная ударная сила Флатландского королевства. В свободное от бросания девяностопятикилограммовых валунов время Джон любит наблюдать за плоскими лучниками, играющими в блинчики на плоском озере (блинчики — игра, заключающаяся в бросании плоских камней в воду таким образом, чтобы камень несколько раз отскочил от поверхности воды, прежде чем утонет).

Королевство, которому служит Джон, состоит из n одинаковых провинций, каждая из которых имеет размер d. Так, первой провинции принадлежат точки с x-координатами $(0..d]$, второй — точки с x-координатами $(d..2d]$, и так далее. Более того, каждой провинции соответствует свой коэффициент благополучия w_i. Сам Джон располагается в начале координат.

Будучи настоящим мастером своего дела, Джон может выполнить любой бросок, каким бы сложным тот ни казался на первый взгляд, поэтому развлечения простых солдат кажутся ему донельзя занудными, что и побудило его придумать свой аналог блинчиков.

Бросок Джона состоит из прыжков камня. Бросок начинается с того, что Джон выбирает стартовую длину прыжка $k > 0$, после чего забрасывает камень из начала координат в точку k. Для каждого следующего прыжка рассматривается длина предыдущего прыжка p, Джон выбирает какое-то целое число $u > 1$, такое, что p делится нацело на u, и камень прыгает вперёд на расстояние, равное $\frac{p}{u}$. Если такого u не существует, игра заканчивается.

Например, при стартовой длине 6 существует несколько возможных траекторий броска:

- $0 \to 6 \to 9 \to 10$ (при длинах 6, 3, 1 соответственно),
- $0 \to 6 \to 8 \to 9$ (при длинах 6, 2, 1 соответственно),
- $0 \to 6 \to 7$ (при длинах 6, 1 соответственно).

Изначально у Джона 0 очков. Каждый раз, когда камень касается земли, Джону начисляется количество очков, равное коэффициенту благополучия той провинции, в которой камень приземлился. При этом камень не может отскочить от земли одной и той же провинции более одного раза: в таком случае очки начислятся только за первое касание, а дальше игра завершается. Также игра завершается в том случае, если камень отскочит от земли, находящейся за пределами королевства. Заметьте, что, несмотря на то, что до броска камень формально находится в начале координат, он не касается земли в этой точке.

Джон просит вас помочь ему и определить, какое наибольшее количество очков он может набрать, бросив камень наилучшим образом.

Формат входных данных

В первой строке заданы два целых положительных числа — число провинций n и длина одной провинции на карте Флатландского королевства d ($1 \leqslant n \cdot d \leqslant 2 \cdot 10^5$). Вторая строка содержит n целых чисел — коэффициенты благосостояния ($0 \leqslant w_i \leqslant 10^9$).

Формат выходных данных

Выведите одно число — максимальное число очков, которое Джон может набрать в своей игре.

Примеры

стандартный ввод	*стандартный вывод*
3 5 5 6 6	12
5 20000 1 2 3 4 5	12

Пояснение к примеру

В первом примере выгодно пойти путём $0 \to 8 \to 12$ или $0 \to 9 \to 12$.