

Задача А. «Четверта вершина прямокутника»

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

Знаючи координати трьох вершин прямокутника на координатній площині, визначити координати четвертої вершини. Сторони прямокутника не зобов'язані бути паралельні вісям координат. Три вершини не обов'язково задані підряд (пропущена вершина може бути як після них, так і де завгодно всередині).

Вхідні дані. В одному рядку записані шість чисел — координати трьох вершин прямокутника (спочатку x та y однієї вершини, потім x та y іншої, потім x та y ще однієї). Числові значення координат цілі, абсолютна величина (модуль) кожного не перевищує 100.

Результати. Виведіть в одному рядку через пропуск (пробіл) два числа — координати четвертої вершини прямокутника.

Приклади:

Або клавіатура, або input.txt	Або екран, або output.txt
-2 3 4 3 4 -1	-2 -1
-1 2 0 0 6 3	5 5

Задача В. «Трикутник і точка»

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

Задано (координатами вершин) трикутник ABC і точку D . Визначити розміщення точки відносно трикутника, а саме: вивести "In", якщо точка лежить строго всередині трикутника; "Edge", якщо точка лежить на стороні; "Vertex", якщо точка лежить на вершині; "Out", якщо точка лежить поза трикутником.

Вхідні дані. Чотири рядки по 2 цілих числа, що не перевищують по модулю 10 000 000.

Перші три рядки — координати вершин трикутника A, B, C . Четвертий — координати точки O .

Результати. Відповідь до задачі ("In", "Edge", "Vertex" або "Out").

Приклад:

Або клавіатура, або input.txt	Або екран, або output.txt
-2 -2 3 1 0 1 0 0	In

Задача С. «Спільні дотичні–1»

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

Як відомо, дотичною до кола є пряма, яка має рівно одну спільну точку з цим колом. Можлива ситуація, коли одна й та сама пряма є дотичною відразу до двох кіл. Тоді вона називається спільною дотичною. Напишіть програму, яка знаходитиме кількість різних спільних дотичних для заданих двох кіл. При виведенні врахуйте стародавню традицію приписувати числу 7 значення «багато». Тобто, коли кількість спільних дотичних строго більша 6, незалежно від справжньої кількості виводьте 7.

Вхідні дані. Шість цілих чисел, розділених пропусками (пробілами) X_1 , Y_1 , R_1 , X_2 , Y_2 , R_2 — відповідно координати центра і радіуси 1-го і 2-го кола. Для всіх координат, абсолютна величина (модуль) не перевищує мільйон. Для обох радіусів, значення у межах від 1 до мільйона (обидві межі включно).

Результати. Програма виводить єдине число — шукану кількість, з урахуванням згаданої стародавньої традиції.

Приклад:

Або клавіатура, або input.txt	Або екран, або output.txt
20 0 4 50 0 10	4

Задача D. «Спільні дотичні–2»

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

Як відомо, дотичною до кола є пряма, яка має рівно одну спільну точку з цим колом. Можлива ситуація, коли одна й та сама пряма є дотичною відразу до двох кіл. Тоді вона називається спільною дотичною. Напишіть програму, яка знаходитиме спільні дотичні для заданих двох кіл. При виведенні врахуйте стародавню традицію приписувати числу 7 значення «багато». Тобто, коли кількість спільних дотичних строго більша 6, незалежно від справжньої кількості виводьте будь-які сім з усіх можливих спільних дотичних.

Вхідні дані. Шість цілих чисел, розділених пропусками (пробілами) $X_1, Y_1, R_1, X_2, Y_2, R_2$ — відповідно координати центра і радіуси 1-го і 2-го кола. Для всіх координат, абсолютна величина (модуль) не перевищує мільйон. Для обох радіусів, значення у межах від 1 до мільйона (обидві межі включно).

Результати. Програма виводить у першому рядку єдине число K — кількість шуканих спільних дотичних (з урахуванням згаданої стародавньої традиції); далі повинно йти рівно K рядків, кожен з яких повинен містити чотири дійсні числа — координати двох різних точок з відповідної дотичної (спочатку x - та y -координати однієї точки, потім x - та y -координати іншої).

Відповідь зараховуватиметься, коли виконуватимуться всі вимоги:

1. Кількість спільних дотичних знайдено правильно (з урахуванням вказаної стародавньої традиції).
2. Кожна дотична описується двома помітно різними точками (відстань між двома точками, що задають одну дотичну, не менша 1).
3. Пряма, що проходить через кожну з пари точок, або справді є дотичною (має рівно одну спільну точку з колом), або є добрим наближенням до дотичної (або проходить поза колом на відстані не більш як 10^{-6} від нього, або заходить всередину кола так, що довжина частини цієї прямої всередині цього кола не перевищує однієї мільйонної від радіуса цього кола).
4. Кожен з K рядків, що описують дотичні, описує свою власну дотичну, відмінну від інших; формально кажучи, якщо взяти j -й та k -й рядки ($j \neq k$), де дотичні задані точками A_j, B_j та A_k, B_k , то повинно виконуватися

$$((dist(A_j, A_k B_k) \geq 0,1) \text{ or } (dist(B_j, A_k B_k) \geq 0,1))$$

and

$$((dist(A_k, A_j B_j) \geq 0,1) \text{ or } (dist(B_k, A_j B_j) \geq 0,1))$$

де $dist$ — відстань від точки до прямої, що рахується уздовж перпендикуляру; смисл усього виразу разом узятого — хоча б одна з двох точок, які задають пряму, знаходиться на відстані хоча б 0,1 від іншої прямої.

Приклад:

Вхідні дані	Результати
20 0 4 50 0 10	4 48 -9.79795897113 19.2 -3.91918358845 48 9.79795897113 19.2 3.91918358845 45.3333333333 -8.84433277428 21.8666666667 3.53773310971 45.3333333333 8.84433277428 21.8666666667 -3.53773310971

Задача Е. «Годинник на сканері»

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

Секундна стрілка годинника переміщується стрибками, тобто протягом секунди нерухома, а потім дуже швидко повертає на $\frac{1}{60}$ повного оберту. Стрілка являє собою тонкий відрізок довжини d мм, що виходить з центру годинника. Годинник поклали на сканер, орієнтувавши звичайним чином (позначка "12" згори) й підібрали параметри сканування так, що:



1. Сканування запускається відразу після того, як секундна стрілка виконала черговий стрибок і почала показувати s секунд.
2. Область сканування вибрана як квадрат розмірами $2d$ мм \times $2d$ мм, так, що вона в точності охоплює круг, який покриває секундна стрілка.
3. Сканер за (кожну) 1 с встигає отримати прямокутне зображення висотою рівно k мм (та шириною $2d$ мм, тобто в усю область сканування).
4. Роздільча здатність сканера досить висока, щоб можна було знехтувати дискретністю зображень всередині кожної k -міліметрової смужки й рахувати відстані за звичайними геометричними формулами.

Знайдіть сумарну довжину зображень секундної стрілки в отриманій картинці (зображення інших елементів годинника не створюють проблем, бо секундна стрілка має зовсім інший колір).

Вхідні дані. Три цілі числа: k (висота області, яку сканують за 1 с), d (довжина стрілки) та s (скільки секунд почала показувати стрілка у момент початку сканування).

$1 \leq k \leq 50$, $0.75k \leq d \leq 100k$, відношення $\frac{d}{k}$ гарантовано не є цілим числом, $0 \leq s < 60$.

Результати. Єдине дійсне число l — сумарну довжину зображень секундної стрілки.

Відповідь буде зарахована, якщо відносна або абсолютна похибка (хоча б одна з них) не перевищить 10^{-6} .

Приклад:

Або клавіатура, або input.txt	Або екран, або output.txt
36 90 10	103.994544

Задача F. «Відстань від точки до відрізка»

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

Дано точку P з координатами $P_x P_y$ та відрізок AB , кінці якого мають координати $A_x A_y$ та $B_x B_y$. Відрізок гарантовано не вироджений, тобто A та B — різні точки.

Напишіть програму, яка знаходитиме відстань між точкою P та відрізком AB .

Примітка. Відстань між точкою та відрізком слід трактувати згідно зі стандартним означенням відстані між точкою та складним геометричним об'єктом: якщо точка належить цьому об'єкту, відстань рівна нулю; якщо не належить, відстань рівна довжині найкоротшого з можливих відрізків, для яких одним з кінців є дана точка, а інший кінець належить цьому об'єкту.

Вхідні дані. слід прочитати зі стандартного входу (клавіатури), у форматі $P_x P_y A_x A_y B_x B_y$ (в одному рядку). Всі координати цілі й не перевищують по модулю 10000.

Результати. Вивести єдине число — знайдену відстань від точки до відрізка. Виводити можна хоч у експоненційній формі, хоч стандартним десятковим дробом. Результат зараховується, коли похибка (абсолютна або відносна, тобто хоча б одна з них) не перевищує 10^{-6} .

Приклад:

Або клавіатура, або input.txt	Або екран, або output.txt
0 4 2 3 2 5	2.0

Задача G. «Площа простого многокутника»

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

Многокутник на площині задано цілочисельними координатами N вершин. Потрібно знайти його площу.

Многокутник простий, тобто його сторони не перетинаються і не дотикаються (за винятком сусідніх, у вершинах), але він не обов'язково опуклий.

Вхідні дані. слід прочитати зі стандартного входу (клавіатури). У першому рядку задано кількість вершин N ($1 \leq N \leq 50000$). У наступних N рядках записані пари чисел — координати вершин. Сторони многокутника — відрізки між 1-ою і 2-ою, 2-ою і 3-ьою, ..., $(N-1)$ -ою і N -ою, N -ою і 1-ою вершинами. Значення координат — цілі числа, не перевищують по модулю мільйон.

Результати. Вивести єдине число — знайдену площу многокутника. Виводити можна хоч у експоненційній формі, хоч стандартним десятковим дробом. Результат зараховується, коли похибка (абсолютна або відносна, тобто хоча б одна з них) не перевищує 10^{-6} .

Приклад:

Або клавіатура, або input.txt	Або екран, або output.txt
4 0 4 0 0 3 0 1 1	3.5

Задача Н. «Чи є маршрут приємним?»

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

Туристу набридло подорожувати уздовж координатної вісі, тому він вирішив помандрувати ще по координатній площині. Він розпочинає зі своєї бази в точці A_1 з координатами (x_1, y_1) , рухається найкоротшим маршрутом до визначної пам'ятки A_2 з координатами (x_2, y_2) , далі, не зупиняючись, рухається найкоротшим маршрутом до визначної пам'ятки A_3 з координатами (x_3, y_3) , і так далі. Дійшовши до останньої визначної пам'ятки A_n з координатами (x_n, y_n) , він, не зупиняючись, рухається до своєї бази. Турист вважає свій маршрут *неприємним*, якщо існує така пряма, що він уздовж неї не рухався, і разом з тим перетинав її строго більше двох разів. Якщо маршрут не є неприємним, турист вважає його *приємним*.

Турист вважає, що перетинав пряму, якщо в деякий момент часу перебував у одній півплощині відносно неї, а через деякий проміжок часу — в іншій півплощині (сама пряма не належить жодній з півплощин).

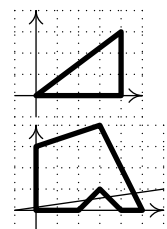
Напишіть програму, яка, прочитавши описи кількох маршрутів, визначить, чи приємний кожен з них.

Вхідні дані. Програма повинна прочитати спочатку кількість маршрутів K ($2 \leq K \leq 12$), потім K однотипних блоків, кожен з яких описує маршрут. Кожен блок опису маршруту починається числом n ($2 \leq n \leq 98765$), далі йдуть n пар цілих чисел, що не перевищують 10^8 за абсолютною величиною — координати $x_1 y_1 x_2 y_2 \dots x_n y_n$. Всі числа всіх маршрутів записані в одному рядку й розділені одинарними пробілами. Сумарна кількість всіх вершин усіх маршрутів, які програма має обробити за один запуск, не перевищуватиме 123456.

Результати. Програма повинна вивести у один рядок K розділених пробілами нулів та/або одиниць, які позначають, приємними (1) чи неприємними (0) були відповідні маршрути.

Приклад:

Вхідні дані	Результати
2 3 0 0 4 0 4 3 7 0 3 0 0 2 0 3 1 4 0 5 0 3 4	1 0



Примітка. Многокутники з прикладу вхідних даних зображені на рисунку. У першому випадку неможливо провести таку пряму, щоб турист уздовж неї не рухався, але перетинав її строго більше двох разів. У другому випадку наведено один з багатьох можливих прикладів саме такої прямої.