

Scissors & Tape (scissors)

Day	2
Language	Romanian
Time limit:	1 second
Memory limit:	1024 megabytes

Vi se da o foaie de hartie in forma unui poligon simplu S . Cerinta voastra este sa o transformati intr-un alt poligon simplu T ce are aria egala cu poligonul initial S .

Puteti folosi urmatoarele doua operatii: scissors si tape. Scissors poate fi folosita pentru a taia un poligon in mai multe poligoane mai mici. Tape poate fi folosita pentru a combina mai multe poligoane mici intr-un poligon mai mare. Puteti folosi oricare dintre operatii de mai multe ori, in orice ordine.

Poligonul primit ca input are varfurile in coordonate intregi, dar aveti posibilitatea sa produceti ca modele intermediare poligoane cu varfuri la coordonate **numerele reale (non-integer)**.

O definitie formală a cerintei ar fi urmatoarea:

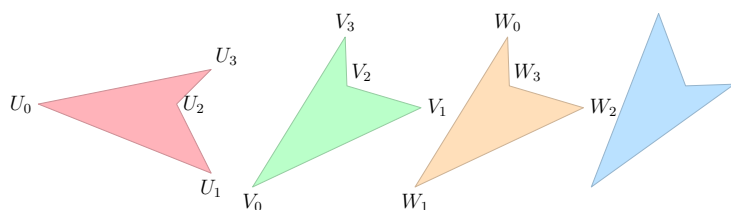
Un **model** $Q = (Q_0, \dots, Q_{n-1})$ este o secventa de 3 sau mai multe puncte in plan astfel incat:

- Linia poligonala inchisa (polyline) $Q_0Q_1Q_2 \dots Q_{n-1}Q_0$ nu se atinge pe sine sau autointersecteaza, constituind astfel perimetrul unui poligon simplu.
- Linia poligonala (polyline) merge de-a lungul perimetrului poligonului in sens trigonometric (counterclockwise).

Poligonul corespunzator perimetrului definit de modelul Q va fi notat $P(Q)$.

Doua forme se numesc **echivalente** daca una se poate obtine prin translatia si/sau rotirea celeilalte..

Oglindirea (mirroring) unei forme nu este permisa. Luati seama ca ordinea punctelor din model conteaza: modelul $(Q_1, \dots, Q_{n-1}, Q_0)$ nu este obligatoriu echivalenta cu modelul (Q_0, \dots, Q_{n-1}) .



In figura din stanga: Formele U si V sunt echivalente. Insa modelul W nu este echivalent cu primele doua pentru ca punctele din modelul W sunt date intr-o ordine diferita. Independent de ordine, cel de-al patrulea model nu este echivalent cu cele precedente deoarece oglindirea nu este permisa.

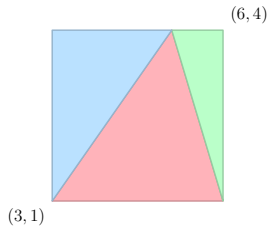
Atat in input cat si in output, un model de n puncte este reprezentat ca o singura linie ce contine $2n + 1$ numere separate prin spatiu. Primul dintre aceste numere este n . Numerele ramase reprezinta coordonatele punctelor: $Q_{0,x}, Q_{0,y}, Q_{1,x}, \dots$

Modelul are un **numar de identificare** (IDs). Modelul primit S are ID-ul 0, modelele produse in solutie primesc ID-urile 1, 2, 3, \dots , exact in ordinea in care au fost produse.

Modelele B_1, \dots, B_k formeaza o **subdiviziune** a modelului A daca:

- Reuniunea tuturor $P(B_i)$ este exact $P(A)$.
- Pentru oricare $i \neq j$, aria intersectiei dintre $P(B_i)$ si $P(B_j)$ este zero.

Operatia **scissors** taie un model deja existent A si produce unul sau mai multe modele B_1, \dots, B_k ce constituie o subdiviziune a lui A .



In figura din stanga: Modelul A (un patrat) este taiat in 3 modele B_1, B_2, B_3 (3 triunghiuri). Un mod valid de a descrie unul dintre B_i este “3 3 1 6 1 5.1 4”.

Operatia **tape** lipeste unul sau mai multe modele existente A_1, \dots, A_k si produce un model nou B . Pentru a aplica aceasta operatie, aveti nevoie sa specificati modelele C_1, \dots, C_k si abia apoi modelul final B . Aceste modele trebuie sa satisfaca urmatoarele:

- Pentru orice i , modelul C_i trebuie sa fie echivalent cu modelul A_i .
- Modelele C_1, \dots, C_k trebuie sa formeze o subdiviziune a modelului B .

Informal, alegeti un model B si prezentati cum mutati fiecare dintre modelele existente A_i la noua locatie C_i in interiorul lui B . Luati aminte ca doar modelul nou format B primeste un ID, modelele auxiliare folosite C_i sunt doar temporare.

Input

Prima linie contine modelul sursa S .

Cea de-a doua linie contine modelul target T .

Fiecare model contine intre 3 si 10 puncte, inclusiv. Amandoua modelele respecta formatul mentionat anterior.

Toate coordonatele din input sunt numere intregi cu valori intre -10^6 si 10^6 , inclusiv.

In fiecare model, nu exista 3 puncte ce sa formeze un unghi de mai putin de 3 grade. (Aceasta mentine garanteaza ca nu exista 3 puncte coliniare si se aplica pentru oricare 3 puncte, ce nu trebuie neaparat sa fie consecutive in definirea modelului.)

Poligoanele $P(S)$ si $P(T)$ au exact aceeasi arie.

Output

Cand folositi operatia scissors, afisati un bloc de linii cu urmatorul format:

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

unde $id(A)$ este ID-ul modelului ce doriti sa il modificati, k este numarul de noi modele ce doriti sa le produceti si B_1, \dots, B_k fiind aceste modele.

Cand folositi operatia tape, afisati un bloc de linii cu urmatorul format:

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

unde k este numarul de modele ce vor fi lipite cu $id(A_1), \dots, id(A_k)$ ID-urile lor. C_1, \dots, C_k sunt modelele echivalente ce arata pozitia lor in interiorul modelului final B . B este modelul ce va fi obtinut in urma operatiei de lipire.



Este recomandat ca afisarea coordonatelor punctelor sa contina cel putin 10 zecimale.

Afisarea trebuie sa respecte urmatoarele:

- Toate coordonatele punctelor afisate trebuie sa aiba valori intre -10^7 si 10^7 , inclusiv.
- Fiecare model din output trebuie sa contina cel mult 100 de puncte.
- In fiecare operatie numarul k de modele produse/primate trebuie sa fie intre 1 si 100 inclusiv.
- Numarul de operatii nu trebuie sa depaseasca 2000.
- Numarul total de puncte din toate modelele afisate nu trebuie sa depaseasca 20000.
- La final trebuie sa existe un singur model si acela trebuie sa fie echivalent cu T .
- Toate operatiile facute trebuie sa fie valide in acord cu checker-ul. Solutii cu erori mici vor fi acceptate. (Checker-ul verifica toate comparatiile pentru o valoare absoluta a erorii de maximum 10^{-3} pentru fiecare conditie.)

Handouts

- Instructiuni referitoare la printarea numerelor reale (floating-point) sunt disponibile in documentatia limbajului de programare ales din sectiunea Handouts.
- Binarul ce executa checker-ul oficial se poate downloada din Handouts: fisierul binar **scissors-checker**. Pentru a face fisierul executabil rulati initial comanda de terminal (**chmod a+x scissors-checker**), avand terminalul deschis in folderul ce contine checker-ul. Pentru a verifica corectitudinea outputului generat si copiat in acelasi director, executati comanda de terminal (**./scissors-checker input your_output**).

Scoring

Un model este numit **dreptunghi misto** daca acesta este de forma $((0,0), (x,0), (x,y), (0,y))$ unde x si y sunt numere pozitive intregi.

Un model este numit **patrat misto** daca, in plus, $x = y$.

Un model A este numit **strict convex** daca toate unghiurile interioare ale poligonului $P(A)$ sunt mai mici de 180 de grade.

Subtask 1 (5 puncte): S si T sunt dreptunghiuri misto. Toate coordonatele tuturor punctelor sunt numere intregi cuprinse intre 0 si 10 inclusiv.

Subtask 2 (13 puncte): S este un dreptunghi misto cu $x > y$, si T este un patrat misto

Subtask 3 (12 puncte): S si T sunt dreptunghiuri misto

Subtask 4 (14 puncte): S este un triunghi si T este un patrat misto

Subtask 5 (10 puncte): S si T sunt triunghiuri

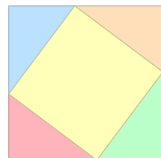
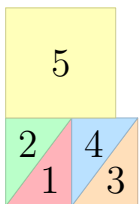
Subtask 6 (16 puncte): S este un poligon strict convex si T este un dreptunghi misto

Subtask 7 (11 puncte): T este un dreptunghi misto

Subtask 8 (19 puncte): fara restrictii suplimentare

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

Note



In figura din stanga este prezentat outputul primului exemplu. In partea stanga este figura originala dupa folosirea operatiilor scissors, in dreapta sunt C_i -urile corespunzatoare aplicarii operatiei tape.

In al doilea exemplu subliniem ca este suficient ca modelul rezultat sa fie echivalent cu cel dorit/target, nu trebuie neaprat sa fie identice.

Figura de mai jos prezinta cei 3 pasi aplicati asupra celui de-al treilea exemplu. Prima data taiem dreptunghiul primit in input in dreptunghiuri mai mici, apoi taiem pe cel mai mare dintre ele in alte doua. Starea de dupa aceste taieri este prezentata in partea de sus stanga a figurii.

In continuare, aplicam operatia tape pe cele 2 dreptunghiuri pentru a forma un poligon de 6 laturi, apoi taiem poligonul in 3 dreptunghiuri 2x1 si un alt dreptunghi mai mic. Acestea sunt prezentate in partea jos stanga.

In final, luam dreptunghiurile ramase de dupa primul pas plus cele 4 noi formate si le lipim in patratul target 3x3.

