

Задача Душа жадає злету... (*tourist3D*).

Втомилася баба Яга робити капості людям і вирішила відпочити. Як відомо, найкращий відпочинок – це туризм, а тому вона прийняла рішення відправитися у далекі країни: себе показати та на інших подивитися. Літати на ступі їй смертельно набридло і вирішила вона відправитися подорожувати на новому транспортному засобі – душа жадає злету... Крім того, мандрувати одній не цікаво, а тому запропонувала баба Яга подорожувати разом Коливану.

Одного чудового дня осіддали баба Яга та Коливан два ідентичних абсолютно пружних тіла і поскакали вони по плоскій поверхні в однорідному гравітаційному полі, ігноруючи опір повітря та інші дрібниці.

У початковий момент часу 0 їх абсолютно пружні тіла а вперше відірвалися від плоскої поверхні в точках $(x_1, y_1, 0)$ та $(x_2, y_2, 0)$ зі швидкостями (v_{x1}, v_{y1}, v_{z1}) та (v_{x2}, v_{y2}, v_{z2}) . Прискорення вільного падіння g у вибраній бабою Ягою місцевості дорівнює в точності 10. Необхідно знайти мінімальне (за весь час польоту в однорідному гравітаційному полі над плоскою поверхнею) відстань між оседланими бабою Ягою та Коливаном абсолютно пружними тілами.

Формат введення/виведення.

Програма *tourist3D* зчитує з клавіатури (стандартного пристрою введення) числа через пробіл $x_1, y_1, 0, v_{x1}, v_{y1}, v_{z1}, x_2, y_2, 0, v_{x2}, v_{y2}, v_{z2}$ (координати z гарантовано дорівнюють нулю, але все одно задаються у вхідних даних). Всі значення координат та швидкостей являються цілими числами, що не перевищують по модулю 1000. Крім того, точки старту $(x_1, y_1, 0)$ та $(x_2, y_2, 0)$ не рівні між собою та $v_{z1} > 0, v_{z2} > 0$.

Програма *tourist3D* виводить на екран (стандартний пристрій виведення) єдине число – значення мінімальної відстані з точністю до шести знаків після десяткової точки.

Приклад вхідних та вихідних даних.

Введення	Виведення
0 1 0 1 1 5 2 1 0 -1 1 10	1.856953

Задача Квадратне рівняння (*ax2rbxpc*).

Після приємного відпочинку в товаристві баби Яги Коливан вирішив розім'яти мозок математичними обчисленнями, щоб не помилятися у підрахунках ігрових боргів. Найулюбленишим його заняттям на дозвіллі є розв'язання квадратних рівнянь. Проте при розв'язанні їх великої кількості помилитися недовго! Ось і просить вас Коливан написати програму, яка розв'язує квадратні рівняння виду $ax^2+bx+c=0$, якщо відомо, що жоден з коефіцієнтів a, b, c не дорівнює 0 в точності, але деякі з них можуть бути дуже близькі до нуля.

Формат введення/виведення.

Програма *ax2rbxpc* зчитує з клавіатури (стандартного пристрою введення) три дійсних числа a, b та c , причому модуль кожного з них — в діапазоні від 10^{-12} до 10^{+12} . Гарантовано

будуть вводитися тільки такі трійки a, b, c , за яких рівняння має два різних кореня, жоден з яких не дорівнює нулю в точності.

Програма **ax2pbxpc** виводить на екран (стандартний пристрій виведення) два знайдені корені через пробіл (спочатку менший, потім більший). Відповідь буде зарахована, якщо значення обох коренів знайдено з відносною похибкою не гірше 1%.

Приклад вхідних та вихідних даних.

Введення	Виведення
17 2015 22	-118.518 -0.01092
17 2e9 22	-1.17647e8 -1.1e-8

Задача I знову бульбашка (*bubblesort*).

Коливан вирішив терміново зібрати зі всіх своїх боржників борги. Щоб не забути розмір боргу кожного з них, він записав у велику таблицю кількість золотих злитків, які йому винен кожен з боржників. Тепер Коливану необхідно відсортувати цю таблицю, щоб стягувати борги строго у порядку спадання, а жодного алгоритму сортування він не знає. Коливану вдалося викрасти у Київського князя секретний документ, що містить опис алгоритму сортування наступного виду:

для k от 1 до $N-1$

для j от 1 до $N-k$

якщо $A[j] < A[j+1]$

переставити $A[j]$ та $A[j+1]$

Почавши виконувати перестановки за описаним алгоритмом, Коливан зрозумів, що робота виконується дуже повільно, і хоче оцінити необхідний час для її виконання. Допоможіть Коливану оцінити час роботи сортування, підрахував для кожного елемента таблиці кількість операцій перестановки, проведених над ним.

Формат введення/виведення.

Програма **bubblesort** зчитує з клавіатури (стандартного пристрою введення) ціле число N ($1 \leq N \leq 100000$) – кількість чисел, що будуть сортуватися; а потім N цілих чисел – елементи таблиці, що сортується ($1 \leq A[i] \leq 10^9$, всі елементи різні), що розділені пробілами.

Програма **bubblesort** виводить на екран (стандартний пристрій виведення) N чисел, кожне з яких дорівнює кількості операцій перестановки, що були проведені над числом, яке стояло на цьому місці у вихідній таблиці.

Приклад вхідних та вихідних даних.

Введення	Виведення
4 1 2 5 4	3 3 2 2

Задача Розпил (SLICE)

Після численних спроб зекономити та внесення змін у проект будівництва Коливан нарешті отримав басейн кубічної форми з ребром 1 м. Але навіть після таких суттєвих зменшень витрат він все одно не може заспокоїтися і хоче заробити на експлуатації басейну.

Випадково він дізнався, що богатирі збираються проводити змагання скейтбордистів і їм потрібен полігон. Полігоном може слугувати площина, яка проходить через три точки, що знаходяться на поверхні уявного куба, утвореного басейном. Щоб не заплутатись, Коливан зв'язав один з кутів куба з системою декартових координат. Потім, використовуючи усі свої зв'язки, дізнався координати точок на поверхні цього уявного куба. Тепер його мета обчислити площу полігону, який проходить через вказані три точки та обмежений гранями уявного куба, щоб визначити оплату за користування цим полігоном.

Формат введення/виведення.

Програма **SLICE** зчитує з клавіатури (стандартного пристрою введення) три трійки дійсних чисел – координати трьох точок (числа в діапазоні $[0; 1]$), розділені пробілами. Гарантується, що точки не лежать на одній прямій і не належать всі одночасно одній грані куба.

Програма **SLICE** виводить на екран (стандартний пристрій виведення) одне дійсне число – шукану площу. Відповідь потрібно отримати з максимально можливою точністю.

Приклад вхідних та вихідних даних

<i>Введення</i>	<i>Виведення</i>
0.5 1 1 1 0.5 1 1 1 0.5	0.2165063509461097