

Ножиці та скотч (scissors)

День	2
Мова	Українська
Обмеження по часу:	1 секунда
Обмеження по пам'яті:	1024 мегабайт

Дано шматок паперу у формі простого багатокутника S . Ваше завдання полягає у тому, щоб перетворити його на простий багатокутник T , що має ту ж саму площу, що і S .

Можна використовувати два інструменти: ножиці та скотч. Ножиці можуть бути використані для розрізання будь-якого багатокутника на менші багатокутні частини. Скотч може використовуватися для об'єднання менших частин у великі багатокутники. Кожен інструмент можна використовувати кілька разів, у будь-якому порядку.

Багатокутники задані у вхідних даних мають цілочисельні координати, але ви можете робити форми з **не цілочисельними координатами** у вашому виводі.

Далі йде формальне визначення завдання.

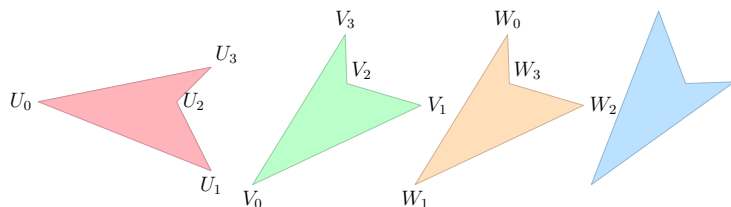
Форма $Q = (Q_0, \dots, Q_{n-1})$ являє собою послідовність з трьох або більше точок на площині так, що:

- Замкнена ламана $Q_0Q_1Q_2 \dots Q_{n-1}Q_0$ ніколи не доторкається та не перетинає саму себе, тому формує з себе контур замкнутого багатокутника.
- Ламана рухається по контуру проти годинникової стрілки.

Багатокутник, контур якого — форма Q буде позначено $P(Q)$.

Дві форми називаються **еквівалентними**, якщо одну можна перекласти та/або повернути так, щоб вона була рівна другій.

Зауважте, що відображення фігур не дозволяється. Також зауважте, що має значення порядок обходу вершин: форма $(Q_1, \dots, Q_{n-1}, Q_0)$ не обов'язково еквівалентна формі (Q_0, \dots, Q_{n-1}) .



На малюнку зліва: форми U та V еквівалентні. Форма W не є еквівалентною до них тому, що точки W задано в іншому порядку. Незалежно від порядку точок, четверта форма не еквівалентна попереднім, так як не можна перевертати форму.

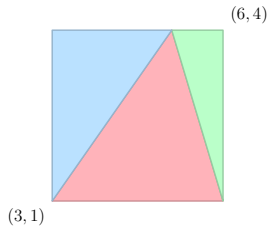
У вхідних та вихідних даних форма з n точок задана у одному рядку з $2n + 1$ чисел, розділених пробілами. Першим з цих чисел є число n . Інші числа — це координати точок: $Q_{0,x}, Q_{0,y}, Q_{1,x}, \dots$

Форми мають **ідентифікаційні номери (IDs)**. Задана форма S має ID 0, форми які ви оброблюєте у ваших рішеннях мають ID 1, 2, 3, \dots , у порядку, в якому вони утворюються.

Форми B_1, \dots, B_k формують **розділення** форми A , якщо:

- Об'єднання усіх $P(B_i)$ рівне $P(A)$.
- Для кожного $i \neq j$, ділянка перетину $P(B_i)$ та $P(B_j)$ є порожньою.

Операція **ножиці** видаляє одну існуючу форму A та додає одну, або декілька форм B_1, \dots, B_k , що формують розділення A .



На малюнку зліва: форму A (квадрат) розділено на форми B_1, B_2, B_3 (три трикутники). Один правильний спосіб описати один з B_i це «3 3 1 6 1 5.1 4».

Операція **скотч** видаляє одну, або декілька існуючих форм A_1, \dots, A_k та додає нову форму B . Для виконання цієї операції, спочатку потрібно вказати форми C_1, \dots, C_k та тільки потім одну форму B . Ці форми повинні задовольняти наступним умовам:

- Для кожного i , форма C_i еквівалентна формі A_i .
- Форми C_1, \dots, C_k формують розділення B .

Тобто ви вибираєте форму B і показуєте, як рухати кожну з існуючих A_i до правильної позиції C_i всередині B . Зверніть увагу, що тільки форма B отримує нове ID, а форми C_i не отримують.

Вхідні дані

Перший рядок містить початкову форму S .

Другий рядок містить необхідну форму T .

Кожна форма має від 3 до 10 точок, включно. Обидві форми подаються у форматі, вказаному вище.

Всі координати вхідних даних є цілими числами між -10^6 та 10^6 , включно.

У кожній формі жодні три точки не утворюють кут, менший за 3 градуси. (Це включає в себе не послідовні точки і має на увазі, що жодні три точки не є колінеарними.)

Багатокутники $P(S)$ та $P(T)$ мають однакову площу.

Вихідні дані

Коли ви використовуєте операцію ножиці, виведіть набір рядків у вигляді:

```
scissors
id(A) k
B_1
B_2
...
B_k
```

Де $id(A)$ — це ID форми, яку ви хочете видалити, k — це кількість нових форм, які ви хочете отримати, та B_1, \dots, B_k — це ці форми.

Кожен раз, коли ви використовуєте операцію скотч, виведіть набір рядків у вигляді:

```
tape
k id(A_1) ... id(A_k)
C_1
C_2
...
C_k
B
```

Де k — це кількість форм, які ви хочете склеїти разом, $id(A_1), \dots, id(A_k)$ — це їх ID, C_1, \dots, C_k є еквівалентними формами, що показують їх позиції всередині B , та B є кінцевою формою отриманою склеюванням їх разом.

Рекомендується виводити координати точок не менше, ніж з 10 знаками після коми.

Вихідні дані повинні задовольняти наступним умовам:



- Всі координати точок на виході повинні бути між -10^7 та 10^7 , включно.
- Кожна форма на виході повинна мати не більше 100 точок.
- У кожній операції кількість форм k має бути між 1 та 100, включно.
- Кількість операцій не повинна перевищувати 2000.
- Загальна кількість точок у всіх формах на виході не повинна перевищувати 20000.
- В кінці має бути одна форма (яка не була знищеною), і ця форма повинна бути еквівалентна T .
- Всі операції мають вважатись правильними відповідно до чекера. Рішення з малою точністю приймаються. (Два числа вважаються рівними, якщо їхня абсолютна або відносна похибка не перевищує 10^{-3} .)

Handouts

- Інструкції, як виводити дійсні числа, доступні в додатках до вашої мови програмування.
- Ви можете завантажити бінарний файл `scissors-checker`, можете зробити його виконуваним (`chmod a+x scissors-checker`) та використовувати його локально для перевірки правильності вашої відповіді (`./scissors-checker input your_output`).

Оцінювання

Форма називається **гарним прямокутником**, якщо вона має форму $((0, 0), (x, 0), (x, y), (0, y))$ для деяких додатних цілих x та y .

Форма називається **гарним квадратом**, якщо додатково $x = y$.

Форма A називається **строго опуклою**, якщо всі внутрішні кути багатокутника $P(A)$ менше 180 градусів.

Підзадача 1 (5 балів): S та T гарні прямокутники. Всі координати всіх точок є цілими числами від 0 до 10 включно

Підзадача 2 (13 балів): S є гарним прямокутником з $x > y$, та T це гарний квадрат

Підзадача 3 (12 балів): S та T це гарні прямокутники

Підзадача 4 (14 балів): S це трикутник та T є гарним квадратом

Підзадача 5 (10 балів): S та T є трикутниками

Підзадача 6 (16 балів): S є строго опуклим багатокутником та T є гарним прямокутником

Підзадача 7 (11 балів): T є гарним прямокутником

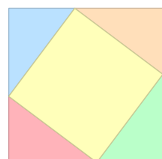
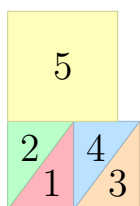
Підзадача 8 (19 балів): ніяких додаткових обмежень

Приклади

standard input	standard output
6 0 0 6 0 6 4 5 4 5 9 0 9 4 0 0 7 0 7 7 0 7	scissors 0 5 3 0 0 3 0 3 4 3 3 4 0 4 0 0 3 3 0 6 0 6 4 3 6 4 3 4 3 0 4 0 4 5 4 5 9 0 9 tape 5 1 2 5 3 4 3 0 3 0 0 4 0 3 4 0 7 0 7 4 4 0 3 4 0 7 4 3 7 3 7 4 7 7 3 7 3 3 7 0 7 0 3 4 0 0 7 0 7 7 0 7
4 0 0 3 0 3 3 0 3 4 7 -1 10 -1 11 2 8 2	scissors 0 2 3 0 0 1 3 0 3 4 1 3 0 0 3 0 3 3 tape 2 1 2 3 110 -1 111 2 110 2 4 108 2 107 -1 110 -1 110 2 4 107 -1 110 -1 111 2 108 2

standard input	standard output
4 0 0 9 0 9 1 0 1 4 0 0 3 0 3 3 0 3	scissors 0 2 4 1.4700000000 0 9 0 9 1 1.470000000 1 4 0 0 1.470000000 0 1.470000000 1 0 1 scissors 1 2 4 1.470000000 0 6 0 6 1 1.470000000 1 4 9 0 9 1 6 1 6 0 tape 2 4 3 4 3 2 3 1 6 1 6 2 4 6 1 1.470000000 1 1.470000000 0 6 0 6 1.470000000 0 6 0 6 2 3 2 3 1 1.47 scissors 5 4 4 1.470000000 0 3 0 3 1 1.470000000 1 4 3 0 4 0 4 2 3 2 4 4 2 4 0 5 0 5 2 4 5 0 6 0 6 2 5 2 tape 5 2 6 7 8 9 4 0 0 1.470000000 0 1.470000000 1 0 1 4 1.470000000 0 3 0 3 1 1.470000000 1 4 0 2 0 1 2 1 2 2 4 0 2 2 2 2 3 0 3 4 3 3 2 3 2 1 3 1 4 0 0 3 0 3 3 0 3

Примітка



Малюнок зліва показує приклад першого тесту. Зліва розташований вихідний малюнок після використання ножиць, праворуч — відповідний C_i коли ми склеюємо ці частини разом.

У другому прикладі вихідних даних зауважте, що достатньо, якщо кінцева форма еквівалентна потрібній, вони не обов'язково повинні бути ідентичними.

На малюнку нижче показано три етапи третього прикладу. По-перше, ми розрізали вхідний прямокутник на два менші прямокутники. Далі ми розрізали більший з цих двох прямокутників на два менших. Стан після цих розрізів показаний у верхній лівій частині малюнка.

Продовжуючи, ми склеюємо два нових прямокутника разом, щоб сформувати шестигранний багатокутник, а потім розрізаємо цей багатокутник на три прямокутники 2 на 1 та один менший прямокутник. Це показано в нижній лівій частині малюнка.

Нарешті, ми беремо прямокутник, який ми маємо ще з першого кроку та чотири нових прямокутники, і збираємо їх у бажаний квадрат 3 на 3.

