

Задача А. «MaxSum (стрибки у будь-який стовпчик)»

Вхідні дані: Або клавіатура, або input.txt Обмеження часу: 1 сек
Результати: Або екран, або output.txt Обмеження пам'яті: 64 мегабайти

Є прямокутна таблиця розміром N рядків на M стовпчиків. У кожній клітинці записане ціле число. По ній потрібно пройти згори донизу, починаючи з будь-якої клітинки верхнього рядка, далі переходячи щоразу в *будь-яку* клітинку наступного рядка, і закінчити маршрут у якій-небудь клітинці нижнього рядка.

Напишіть програму, яка знаходитиме максимально можливу суму значень пройдених клітинок серед усіх допустимих шляхів.

Вхідні дані. У першому рядку записані N та M — кількість рядків і кількість стовпчиків ($1 \leq N, M \leq 200$); далі у кожному з наступних N рядків записано рівно по M розділених пробілами цілих чисел (модуль кожного не перевищує 10^6) — значення клітинок таблиці.

Результати. Вивести єдине ціле число — максимально можливу суму за маршрутами зазначеного вигляду.

Приклади:

Або клавіатура, або input.txt	Або екран, або output.txt
4 3 1 15 2 9 7 5 9 2 4 6 9 -1	42
3 3 1 1 100 1 1 10 100 1 1	210

Примітки. У першому тесті трапився такий збіг обставин, що саме для цього тесту, хоч і можна переходити в *будь-яку* клітинку наступного рядка (дозволені стрибки з 1-го стовпчика у 3-ій), це не дає можливості збільшити цільове значення у порівнянні з «базовим» варіантом задачі. У другому ж тесті цю можливість варто використати й отримати 210 як $100 + 10 + 100$.

Задача В. «Комп'ютерна гра (платформи) — необмежена довжина стрибку»

Вхідні дані: Або клавіатура, або input.txt Обмеження часу: 1 сек
Результати: Або екран, або output.txt Обмеження пам'яті: 64 мегабайти

У старих іграх можна зіткнутися з такою ситуацією. Герой стрибає по платформах, які висять у повітрі. Він повинен перебратися від одного краю екрана до іншого. Гравець може стрибнути з будь-якої платформи i на будь-яку платформу k , затративши при цьому $(i - k)^2 \cdot (y_i - y_k)^2$ одиниць енергії, де y_i та y_k — висоти, на яких розташовані ці платформи. Ясна річ, енергію слід витратити якнайекономніше.

Відомі висоти платформ є порядку від лівого краю до правого. Знайдіть мінімальну кількість енергії, достатню, щоб дістатися з 1-ої платформи до n -ої (останньої).

Вхідні дані. Перший рядок містить кількість платформ n ($1 \leq n \leq 4000$), другий — N цілих чисел, значення яких не перевищують за модулем 200000 — висоти платформ.

Результати. Виведіть єдине число — мінімальну кількість енергії, яку має витратити гравець на подолання платформ (звісно ж у припущенні, що cheat-коди використовувати не можна).

Приклади:

Або клавіатура, або input.txt	Або екран, або output.txt
4 1 2 3 30	731

Примітка. $731 = 1^2 \cdot 1^2 + 1^2 \cdot 1^2 + 1^2 \cdot 27^2$.

Задача С. «MaxSum (щаслива)»

Вхідні дані: Або клавіатура, або input.txt Обмеження часу: 1 сек
Результати: Або екран, або output.txt Обмеження пам'яті: 64 мегабайти

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 чісла, 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 саме сума, а обмежень щодо окремих доданків нема.

Вхідні дані. У першому рядку записані N та M — кількість рядків і кількість стовпчиків ($1 \leq N, M \leq 12$); 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:

Або клавіатура, або input.txt	Або екран, або output.txt
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» саме 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 D. «Єгипетські дроби»

Вхідні дані: Або клавіатура, або input.txt Обмеження часу: 1 сек
Результати: Або екран, або output.txt Обмеження пам'яті: 64 мегабайти

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 $2/5$ виражалось 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 ($1 \leq n < m \leq 1000$) — 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.

Задача Е. «Розподіл станцій по зонам»

Вхідні дані: Або клавіатура, або input.txt

Обмеження часу: 1 сек

Результати: Або екран, або output.txt

Обмеження пам'яті: 64 мегабайти

Керівництво Дуже Великої Залізниці (ДВЗ) вирішило встановити нову систему оплати за проїзд. ДВЗ являє собою відрізок прямої, на якій послідовно розміщені N станцій. Планується розбити їх на M неперервних зон, що сліду-ють підряд, таким чином, щоб кожна зона містила хоча б одну станцію. Оплату проїзду від станції j до станції k необхідно встановити рівною $1+|z_j-z_k|$, де z_j і z_k — номери зон, яким належать станції j та k відповідно. Відома кількість пасажирів, які відправляються за день з кожної станції на кожную іншу.

Напишіть програму, що визначатиме, яку максимальну денну виручку можна отримати за новою системою при оптимальному розбитті на зони.

Вхідні дані.

Перший рядок містить два цілих числа N та M ($1 \leq M \leq N \leq 1000$). Другий — одне число, що означає кількість пасажирів, які їдуть між станціями 1 та 2. Третій — два числа, що означають кількість пасажирів, які їдуть між станціями 1 та 3 та між станціями 2 та 3, відповідно. І так далі. В N -ому рядку міститься $N-1$ число, i -е з них визначає кількість пасажирів від станції i до станції N .

Кількість пасажирів для кожної пари станцій дано з урахуванням руху в обидві сторони. Всі числа цілі, невід'ємні та не перевищують 10000.

Результати. Програма повинна вивести єдине ціле число — шукану максимальну денну виручку.

Приклад:

Або клавіатура, або input.txt	Або екран, або output.txt
3 2 200 10 20	440

Примітки. Іншими словами, рядки вхідних даних з 2-го по N -й являють собою нижню-ліву половину симетричної матриці пасажиропотоку з нулями по головній діагоналі (де симетрично розміщені елементи не треба додавати, бо кожен з них — вже сума потоків туди й назад). Зокрема, у прикладі по суті задано матрицю

0	200	10
200	0	20
10	20	0

Денну виручку 440 можна отримати, якщо розбити станції на зони як «1-а станція у 1-ій зоні, 2-а та 3-я станції у 2-ій зоні». Тоді ціну $1+1=2$ платитимуть 200+10 пасажирів (які їздять між 1-ю та 2-ю та між 1-ю та 3-ю станціями відповідно), а ціну $1+0=1$ платитимуть 20 пасажирів (які їздять між 2-ю та 3-ю станціями); $(200+10) \times 2 + 20 \times 1 = 440$.

Денної виручки, більшої за 440, досягти неможливо.

Задача F. «Ендшпіль»

Вхідні дані: Або клавіатура, або input.txt Обмеження часу: 1 сек
Результати: Або екран, або output.txt Обмеження пам'яті: 64 мегабайти

Нагадаємо деякі істотні для цієї задачі стандартні правила гри в шахи. Грають два гравці, один грає білими, інший чорними. Гра відбувається на шахівниці, тобто дошці 8×8 , стовпчики позначаються буквами від "a" до "h" зліва направо, рядки — цифрами від 1 до 8 знизу догори. Кожна клітинка дошки або порожня, або містить одну фігуру. Якщо фігура A (не пішак) може походити згідно з правилами у клітинку, зайняту чужою фігурою B , то внаслідок такого ходу фігуру B б'ють, тобто знімають з дошки. Тому про всі клітинки, куди деяка фігура може походити, кажуть, що вони знаходяться «під боєм» цієї фігури.

Королю заборонено ходити у клітинки, які перебувають під боєм будь-якої чужої фігури (чи чужого пішака). Якщо один з гравців зробив такий хід, що король суперника опинився під боєм (це називають «шах»), суперник зобов'язаний відповісти таким ходом, щоб його король вже не був під боєм чужої фігури (чи чужого пішака). Якщо такого ходу не існує, то це називають «мат».

Король може ходити на одну клітинку в будь-якому з 8-ми напрямків (ліворуч, праворуч, вперед, назад, в будь-якому напрямку за будь-якою діагоналлю). Ферзь може ходити в будь-якому з цих самих 8-ми напрямків на будь-яку кількість клітинок, але не перетинаючи клітинок, зайнятих фігурами (в тому числі й пішаками; байдуже, своїми чи чужими).

Нехай на шахівниці знаходиться три фігури: білий король, білий ферзь і чорний король. Зараз хід білих. За яку мінімальну кількість ходів вони гарантовано зможуть поставити мат? Чорні робитимуть усе, допустиме правилами гри, щоб уникнути мату або відтермінувати його.

Вхідні дані. Програма повинна прочитати спочатку кількість тестових блоків $TEST_NUM$ ($1 \leq TEST_NUM \leq 70000$), потім самі блоки. Кожен блок є окремим рядком, у якому записані позначення трьох клітинок, де розміщені білий король, білий ферзь і чорний король. Позначення клітинки складається із записаних разом букви вертикалі і номера горизонталі, позначення клітинок всередині рядка розділені одиничними пропусками (пробілами).

Усі задані позиції гарантовано допустимі з точки зору шахових правил (зокрема, чорний король не під боєм).

Результати. Ваша програма повинна вивести для кожного тесту єдине число — мінімальну кількість ходів. Рахується лише кількість ходів білих (кількість ходів-відповідей чорних не додається).

Приклади:

Або клавіатура, або input.txt	Або екран, або output.txt
2	1
a3 b3 a1	2
a3 e3 b1	