

## Числовий багатокутник (*Npolygon*)

Якось до Дім Діміча в гості завітав його друг Васька. Вони разом готувалися до олімпіади з усного рахунку, а тому постійно додавали всі числа, які тільки бачили. Сімка вирішила допомогти друзям та намалювала  $N$ -кутник, до кожної з вершин якого приписала число. Васька, побачивши зображення, запропонував додавати не всі числа, а тільки  $k$  ( $1 < k < N$ ), що йдуть підряд за годинниковою стрілкою. Дім Діміч запропонував не всім вершинам багатокутника приписувати одне й те саме число. Друзі з подивом помітили, що при цьому іноді суми отримуються однаковими для всіх вершин. Вони позначили  $S(i,k)$  як суму чисел в  $k$  послідовних вершинах, починаючи з вершини  $i$ . Друзі вирішили знайти максимальне  $k$ , при якому всі такі суми рівні.

### Формат введення-виведення:

Програма *Npolygon* зчитує з клавіатури (стандартного пристрою введення) єдине ціле число  $N$  кількість кутів  $N$ -кутника ( $3 \leq N \leq 10^{15}$ ).

Програма *Npolygon* виводить на екран (стандартний пристрій виведення) єдине число  $k$  ( $1 < k < N$ ) – максимальну кількість чисел, що розташовані підряд за годинниковою стрілкою, серед яких хоча б два різних, таких, що для всіх  $i$   $S(i,k)$  рівні між собою. Якщо таку послідовність чисел у заданого  $N$ -кутника написати немає можливості, виведіть «-1» (без лапок).

### Приклад вхідних та вихідних даних

Введення	Виведення
6	4
9	6

## Кондитерська фабрика (*fixcandy*)

Як і всі діти, Дім Діміч великий ласун, а тому, коли його спитали, де він хоче працювати, коли виросте, він, очевидно, відповів – виробляти цукерки. Коли ж фіксіки поцікавились, чи уявляє він, у чому буде полягати його робота, він відповів, що це не важливо, аби можна було з'їсти цукерок скільки завгодно. Сімка вирішила продемонструвати Дім Дімічу роботу на фабриці по упаковці вже готових цукерок.

Виконує упаковку фасувально-пакувальний автомат, який має  $N$  резервуарів, у кожен з яких кожен секунду потрапляє рівно одна цукерка. Цукерки пакуються рівно по  $k$  штук у коробці, а тому, як тільки в одному з резервуарів трапляється вказана кількість цукерок, вони відправляються на упаковку і резервуар стає пустим. Далі заповнення резервуару починається спочатку.

Коли Дім Діміч з'явився на фабриці, в кожному резервуарі автомата знаходилась деяка кількість цукерок. Йому доручили виготовити не менше  $L$  коробок по  $k$  цукерок, а після виконання завдання звільнити всі резервуари від цукерок, що залишилися. Сімка, почувши це, жажнулася, прекрасно розуміючи, що звільнення резервуарів перетвориться на їх поживу. А це

означало, що хворі зуби Дім Дімичу забезпечені.

Поки робота тільки налагоджується, вона звернулася по допомогу до вас: мінімізувати кількість цукерок, що потрібно з'їсти. А для цього потрібно визначити, за який мінімальний час Дім Дімич гарантовано виконає (а може, і перевиконає) завдання, а кількість цукерок у резервуарах залишиться мінімально можливою.

#### Формат введення-виведення:

Програма **fixcandy** спочатку зчитує з клавіатури (стандартного пристрою введення) три цілих числа  $N, K, L$  ( $1 \leq N \leq 10^6$ ,  $1 \leq K \leq 10^9$ ,  $0 \leq L \leq 10^9$ ). Потім зчитується  $N$  невід'ємних цілих чисел  $a_1, a_2, \dots, a_N$  строго менших  $K$  – кількість цукерок у кожному з резервуарів автомата перед початком роботи.

Програма **fixcandy** виводить на екран (стандартний пристрій виведення) одне число – мінімальну тривалість зміни, при якій гарантовано виконано завдання і при якій Дім Дімичу доведеться з'їсти мінімальну кількість цукерок.

#### Приклад вхідних та вихідних даних

Введення	Виведення
3 3 2 1 1 2	2

**Примітка:** В момент часу 0 (початок зміни) у резервуарах знаходиться (1, 1, 2) цукерок.

0 сек: (1, 1, 2)

1 сек: (2, 2, 0) => 1 упаковка

2 сек: (0, 0, 1) => 2 упаковки, всього 3

Таким чином, на другій секунді завдання вже виконано, а сума цукерок, яку потрібно з'їсти, мінімальна.

#### Пирог с фиксиками (*fixpie*)

Сьогодні у Дім Дімича велике свято – матуся пече пиріг. И поки вона хазяює на кухні, Дім Дімич розважається запуском дитячого залізничного поїзда. Все було просто чудово, аж поки не з'явився Нолик. С його сестрою Сімкою та її подругою Шпулею знову трапилася неприємність: дівчатка розважалися на кухні і попали в пиріг, який мама вже засунула в духовку. Що робити? Потрібно ж якось убезпечити своїх домашніх від ковтання гвинтиків, у які перетворюються фіксики! А для того, звичайно, потрібно з'їсти побільше пирога.

Аби легше було ділити круглий пиріг між бажаними поласувати, мама на пиріг нанесла  $N$  ( $N \leq 360$ ) радіальних міток для розрізу під кутами  $0.0 \leq a_1 < a_2 < \dots < a_N < 360.0$ . Бажаючих спробувати пиріг рівно  $K$  ( $2 \leq K \leq N$ ). Аби навіть у найгіршому випадку Дім Дімич зміг з'їсти шматок побільше, йому потрібно розрізати пиріг таким чином, аби розмір самого маленького з  $k$  шматків був як можна більшим. Допоможіть Дім Дімичу знайти кутовий розмір  $L$

найменшого шматка з точністю до  $\varepsilon = 10^{-5}$ .

#### Формат введення-виведення:

Програма *fixpie* читає з клавіатури (стандартного пристрою введення) два цілих числа  $N$  і  $k$  – кількість відміток на пирогу і кількість шматків, які треба отримати. Далі зчитується  $N$  чисел  $a_1, a_2, \dots, a_N$  – кути, під якими нанесено радіальні мітки.

Програма *fixpie* виводить на екран (стандартний пристрій виведення) одне число – відповідь на поставлене питання.

#### Приклад вхідних и вихідних даних:

Введення	Виведення
5 3	75.0
25.5 70.0 100.5 170.0 220.5	

**Примітка:** В цьому прикладі пиріг розрізали під кутами 25.5, 100.5 і 220.5, отримавши три шматка з кутковими розмірами 75.0 (шуканий шматок), 120.0 і 165.0.

#### Маніпулятор (*manipul*)

Після непорозуміння, що сталося з маніпулятором, яке налякало секретаря Лізоньку, професор Геній Євгенович вирішив його удосконалити і підключив до комп'ютера. Для управління маніпулятором він написав програму, яка представляла собою величезну таблицю  $A$  розміром  $N \times N$ , що складається з нулів і одиниць, причому одиниці стояли тільки в тих комірках таблиці, у яких залишок від ділення номера рядка і номера стовпчика на деяке число  $M$  був однаковим, тобто  $A[\text{row}][\text{column}] = 1$  тільки якщо

$$\text{row} \% M = \text{column} \% M.$$

Вночі в лабораторію Генія Євгеновича заглянули фіксіки і випадково пересунули на його столі магніт так, що він зіпсував практично всю інформацію на вінчестері. Дивом уціліла рівно  $P-1$  одиниця в програмі для маніпулятора, всі інші комірки обнулились. Сімка страшенно злякалася і спробувала відновити програму. Однак їй вдалося домогтися появи всього однієї додаткової одиниці в таблиці, причому на довільному місці. Оскільки програма була безнадійно зіпсована, професор запропонував вам знайти максимальне з усіх можливих значень  $M$  для вихідної таблиці.

#### Формат ведення-виведення:

Програма *manipul* спочатку читає з клавіатури (стандартного пристрою введення) два цілих числа  $N$  і  $P$  ( $2 \leq N \leq 10^5$ ;  $2 \leq P \leq \min(10^5, N^2)$ ). Потім зчитується  $P$  пар цілих невід'ємних чисел, що не перевищують значення  $N-1$  - позиції збережених одиниць. Правда, невідомо, яка з цих одиниць виникла як побічний ефект після маніпуляцій Сімки.

Програма *manipul* виводить на екран (стандартне пристрої виведення) єдине число - максимальне можливе число  $M$  для початкової таблиці. Якщо максимальне значення

однозначно не визначається, необхідно вивести «-1» (без лапок). Значення М вважається можливим, якщо в програмі для маніпулятора, принаймні на Р-1 позиції, заданих у вхідних даних, буде стояти одиниця.

**Приклад вхідних та вихідних даних:**

Введення	Виведення
3 5 0 0 0 1 1 0 2 2 2 1	1
5 5 0 0 1 1 0 3 1 3 4 4	3
3 2 0 0 0 1	-1

**Пояснення до вхідних тестів:**

У другому тесті таблиця після дії магніту має вигляд: **01010**

**00000**

**00000**

**00001**

Для такої таблиці можливі значення М=2 и М=3. При М=3 програма має такий вигляд:

**10010**

**01001**

**00100**

**10010**

**01001**

Тобто одиниця в позиції 00 (1,3) була побічним ефектом.

В третьому тесті таблиця після дії мала вигляд:

**110**

**000**

**000**

Для такої таблиці підходять всі цілі числа, тому відповідь «-1». Для М>1 вважається, що друга одиниця в першому рядку – побічний ефект.

## Числовой многоугольник (*Npolygon*)

Как-то раз к Дим Димычу в гости пожаловал его друг Васька. Они вместе готовились к олимпиаде по устному счету и поэтому постоянно складывали все числа, которые попадались им на глаза. Симка решила помочь друзьям и нарисовала  $N$ -угольник, к каждой вершине которого приписала число. Васька, увидев изображение, предложил складывать не все числа, а только  $k$  ( $1 < k < N$ ), идущих подряд по часовой стрелке. Дим Димыч предложил не всем вершинам многоугольника приписывать одно и то же число. Друзья с удивлением заметили, что при этом иногда суммы получаются одинаковыми для всех вершин. Они обозначили  $S(i,k)$  как сумму чисел в  $k$  последовательных вершинах, начиная с вершины  $i$ . Друзья решили найти максимальное  $k$ , при котором все такие суммы равны.

### Формат ввода-вывода:

Программа *Npolygon* читает с клавиатуры (стандартного устройства ввода) единственное целое число  $N$  количество углов  $N$ -угольника ( $3 \leq N \leq 10^{15}$ ).

Программа *Npolygon* выводит на экран (стандартное устройства вывода) единственное число  $k$  ( $1 < k < N$ ) – максимальное количество подряд идущих по часовой стрелке чисел, среди которых хотя бы два разных, таких, что для всех  $i$   $S(i,k)$  равны между собой. Если такую последовательность чисел у заданного  $N$ -угольника написать нельзя, выведите «-1» (без кавычек).

### Пример входных и выходных данных:

Ввод	Вывод
6	4
9	6

## Кондитерская фабрика (*fixcandy*)

Как и все дети, Дим Димыч большой сладкоежка, а поэтому, когда его спросили, где он хочет работать, когда вырастет, он, естественно, ответил – изготавливать конфеты. Когда же фиксик поинтересовались, представляет ли он, в чем будет заключаться его работа, он ответил, что это не важно, лишь бы можно было есть конфет сколько угодно. Симка решила показать Дим Димычу работу на фабрике по упаковке уже готовых конфет.

Выполняет упаковку фасовочно-упаковочный автомат, который имеет  $N$  емкостей, в каждую из которых каждую секунду поступает ровно одна конфета. Конфеты пакуются ровно по  $k$  штук в коробке, а потому, как только в одной из емкостей оказывается указанное количество конфет, они отправляются на упаковку и емкость опустошается. Далее заполнение емкости начинается сначала.

Когда Дим Димыч появился на фабрике, в каждой из емкостей автомата находилось некоторое количество конфет. Ему поручили упаковать не менее  $L$  коробок по  $k$  конфет, а после выполнения задания освободить все емкости от оставшихся конфет. Симка, услышав

это, пришла в ужас, прекрасно понимая, что освобождение емкостей превратится в их съедение. А это означало, что больные зубы Дим Димычу обеспечены.

Пока работа только налаживается, она обратилась за помощью к вам: минимизировать количество конфет, подлежащих съедению. А для этого надо определить, за какое минимальное время Дим Димыч гарантированно выполнит (а может, и перевыполнит) задание, а количество конфет в емкостях останется минимально возможным.

#### Формат ввода-вывода:

Программа **fixcandy** сначала читает с клавиатуры (стандартного устройства ввода) три целых числа  $N, K, L$  ( $1 \leq N \leq 10^6$ ,  $1 \leq K \leq 10^9$ ,  $0 \leq L \leq 10^9$ ). Затем считывается  $N$  неотрицательных целых чисел  $a_1, a_2, \dots, a_N$  строго меньших  $K$  – количество конфет в каждой из емкостей автомата перед началом работы.

Программа **fixcandy** выводит на экран (стандартное устройства вывода) одно число – минимальную продолжительность смены, при которой гарантированно выполнен план и при которой Дим Димычу придется съесть минимальное количество конфет.

#### Пример входных и выходных данных:

Ввод	Вывод
3 3 2	
1 1 2	2

**Примечание:** В момент времени 0 (начало смены) в емкостях находится (1, 1, 2) конфет.

0 сек: (1, 1, 2)

1 сек: (2, 2, 0) => 1 упаковка

2 сек: (0, 0, 1) => 2 упаковки, итого уже 3

Таким образом, на второй секунде задание уже выполнено, а сумма конфет, подлежащих съедению минимальна.

#### Пирог с фиксиками (*fixpie*)

Сегодня у Дим Димыча большой праздник – мама печет пирог. И пока она колдует на кухне, Дим Димыч развлекается запуском детского железнодорожного состава. Все было просто замечательно, пока не появился Нолик. С его сестрой Симкой и ее подружкой Шпулей опять случилась неприятность: девчонки баловались на кухне и попали в пирог, который мама уже засунула в духовку. Что делать? Надо как-то обезопасить своих домочадцев от проглатывания винтиков, в которые превращаются фиксики! А для этого, естественно, надо съесть побольше пирога.

Чтобы легче было делить круглый пирог между желающими полакомиться, мама на него нанесла  $N$  ( $N \leq 360$ ) радиальных отметок для разреза под углами  $0.0 \leq a_1 < a_2 < \dots < a_N < 360.0$ .

Желающих попробовать пирог ровно  $K$  ( $2 \leq K \leq N$ ) человек. Теперь, чтобы даже в худшем случае Дим Димыч мог съесть кусок побольше, ему надо разрезать пирог таким образом, чтобы размер самого маленького из  $k$  кусков был как можно больше. Помогите Дим Димычу найти угловой размер  $L$  наименьшего куска с точностью до  $\varepsilon = 10^{-5}$ .

#### Формат ввода-вывода:

Программа *fixpie* сначала читает с клавиатуры (стандартного устройства ввода) два целых числа  $N$  и  $k$  – количество отметок на пироге и количество кусков, которые надо получить. Затем считывается  $N$  чисел  $a_1, a_2, \dots, a_N$  – углы, под которыми нанесены радиальные отметки.

Программа *fixpie* выводит на экран (стандартное устройства вывода) выводит одно число – ответ на поставленный вопрос.

#### Пример входных и выходных данных:

Ввод	Вывод
5 3	75.0
25.5 70.0 100.5 170.0 220.5	

**Примечание:** В этом примере, пирог разрезали под углами 25.5, 100.5 и 220.5, получив три куска с угловыми размерами 75.0 (искомый кусок), 120.0 и 165.0.

#### Манипулятор (*manipul*)

После недоразумения, случившегося с манипулятором, напугавшим секретаря Лизоньку, профессор Гений Евгеньевич решил его усовершенствовать и подключил к компьютеру. Для управления манипулятором он написал программу, которая представляла собой огромную таблицу  $A$  размером  $N \times N$ , состоящую из нулей и единиц, причем единицы стояли только в тех ячейках таблицы, у которых остаток от деления номера строки и номера столбика на некоторое число  $M$  был одинаковым, т.е.  $A[\text{row}][\text{column}] = 1$  только если  $\text{row} \% M = \text{column} \% M$ .

Ночью в лабораторию Гения Евгеньевича заглянули фиксика и случайно передвинули на его столе магнит так, что он испортил практически всю информацию на винчестере. Чудом уцелело ровно  $P-1$  единица в программе для манипулятора, все остальные ячейки обнулились. Симка страшно испугалась и попыталась восстановить программу. Однако ей удалось добиться появления всего одной дополнительной единички в таблице, причем на произвольном месте.

Поскольку программа была безнадежно испорчена, профессор предложил Вам найти максимальное из всех возможных значений  $M$  для исходной таблицы.

#### Формат ввода-вывода:

Программа *manipul* сначала читает с клавиатуры (стандартного устройства ввода) два целых числа  $N$  и  $P$  ( $2 \leq N \leq 10^5$ ;  $2 \leq P \leq \min(10^5, N^2)$ ). Затем считывается  $P$  пар целых неотрицательных чисел, не превышающих значение  $N-1$  – позиции сохранившихся единичек. Правда, не известно, какая из этих единичек возникла как побочный эффект после манипуляций Симки.

Программа *manipul* выводит на экран (стандартное устройства вывода) единственное число – максимально возможное число  $M$  для первоначальной таблицы. Если максимальное значение однозначно не определяется, необходимо вывести «-1» (без кавычек). Значение  $M$  считается возможным, если в программе для манипулятора, по крайней мере на  $P-1$  позиции, заданных во входных данных, будет стоять единица.

**Пример входных и выходных данных:**

<b>Ввод</b>	<b>Вывод</b>
-------------	--------------

3 5	1
-----	---

0 0 0 1 1 0 2 2 2 1

5 5	3
-----	---

0 0 1 1 0 3 1 3 4 4

3 2	-1
-----	----

0 0 0 1

**Пояснение к входным тестам:**

Во втором тесте таблица после воздействия магнитом имеет вид:

10010

01010

00000

00000

00001

Для такой таблицы возможны значения  $M=2$  и  $M=3$ . При  $M=3$  программа имеет такой вид:

10010

01001

00100

10010

01001



Т.е. единица в позиции (1,3) была побочным эффектом.

В третьем тесте таблица после воздействия имела вид:

**110**

**000**

**000**

Для такой таблицы подходят все целые числа, поэтому ответ «-1». Для  $M > 1$  считается, что вторая единица в первом ряду – побочный эффект.