Задача 1A «Швидке піднесення до степеню (за модулем, до 2^{63})»

Для натуральних чисел a, N, p ($1 \leqslant a < 2^{63}$, $1 \leqslant N < 2^{63}$, $1 \leqslant p < 2^{63}$) виведіть $(a^N) \bmod p$, тобто залишок від цілочисельного ділення a^N на p.

Вхідні дані. В один рядок через пропуски (пробіли) будуть вводитися багато трійок натуральних чисел a, N, p, всере́дині кожної з трійок a, N, p задані в указаному порядку, та взяті з діапазонів, указаних у попередньому абзаці. Кількість трійок ні́як не вказується, слід читати, доки вони не вичерпаються. Гарантовано, що між числами все́редині рядка рівно по одному пропуску (пробілу); після останнього числа́ останньої трійки нема пропуску і є переведення рядка; зразу після переведення рядка вхідні дані закінчуються.

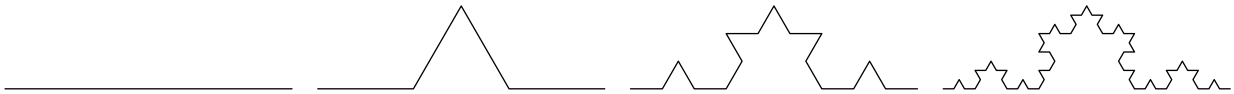
Приклад.

Вхідні дані	Результати
2 10 100 10 2 32	24 4

Результати. Вивести стільки знайдених значень $(a^N) \bmod p$, скільки у вхідних даних було трійок; кожне число виводити в окремому рядку.

Задача 1В «Сніжинка Коха»

Конкретно цю програму пропонується писати, не здаючи у систему автоматичної перевірки.



$n = 0$ $n = 1$ $n = 2$ $n = 3$

Рис. 1. Сніжинки Коха порядків 0, 1, 2 та 3.

За формальне означення самоподібної ламаної «сніжинка Коха» можна взяти будь-яке з таких:

А) Сніжинка Коха нульовóго порядку є відрізком, а будь-який n-й порядок (де n — ціле строго додатне) можна отримати зі сніжинки Коха $(n-1)$-го порядку заміною кожного відрізка на ламану, за такими правилами: відрізок ділиться на три рівні частини, на середній будують рівносторонній трикутник і замнюють основу трикутника на решту дві сторони.

Б) Сніжинка Коха нульовóго порядку є відрізком, а будь-який n-й порядок (де n — ціле строго додатне) складається з чотирьох сніжинок Коха $(n-1)$-го порядку втричі менших лінійних розмірів, причому дрýга частина повернута відносно першої на 60° ліворуч, третя відносно дрýгої — на 120° праворуч, а четверта орієнтована так, як перша (тобто повернута відносно третьої на 60° ліворуч).

Напишіть програму, що зображає n-й порядок сніжинки Коха. Рекомендується писати програму мовою Python з використанням «черепашої графіки», тобто виконавця, який вміє виконувати команди «проїхати вперед на таку-то відстань», «повернути ліворуч на стільки-то градусів», «повернути праворуч на стільки-то градусів» (і ще деякі, але вони зáраз не важливі).

Задача 1C «Сніжинка Коха–2»

Означення самоподібної ламаної «Сніжинка Коха» див. у попередній задачі. Якщо говорити не про текст програми для «черепашки», а про хід її застосування, то «черепашка» завжди малює відрізки однаковісінької довжини́, а красивий малюнок формується суто послідовністю поворотів. Більш того, тут є лише два види поворотів: лівий поворот на $60°$ (позначимо його буквою “L”) та правий поворот на $120°$ (позначимо буквою “R”).

Таким чином, сніжинку Коха 1-го порядку можна описати послідовністю поворотів “LRL” (враховуємо лише повороти *всере́дині* ламаної, тому при чотирьох ланках ламаної поворотів і повинно бути три); аналогічно, сніжинку 2-го порядку можна описати послідовністю “LRLLLRLRLRLLLRL”, 3-го — послідовністю “LRLLLRLRLRLLLRLLLRLLLRLRLRLLLRLRLRLLLRLRLRLLLRLLLRLLLRLRLRLLLRL”, і так далі.

Це, в принципі, можна використати, щоб побудувати ламану, не використовуючи рекурсію (побудувати послідовність таких рядків суто ітеративними, тобто циклічними, засобами, потім виконати послідовність поворотів). Але за́раз розглянемо зворотню задачу: слід, не будуючи цих рядків у пам'яті, вивести підрядок (неперервний фрагмент) такого рядка. Наприклад, вище наведено, який вигляд мають рядки, відповідні 2-му та 3-му порядкам. І можна поставити задачу: вивести увесь підрядок (неперервний фрагмент), починаючи з символу № i та кількістю символів k. Наприклад, при $i=3$, $k=10$ це буде “LLLRLRLRLL” (вважаючи, що нумерація символів починається з 1); який при цьому бра́ти порядок ламаної, насправді несуттєво (аби досить великий), бо рядок кожного подальшого порядку починається в точності з рядка попереднього порядку.

Вхідні дані. Єдиний рядок вхідних даних містить два числа́ i $(1 \leqslant i \leqslant 10^{18})$ та k $(1 \leqslant k \leqslant 10^5)$, розділені одинарним пробілом.

Результати. Підрядок (неперервний фрагмент), починаючи з символу № i та кількістю символів k (вважаючи, що символи рядка занумеровані з 1, і що розглядається досить великий порядок ламаної, щоб символи з такими номерами існували).

Приклади.

Вхідні дані	Результати
3 10	LLLRLRLRLL
8 8	RLRLLLRL

Задача 1D «Драконова ламана»

Конкретно цю програму пропонується писати, не здаючи у систему автоматичної перевірки.

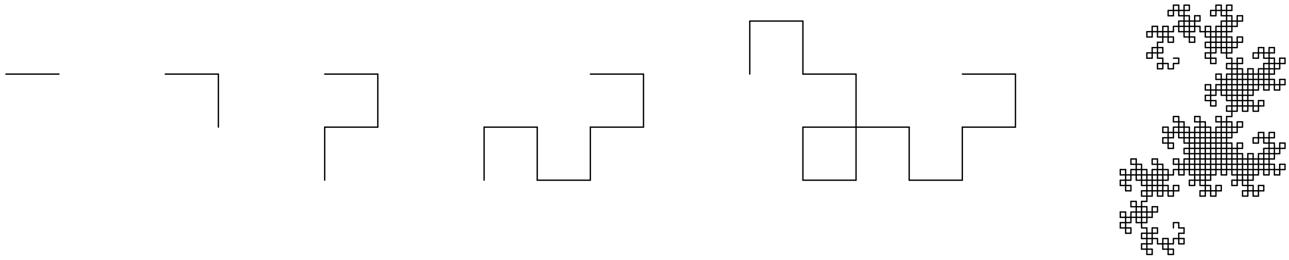


Рис. 2. Драконові ламані порядків 0, 1, 2, 3, 4 та, у дрібнішому масштабі, 10.

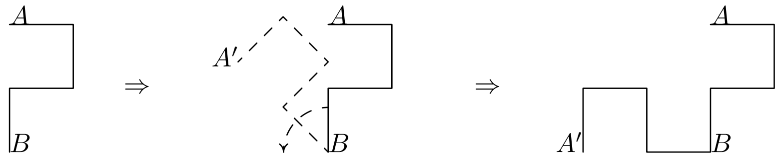


Рис. 3. Процес побудови драконової ламаної n-го порядку з $(n{-}1)$-го.

Самоподібну «драконову ламану» можна означити так:

Драконова ламана 0-го порядку є відрізком. Драконову ламану n-го $(n \geqslant 1)$ порядку можна отримати з драконової ламаної $(n-1)$-го порядку так: «розщепити» лінію на дві, й одну її частину залишити на місці, а іншу повернути відносно хвоста ламаної $(n-1)$-го порядку на 90° проти годинникової стрілки.

Напишіть програму, що зображає n-й порядок драконової ламаної. Рекомендується писати програму мовою Python з використанням «черепашої графіки», тобто виконавця, який вміє виконувати команди «проїхати вперед на таку-то відстань», «повернути ліворуч на стільки-то градусів», «повернути праворуч на стільки-то градусів» (і ще деякі, але вони зáраз не важливі).

Задача 1E «Драконова ламана–2»

Означення самоподібної «драконової ламаної» див. у попередній задачі. Якщо говорити не про текст програми для «черепашки», а про хід її застосування, то «черепашка» завжди малює відрізки однаковісінької довжини́, а красивий малюнок формується суто послідовністю поворотів. Більш того, тут є лише два види поворотів: лівий (позначимо його буквою "L") та правий (позначимо "R"), обидва на $90°$.

Таким чином, драконову ламану 1-го порядку можна описати єдиним поворотом "R" (враховуємо лише повороти *всере́дині* ламаної); аналогічно, ламану 2-го порядку можна описати послідовністю "RRL", 3-го — послідовністю "RRLRRLL", 4-го — "RRLRRLLRRRLLRLL", і так далі.

Це, в принципі, можна використати, щоб побудувати ламану, не використовуючи рекурсію (побудувати послідовність таких рядків суто ітеративними, тобто циклічними, засобами, потім виконати послідовність поворотів). Але за́раз розглянемо зворотню задачу: слід, не будуючи цих рядків у пам'яті, вивести підрядок (неперервний фрагмент) такого рядка. Наприклад, вище наведено, який вигляд мають рядки, відповідні 2-му, 3-му та 4-му порядкам. І можна поставити задачу: вивести увесь підрядок (неперервний фрагмент), починаючи з символу №i та кількістю символів k. Наприклад, при $i=3$, $k=10$ це буде "LRRLLRRRLL" (вважаючи, що нумерація символів починається з 1); який при цьому бра́ти порядок ламаної, насправді несуттєво (аби досить великий), бо рядок кожного подальшого порядку починається в точності з рядка попереднього порядку.

Вхідні дані. Єдиний рядок вхідних даних містить два числа́ i $(1 \leqslant i \leqslant 10^{18})$ та k $(1 \leqslant k \leqslant 10^5)$, розділені одинарним пробілом.

Результати. Підрядок (неперервний фрагмент), починаючи з символу №i та кількістю символів k (вважаючи, що символи рядка занумеровані з 1, і що розглядається досить великий порядок ламаної, щоб символи з такими номерами існували).

Приклади.

Вхідні дані	Результати
3 10	LRRLLRRRLL
8 8	RRRLLRLL

Задача 1F «Сума кубів»

Напишіть програму, що читатиме натуральне число N й подаватиме його у вигляді мінімальної кількості точних натуральних кубів. Інакше кажучи, програма повинна знайти такі натуральні m_1, m_2, $\ldots$, m_k, що $m_1^3 + m_2^3 + \ldots + m_k^3 = N$, а k при цьому якнайменше можливе.

Вхідні дані. Єдиний рядок вхідних даних містить єдине число N $(1 \leqslant N \leqslant 44\,777\,444)$.

Результати. Слід вивести рівно два рядки. Перший має містити k — знайдену мінімальну кількість натуральних кубів. Дру́гий рядок повинен містити такі k натуральних чисел, що сума їхніх кубів дорівнює N.

Приклад.

Вхідні дані	Результати
42	7 2 2 2 2 2 1 1
43	3 3 2 2

Задача 1G «MaxSum (щаслива)»

Є прямокутна таблиця розміром N рядків на M стовпчиків. У кожній клітинці записане невід'ємне ціле число. По ній потрібно пройти згори донизу, починаючи з будь-якої клітинки верхнього рядка, далі переходячи щоразу в одну з «нижньо-сусідніх» і закінчити маршрут у якій-небудь клітинці нижнього рядка. «Нижньо-сусідня» означає, що з клітинки (i, j) можна перейти в $(i+1, j-1)$, або в $(i+1, j)$, або в $(i+1, j+1)$, але не виходячи за межі таблиці (при $j = 1$ перший з наведених варіантів стає неможливим, а при $j = M$ — останній).

Напишіть програму, яка знаходитиме максимально можливу *щасливу* суму значень пройдених клітинок серед усіх допустимих шляхів. Як широко відомо у вузьких колах, щасливими є ті й тільки ті чи́сла, десятковий запис яких містить лише цифри 4 та/або 7 (можна обидві, можна лише якусь одну; але ніяких інших цифр використовувати не можна). Зверніть увагу, що щасливою повинна бути са́ме сума, а обмежень щодо окремих доданків нема.

Вхідні дані. У першому рядку записані N та M — кількість рядків і кількість стовпчиків ($1 \leqslant N, M \leqslant 12$); далі у кожному з наступних N рядків записано рівно по M розділених пробілами невід'ємних цілих чисел, кожне не більш ніж з 12 десяткових цифр — значення клітинок таблиці.

Результати. Вивести або єдине ціле число (знайдену максимальну серед щасливих сум за маршрутами зазначеного вигляду), або рядок «`impossible`» (без лапок, маленькими латинськими буквами). Рядок «`impossible`» має виводитися тільки у разі, коли жоден з допустимих маршрутів не має щасливої суми.

Приклад.

Вхідні дані	Результати
3 4 3 0 10 10 5 0 7 4 4 10 5 4	7

Примітки. Взагалі-то максимально можливою сумою є $27 = 10 + 7 + 10$, але число 27 не є щасливим. Тому відповіддю буде максимальна серед щасливих сума $7 = 3 + 0 + 4$, яка досягається уздовж маршруту `a[1][1]` $\rightarrow$ `a[2][2]` $\rightarrow$ `a[3][1]`.

Наскільки відомо автору задачі, автором «широко відомого у вузьких колах» са́ме такого трактування поняття «щасливого числа» є Василь Білецький, випускник Львівського національного університету імені Івана Франка, котрий був капітаном першої з українських команд, що вибороли золоту медаль на фіналі першості світу ACM ICPC, і тривалий час входив у десятку найсильніших спортивних програмістів світу за рейтингом TopCoder.

Задача 1H «За́мок»

Стародавній за́мок має прямокутну форму. Замок містить щонайменше дві кімнати. Підлогу замка можна умовно поділити на $M \times N$ клітин. Кожна така клітинка містить «0» або «1», які задають порожні ділянки та стіни замку відповідно.

Напишіть програму, яка б знаходила кількість кімнат у замку та площу найбільшої кімнати (яка вимірюється кількістю клітинок).

Вхідні дані. План замку задається у вигляді послідовності чисел, по одному числу, яке характеризує кожну клітинку. Перший рядок містить два цілих числа M та N — кількість рядків та кількість стовпчиків ($3 \leqslant M \leqslant 1000$, $3 \leqslant N \leqslant 1000$). M наступних рядків містить по N нулів або одиниць, що йдуть поспіль (без пробілів). Перший та останній рядок, а також перший та останній стовпчик формують зовнішні стіни замку і складаються лише з одиниць.

Результати. Виведіть кількість кімнат та площу найбільшої кімнати замку (по одному числу в рядку).

Приклади.

Вхідні дані	Результати
6 8 11111111 10011001 10011001 11111001 10101001 11111111	4 8

Вхідні дані	Результати
9 12 111111111111 101001000001 111001011111 100101000001 100011111101 100001000101 111111010101 100000010001 111111111111	4 28

Оцінювання. Значна частина тестів містить план замку з кімнатами лише прямокутної форми, але повні (чи навіть близькі до повних) бали може отримати лише програма, яка правильно враховує, що «кімнати» можуть бути як завгодно «закручені». Не менше половини тестів такі, що $3 \leqslant M \leqslant 20$, $3 \leqslant N \leqslant 50$, але повні (чи навіть близькі до повних) бали може отримати лише програма, яка виходить з повних обмежень M, $N \leqslant 1000$.

Задача 1I «Об'єм об'єднання паралелепіпедів»

Дано N $(2 \leqslant N \leqslant 100)$ прямокутних паралелепіпедів, ребра яких паралельні вісям координат. Про ці паралелепіпеди достеменно відомо, що вони частенько перетинаються по 2–3, але лише $K \leqslant \min(15, N)$ з усіх N паралелепіпедів можуть брати участь у тих перетинах, де одна й та сама частина простору ненульового об'єму належить більш, ніж трьом паралелепіпедам одночасно. Чому дорівнює об'єм об'єднання всіх цих паралелепіпедів? (Зі стандартного означення об'єднання випливає, що об'єми частин простору, належних відразу кільком паралелепіпедам, рахуються рівно один раз, незалежно від того, у скільки паралелепіпедів вони входять.)

Вхідні дані. Перший рядок містить натуральне число N $(2 \leqslant N \leqslant 100)$. Далі йдуть N рядків по 6 цілих чисел — спочатку координати якої-небудь вершини чергового паралелепіпеда, потім координати протилежної його вершини. Всі значення координат не перевищують за модулем (абсолютною величиною) мільйон. Паралелепіпеди задовольняють всім вищеописаним обмеженням.

Приклад.

Вхідні дані	Результати
3 0 0 0 10 10 10 19 19 19 9 9 9 20 30 20 30 20 30	2999

Результати. Виведіть єдине ціле число — шуканий об'єм об'єднання.

Примітка. Властивість «лише $K \leqslant \min(15, N)$ з усіх N паралелепіпедів можуть брати участь у тих перетинах, де …» *сильно* впливає на вибір найдоцільнішого способу розв'язання; цілком аналогічну задачу, де така вимога відсутня, слід було б розв'язувати зовсім інакше.

Задача 1J «Єгипетські дроби»

Математики стародавнього Єгипту не знали дробів у сучасному розумінні, але вміли подавати нецілі числа як суму дробів вигляду $1/k$, причому всі k в цій сумі мали бути різними. Наприклад, сучасне поняття $2/5$ виражалося як «одна третя та одна п'ятнадцята» (справді, $1/3+1/15=\frac{5}{15}+\frac{1}{15}=\frac{6}{15}=2/5$).

Математики Нового часу довели, що будь-який правильний звичайний дріб можна подати у єгипетському поданні (як суму дробів вигляду $1/k$), причому це подання не єдине. Напишіть програму, яка перетворюватиме правильний звичайний дріб до єгипетського подання із мінімальною кількістю доданків.

Вхідні дані. Єдиний рядок містить два натуральні числа n та m $(1 \leqslant n < m \leqslant 1000)$ — чисельник та знаменник правильного звичайного дробу.

Результати. Виведіть в один рядок, розділяючи пропусками, сукупність знаменників у єгипетському поданні цього дробу. Всі ці знаменники повинні бути різними, а їхня кількість повинна бути мінімальною.

Приклади.

Або клавіатура, або input.txt	Або екран, або output.txt
2 5	3 15
732 733	2 5 8 9 16 65970 105552

Примітки. У цій задачі, для деяких вхідних даних, можуть бути різні правильні відповіді (з однаковою мінімальною кількістю доданків). Ваша програма повинна знайти будь-яку одну.

Як видно з прикладів, значення знаменників відповіді можуть бути значно більшими за значення чисел у вхідних даних. Автор задачі гарантує, що для всіх дозволених умовою вхідних даних існують такі правильні відповіді, що при їх знаходженні «довга» арифметика не потрібна (тобто, існують правильні відповіді, для яких і кінцеві значення знаменників, і всі проміжні значення правильно організованих обчислень не виходять за межі стандартних цілих типів).

При перевірці вимагатиметься, щоб значення кожного окремо зі знайдених знаменників не перевищувало $10^{18}-1$; добуток знайдених знаменників, якщо учаснику так зручно, може й перевищувати.