**XIV Всеукраїнська олімпіада з інформатики**

# Квадрат

Трикутник задано на площині координатами своїх вершин : (*X*1,*Y*1), (*X*2,*Y*2), (*X*3,*Y*3). Знайти довжину *L* сторони квадрата мінімальної площі, в який можна помістити цей трикутник так, щоб всі вершини трикутника знаходились всередині квадрата або на його сторонах.

**Завдання** Напишіть програму SQUARE яка за координатами вершин трикутника знаходить довжину *L* сторони квадрата мінімальної площі, в який можна помістити цей трикутник. *L* достатньо знайти з точністю 10-4.

**Вхідні дані** Файл SQUARE.DAT містить в одному рядку дійсні числа *X*1 *Y*1 *X*2 *Y*2 *X*3 *Y*3, розділені пропусками, – координати вершин трикутника (-10000 ≤ *X*1, *Y*1, *X*2, *Y*2, *X*3, *Y*3 ≤ 10000).

### Приклад вхідного файлу

0.0 0.0 1.1 0.0 0.0 1.1

**Вихідні дані** Файл SQUARE.SOL повинен містити одне число - довжину *L* сторони шуканого квадрата.

### Приклад вихідного файлу

1.1

**XV Всеукраїнська олімпіада з інформатики**

Багатокутники

На площині задана така множина з *N* багатокутників, що виконуються наступні умови:

1. ніякі два багатокутника не мають спільних точок;
2. для кожного *i*–го багатокутника існує *Pi* багатокутників, всередині яких він знаходиться, і   
   *N*-1-*Pi* багатокутників, котрі знаходяться всередині нього, 0≤*Pi*≤*N*-1.

Завдання

Напишіть програму POLYGON, яка для кожного багатокутника видає кількість багатокутників, всередині яких він знаходиться.

Вхідні дані

Перший рядок вхідного файлу POLYGON.DAT містить ціле число *N* — кількість багатокутників, 3≤*N*≤10000. Наступні *N* рядків файлу описують *N* багатокутників. (*i*+1)–ий рядок файлу описує *i*–ий багатокутник. Перше ціле число *Ci* — кількість вершин багатокутника, 3≤*Ci*≤20. Наступні *Ci* пар чисел — координати вершин багатокутника у порядку його обходу. Координати вершин — цілі числа, що належать діапазону від -2 000 000 000 до 2 000 000 000.

Вихідні дані

Єдиний рядок вихідного файлу POLYGON.SOL повинен містити N чисел: *i*–те число рядка повинно бути *Pi* — кількість багатокутників, всередині яких знаходиться *i*–ий багатокутник.

Приклад вхідних та вихідних даних

|  |  |
| --- | --- |
| POLYGON.DAT | POLYGON.SOL |
| 3  3 -2 1 8 9 12 1  3 7 5 6 3 7 4  4 4 3 7 7 9 3 1 2 | 0 2 1 |

XVII Всеукраїнська олімпіада з інформатики

Перший тур

Вектори (100 балів)

На площині задано множину точок (x, y), де x та y – цілі, 1≤x≤M, 1≤y≤N. З кожної точки виходить рівно один з наступних векторів: (-1,-1), (-1,0), (-1,1), (0,1), (1,1), (1,0), (1,-1), (0,-1). Кожен вектор сполучає одну цілочисельну точку площини з іншою. Наприклад, якщо з точки (2, 5) виходить вектор (1, 1), то він сполучає цю точку з (3, 6), але не навпаки.

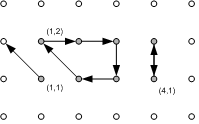
Послідовність з двох і більше точок площини a1, a2,…, ak назвемо циклом, якщо кожна точка ai сполучена вектором з ai+1 (1≤i≤k-1), та ak сполучена вектором з a1. Цикли вважаються різними, якщо вони відрізняються хоча б однією вершиною.

Завдання

Напишіть програму VECTORS, що за інформацією про вектори, що виходять з точок площини, знаходить кількість різних циклів.

Вхідні дані

Перший рядок вхідного файлу VECTORS.DAT містить два цілих числа N (1≤N≤100) та M (1≤M≤100). Кожен з наступних N рядків, містить M пар чисел (тобто, 2⋅M чисел). Пара x, що знаходиться у рядку y задає вектор у точці (x, y).

Вихідні дані

Єдиний рядок вихідного файлу VECTORS.SOL має містити ціле число – кількість циклів, утворених векторами.

Приклад вхідних та вихідних даних

|  |  |
| --- | --- |
| VECTORS.DAT | VECTORS.SOL |
| 2 4  -1 1 -1 1 -1 0 0 1  1 0 1 0 0 -1 0 -1 | 2 |

Зала Круглих Столів (100 балів)

Єдиний спосіб потрапити до Зали Круглих Столів – пройти через Колонний Коридор. Стіни Коридору зображаються на карті прямими лініями, які паралельні вісі OY системи координат. Вхід в коридор знаходиться знизу, а вихід з Коридору до Зали – зверху. В Коридорі є циліндричні (на карті круглі) Колони однакового радіуса R.

Завдання

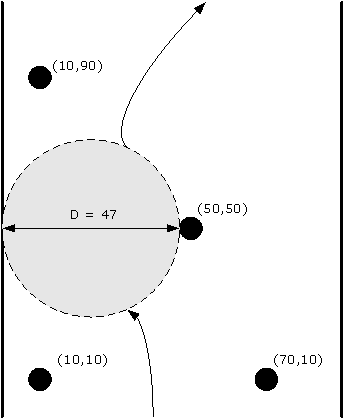
Напишіть програму TABLE, що за інформацією про розміри Коридору, та розміщення Колон визначає діаметр найбільшого з Круглих Столів, який можна пронести через такий Коридор, зберігаючи поверхню Стола горизонтальною.

Вхідні дані

У першому рядку вхідного файлу TABLE.DAT задані два числа XL та XR – x-координати лівої та правої стін Коридору. У другому рядку знаходиться ціле число R (1≤R≤1 000 000) – радіус усіх Колон. У третьому – ціле число N (1≤N≤200), що задає кількість Колон. Далі йдуть N рядків, в кожному з яких по два числа – x- та y-координати центра відповідної Колони.

Всі вхідні координати – цілі числа, що за модулем не перевищують 1 000 000.

Вихідні дані

Єдиний рядок вихідного файлу TABLE.SOL має містити єдине число – шуканий діаметр найбільшого Столу. Діаметр потрібно виводити з точністю 3 знаки після десяткової крапки (навіть у випадку, якщо він виявиться цілим). Якщо не можна пронести жодного Столу, то відповідь має бути: 0.000

Точність 3 знаки після крапки, за звичайними правилами заокруглення, означає, що відповідь, яка видається у вихідний файл, повинна відрізнятися від точної не більше ніж на 5⋅10-4 (тобто на 0.0005). Наприклад, якщо точна відповідь 1.234567, то у файлі повинно знаходитися число 1.235. Якщо точна відповідь 5.0005, то необхідно заокруглювати у більшу сторону, тобто у файл необхідно видати 5.001

Приклад вхідних та вихідних даних

|  |  |
| --- | --- |
| TABLE.DAT | TABLE.SOL |
| 0 90  3  4  10 10  70 10  50 50  10 90 | 47.000 |

XVIII Всеукраїнська олімпіада з інформатики

Театр (100 балів)

У Театрі збираються поставити грандіозну п’єсу з двох актів, в якій освітлення має велике значення. Сцена театру має форму опуклого многокутника, заданого вершинами у декартовій прямокутній системі координат. Над сценою знаходиться прожектор, що може переміщуватися над нею довільним чином. Знаходячись у деякій точці, прожектор освітлює круглу область з центром у цій точці та радіусом R.

У першому акті на сцені лежать квадратні килими розміром HxH, сторони яких паралельні осям координат. Килими можуть частково виходити за межі сцени. Розглянемо фігуру, що складається з усіх точок, знаходячись в яких, прожектор не освітлює жоден з килимів та не освітлює територію поза сценою. Позначимо її площу як *S*1.

Перед другим актом килими прибирають зі сцени. Розглянемо фігуру, яка складається з усіх точок, знаходячись в яких, прожектор не освітлює територію поза сценою. Її площу позначимо як *S*2.

Завдання

За наданими вхідними файлами, кожен з яких описує сцену та розташування на ній килимів у першому акті, створіть відповідні їм вихідні файли, що містять площі *S*1 та *S*2 описаних вище фігур.

Вхідні дані

На вашому диску в каталозі DATA містяться 10 файлів, що мають назви THEATER.D01, THEATER.D02, … , THEATER.D10, такого формату.

У першому рядку задано числа R, H, N, M. Де R – радіус області, яку освітлює прожектор. H – довжина сторони квадрата, що представляє килим. N – кількість вершин опуклого многокутника, що задає сцену. M – кількість килимів. У другому рядку знаходяться N пар чисел – координати вершин многокутника у порядку обходу (за або проти годинникової стрілки). У третьому рядку знаходяться M пар чисел – координати центрів килимів.

Вихідні дані

Створіть 10 вихідних файлів THEATER.S01, THEATER.S02, … , THEATER.S10 у вашому каталозі на дискеті. Ці файли повинні містити відповіді для відповідних вхідних файлів.

Кожен файл повинен містити два числа – цілі частини площ S1 та S2. Вам **не** потрібно здавати програму! Бали будуть нараховуватися за файли з правильними відповідями.

Приклад вхідних та вихідних даних

|  |  |
| --- | --- |
| THEATER.D00 | THEATER.S00 |
| 0.5 2 4 1  1 1 5 1 5 4 1 4  3 4 | 3 6 |

XX Всеукраїнська олімпіада з інформатики

Розділення багатокутників

На площині задано дві фігури, що обмежені опуклими багатокутниками. Фігури розташовані таким чином, що їх вершини не співпадають, а контури мають рівно 2 точки перетину.

Довільним чином розділимо площину прямою. Будемо вважати, що півплощина з одного боку прямої відповідає першій фігурі, а з іншого боку – другій фігурі. Визначимо поняття дефекту розділення – сума площі першої фігури, що розташована на півплощині другої фігури, та площі другої фігури, що розташована на півплощині першої фігури. З двох можливих відповідностей півплощин до фігур оберемо таку відповідність, де значення дефекту менше.



Наприклад, на рисунку зображена пряма, що задає певне розділення багатокутників. Оцінка дефекту цього розділення (два випадки відповідності): (D)+(C+E) та (A+D)+(B+C). З рисунку, D+C+E менше, отже, загалом, оцінка розбиття дефекту розділення утвореного цією прямою є D+C+E.

Завдання

Напишіть програму DIVIDE, що за заданими двома багатокутниками знаходить пряму, що утворює розділення з найменшим дефектом.

Вхідні дані

Перший рядок вхідного файлу DIVIDE.DAT містить одне ціле число N (3≤N≤20) – кількість вершин першого багатокутника. Наступні N рядків містять пари цілих чисел – координати вершин першого багатокутника у порядку обходу. (N+2)-й рядок файлу містить число M (3≤M≤20) – кількість вершин другого багатокутника. Наступні M рядків містять його координати задані таким же чином, як і для першого багатокутника. Координати точок додатні та не перевищують 1000.

Вихідні дані

Єдиний рядок вихідного файлу DIVIDE.SOL має містити дві пари чисел – координат двох точок, що однозначно задають розділення (пряму) з найменшим дефектом. Числа потрібно виводити у порядку: x1 y1 x2 y2. Координати потрібно виводити з точністю 10-3. Координати точок мають бути додатні та не більші за 1000.

Приклад вхідних та вихідних даних

|  |  |
| --- | --- |
| DIVIDE.DAT | DIVIDE.SOL |
| 3  2 1  3 3  4 1  3  5 2  3 2  4 3 | 2 5 4 1 |

XXI Всеукраїнська олімпіада з інформатики

Точки

На площині задано N точок. Окремо на площині задані дві базові точки.

Завдання

Напишіть програму POINTS, що знаходить максимальну кількість точок, що потраплять у смугу створену парою паралельних прямих довільно проведених через базові точки. Базові точки не потрібно включати до суми точок. Якщо точка лежить на прямій – її потрібно врахувати у сумі.



Вхідні дані

Перший рядок вхідного файлу POINTS.DAT містить одне ціле число N (0≤N≤10 000) – кількість точок. Другий рядок містить координати двох базових точок у форматі x1 y1 x2 y2. Кожен з наступних N рядків містить координати точки площини у форматі x y. Координати точок – цілі числа, що за модулем не перевищують 10 000. Базові точки відрізняються принаймні однією координатою.

Вихідні дані

Єдиний рядок вихідного файлу POINTS.SOL має містити ціле число – знайдену максимальну кількість точок, що потраплять у смугу, яка буде утворена оптимально проведеними паралельними прямими через базові точки.

Приклад вхідних та вихідних даних

|  |  |
| --- | --- |
| POINTS.DAT | POINTS.SOL |
| 4  0 0 50 0  0 -50  -1 0  50 0  100 50 | 3 |

XXII Всеукраїнська олімпіада з інформатики

Кола

На площині задано N різних кіл. Два кола перетинаються, якщо мають хоча б одну спільну точку.

Завдання

Напишіть програму CIRCLES, що за координатами центрів кіл та їх радіусами знайде пару кіл, що перетинається.

Вхідні дані

В першому рядку вхідного файлу CIRCLES.DAT міститься ціле число N (1≤N≤10 000) . В кожному з наступних N рядків міститься три натуральних числа *X, Y, R* менших за 10 000, що задають координати центру кола *(X, Y)* та його радіус *R*.

Вихідні дані

Єдиний рядок вихідного файлу CIRCLES.SOL має містити пару номерів кіл, що перетинаються, або єдине число 0, якщо жодні два кола не перетинаються. Кола нумеруються відповідно до порядку у вхідному файлі, починаючи з 1 до *N*. Якщо існує декілька пар кіл, що перетинаються, виведіть будь-яку з них. Елементи пари можуть бути виведені в довільному порядку.

Приклад вхідних та вихідних даних

|  |  |
| --- | --- |
| CIRCLES.DAT | CIRCLES.SOL |
| 5  5 10 4  6 20 3  10 15 3  12 8 2  13 13 1 | 5 3 |

XXIII Всеукраїнська олімпіада з інформатики

Пари точок

На площині своїми координатами задані *N* червоних та *N* синіх точок. Жодні три точки не належать одній прямій. Необхідно побудувати *N* відрізків таких, що ніякі два з них не перетинаються, кожен відрізок з’єднує точки різного кольору, і кожна точка належить рівно одному відрізку.

Завдання Напишіть програму PAIRS, яка за інформацією про розташування точок знайде будь-яке розбиття множини точок на *N* відрізків, кожен з яких сполучає синю і червону точки. При чому, кожна точка належить лише одному відрізку і ніякі два відрізка не перетинаються.

Вхідні дані Перший рядок вхідного файлу PAIRS.DAT містить єдине натуральне число – кількість блоків тестових даних, що задано у файлі. Блоки слідують один за одним. Кожен блок необхідно обробити окремо. Перший рядок кожного тестового блоку містить ціле число N (1≤N≤2500) – кількість точок одного кольору. Далі слідують набори координат синіх точок, а потім – червоних. Набір координат точок одного кольору задається *N* рядками, кожен з яких містить пару цілих чисел – x і y координати точки (|x|≤10 000, |y|≤10 000).

Вихідні дані Вихідний файл PAIRS.SOL має містити відповіді для всіх блоків тестових даних із вхідного файлу без розділення. Кожен з N рядків відповіді кожного з блоків має містити пару цілих чисел від 1 до *N* – номери синьої та червоної точок сполучених відрізком. Точки пронумеровані в тому порядку, в якому вони слідують у вхідному файлі окремо по кожному кольору. У випадку, якщо неможна сполучити точки відрізками без перетинів, єдиний рядок відповіді для блоку має містити число 0.

Оцінювання Щонайменше у 20% тестів буде виконуватись додаткове обмеження N≤10 та кількість блоків не перевищуватиме 2.

Приклад вхідних та вихідних даних

|  |  |
| --- | --- |
| PAIRS.DAT | PAIRS.SOL |
| 2  2  1 1  3 5  4 10  2 2  1  1 1  2 2 | 1 2  2 1  1 1 |

XXIV Всеукраїнська олімпіада з інформатики

Точки

На площині задано N точок.

Завдання Напишіть програму points, яка знайде суму квадратів відстаней між всіма парами точок.

Вхідні дані Перший рядок вхідного файлу points.dat містить єдине натуральне число N (1 ≤ N ≤ 100 000) – кількість точок. Наступні N рядків містять по два цілих числа X і Y (‑10 000 ≤ X, Y ≤10 000) – координати точок.

Вихідні дані Єдиний рядок вихідного файлу points.sol має містити єдине ціле число – суму квадратів відстаней між всіма парами точок.

Оцінювання Щонайменше у 50% тестів будуть виконуватись додаткові обмеження 1 ≤ N ≤ 1000, ‑1000 ≤ X, Y ≤1000. Щонайменше у 70% тестів будуть виконуватись додаткові обмеження 1 ≤ N ≤ 20 000.

Приклад вхідних та вихідних даних

|  |  |
| --- | --- |
| points.dat | points.sol |
| 4  1 1  -1 -1  1 -1  -1 1 | 32 |

XXV Всеукраїнська олімпіада з інформатики

1. Опукла оболонка

Задано множину точок на площині, а також *N* операцій, що модифікують цю множину. Кожна операція може бути одного з двох наступних типів:

1. **Додати до множини точку з цілими координатами. При цьому абсциса (тобто *X* координата) нової точки є строго більшою за абсциси всіх інших точок, які вже знаходяться у множині.**
2. **Видалити з множини точку з найбільшою абсцисою.**

Завдання Напишіть програму convex, яка за заданою послідовністю з *N* операцій промоделює їх виконання та після кожної з них виведе подвоєну площу опуклої оболонки точок множини. Перед виконанням першої операції множина точок є порожньою. Вважатимемо, що площа опуклої оболонки порожньої множини точок рівна нулю.

Опуклою оболонкою множини точок на площині називається опуклий багатокутник найменшої площі, який містить всі точки з множини. Багатокутник називається опуклим, якщо відрізок, що сполучає будь-які дві його точки цілком належить цьому багатокутнику. Тут під багатокутником розуміється границя фігури разом з її внутрішністю.

Вхідні дані Перший рядок вхідного файлу convex.dat містить одне ціле число: *N* (1≤*N*≤100 000) – кількість операцій, які потрібно виконати. Наступні *N* рядків містять по три цілих числа – інформацію про операції у зашифрованому форматі. Для того, щоб отримати параметри операції з номером *i*, потрібно зчитати три цілих числа *T*\*, *X*\* та *Y*\* (0≤*T*\*≤1, 0≤*X*\*, *Y*\*≤ 2 000 000 000) з (*i*+1)-го рядку, після чого отримати число *T* за наступною формулою: *T*=(*T*\*+*S*) mod 2 , де *S* - це подвоєна площа опуклої оболонки точок множини до виконання операції з номером *i*. Можна переконатись, що *S* завжди є цілим числом.

1. Якщо *T*=0, то чергова операція першого типу і координати (*X*,*Y*) нової точки отримуються за наступними формулами: *X*=*L*+((*S*+*X*\*) mod 2 000 000 001), *Y* = ((*Y*\*+*S*) mod 2 000 000 001) – 1 000 000 000. Тут *L* – це максимальна з абсцис усіх точок з множини. Якщо множина порожня, то *L*=-1 000 000 000.
2. Якщо *T*=1, то потрібно *K* разів виконати операцію другого типу. Число *K* знаходиться за формулою: *K*= ((*X*\*+*Y*\*+*S*) mod *Q*)+1. Тут *Q* – це кількість точок у множині.

Гарантується, що при правильній розшифровці всіх операцій виконуються наступні обмеження:

1. Координати усіх точок, які додаються до множині, не перевищують 1 000 000 000 за модулем.
2. При додаванні нової точки її абсциса є строго більшою за абсциси усіх точок, які вже лежать в множині.
3. Операція видалення застосовується лише до непорожньої множини точок.

Вихідні дані Вихідний файл convex.sol має містити *N* рядків. В рядок з номером *i* потрібно вивести подвоєну площу опуклої оболонки множини точок, яка утворилася після виконання перших *i* операцій. Якщо множина точок виявилася порожньою, то площу її опуклої оболонки вважати рівною 0.

Оцінювання Набір тестів складається з 4 окремих блоків, з такими додатковими обмеженнями:

1. 15% тестів: *N*≤300

2. 15% тестів: *N*≤3000

3. 20% тестів *N*≤100 000, кількість рядків, у яких *T* = 1 (після розшифрування) не перевищує 100

4. 50% тестів: *N*≤100 000

Приклад вхідних та вихідних даних

|  |  |
| --- | --- |
| convex.dat | convex.sol |
| 6  0 1000000000 1000000000  0 1000000 1001000000  0 1000000 999000000  0 1001500 1000001500  1 85072946 2  1 61619603 2 | 0  0  3000000000000  6000000000000  3000000000000  0 |

Пояснення Після розшифрування тест виглядає таким чином:

T = 0, X = 0, Y = 0

T = 0, X = 1000000, Y = 1000000

T = 0, X = 2000000, Y = -1000000

T = 0, X = 3000000, Y = 0

T = 1, K = 1

T = 1, K = 2