Škarje in lepilo (scissors)

Dan	2
Jezik	slovenščina
Omejitev časa:	1 sekunda
Omejitev pomnilnika:	1024 MB

Dan je list papirja v obliki enostavnega mnogokotnika S. Tvoja naloga je, da ga preoblikuješ v enostaven mnogokotnik T z enako ploščino.

Uporabljaš lahko dve orodji: škarje in lepilni trak. S škarjami lahko razrežeš mnogokotnik na manjše mnogokotnike. Z lepilnim trakom lahko lepiš manjše koščke v večje mnogokotnike. Obe orodji lahko uporabiš poljubno mnogokrat v poljubnem vrstnem redu.

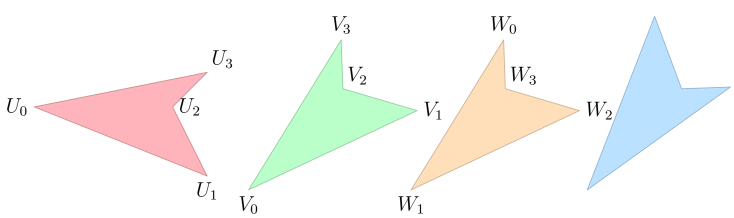
Sledi formalna definicija naloge.

Oblika $Q = (Q_0, \dots, Q_{n-1})$ je zaporedje treh ali več točk v ravnini, tako da velja:

- Sklenjena lomljenka $Q_0 Q_1 Q_2 \dots Q_{n-1} Q_0$ se nikoli ne dotika ali seka in torej predstavlja rob enostavnega mnogokotnika.
- Lomljenka je podana v nasprotni smeri urinega kazalca.

Mnogokotnik z robom Q bomo označili s $P(Q)$.

Dve obliki sta **ekvivalentni**, če lahko eno premaknemo ali zavrtimo tako, da sovpada z drugo. Zrcaljenje oblik ni dovoljeno. Pazi tudi na to, da je vrsti red točk pomemben: oblika $(Q_1, \dots, Q_{n-1}, Q_0)$ ni nujno ekvivalentna obliki $(Q_0, \dots, Q_{n-1})$.



Na sliki na levi imamo naslednjo situacijo. Obliki U in V sta ekvivalentni. Oblika W jima ni ekvivalentna, saj so točke drugače oštevilčene. Ne glede na drug vrstni red točk, četrta oblika ni ekvivalentna nobeni prejšnji, saj zrcaljenje oblik ni dovoljeno.

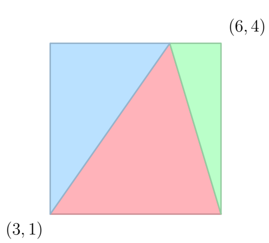
Tako na vhodu kot v izhodu bo oblika z n točkami predstavljena v eni vrstici, ki vsebuje $2n+1$ števil. Prvo od njih je n. Preostale številke so koordinate točk $Q_{0,x}$, $Q_{0,y}$, $Q_{1,x}$, ...

Oblike imajo **identifikacijske številke** (ID). Oblika, dana na vhodu, ima ID 0, oblike, ki jih narediš v svoji rešitvi pa imajo po vrsti ID-je 1, 2, 3, ...

Oblike $B_1, \dots, B_k$ tvorijo **razrez** oblike A, če velja:

- Unija vseh $P(B_i)$ je natanko $P(A)$.
- Za vsak $i \neq j$ je ploščina preseka $P(B_i)$ in $P(B_j)$ enaka 0 (ploščina robu je 0).

Operacija `scissors` uniči eno obstoječo obliko A in ustvari eno ali več oblik $B_1, \dots, B_k$, ki tvorijo razrez A.



Na levi sliki je oblika A (kvadrat) razrezana na oblike B_1, B_2, B_3 (trije trikotniki). En veljaven način, da opišemo enega izmed B_i, je "3 3 1 6 1 5.1 4".

Operacija `tape` uniči eno ali več obstoječih oblik $A_1, \dots, A_k$ in ustvari eno novo obliko B. Za to operacijo je potrebno podati oblike $C_1, \dots, C_k$ in šele nato obliko B. Te oblike morajo ustrezati naslednjim zahtevam:

- Za vsak i mora biti oblika C_i ekvivalentna obliki A_i.
- Oblike $C_1, \ldots, C_k$ morajo tvoriti razrez oblike B.

Neformalno, izbereš si obliko B in pokažeš, kako vsako izmed obstoječih A_i premaknemo na njeno pripadajoče mesto C_i znotraj B. Samo oblika B dobi nov ID, oblike C_i pa ne.

Vhod

Prva vrstica vsebuje začetno obliko S. Druga vrstica vsebuje ciljno obliko T. Mnogokotnika $P(S)$ in $P(T)$ imata enako ploščino.

Vsaka oblika je sestavljena od 3 do 10 točk, vključujoč obe meji. Obe obliki sta podani z zaporedjem koordinat kot navedeno prej. Vse koordinate v vhodu so cela števila med vključno -10^6 in 10^6.

Nobene tri točke ne tvorijo kota manjšega kot 3 stopinje. (To velja tudi za nezaporedne točke in implicira, da nobene tri točke ne ležijo na isti premici, ter da ni težav pri računanju z decimalnimi števili.)

Izhod

Ko uporabiš operacijo `scissors`, izpiši vrstice

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

kjer je id(A) ID oblike, ki jo želimo uničiti, k je število novih oblik, ki jih želiš ustvariti, $B_1, \ldots, B_k$ pa so te oblike, podane s seznamom koordinat, kot opisano prej.

Ko uporabiš operacijo `tape`, izpiši vrstice

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

kjer je k število oblik, ki jih želiš zlepiti, id$(A_1), \ldots,$ id(A_k) so njihovi ID-ji, $C_1, \ldots, C_k$ so ekvivalentne oblike, ki podajajo njihove položaje znotraj B, B pa je končna oblika, ki jo dobimo pri lepljenju.

Priporočeno je, da izpišeš decimalke točk na vsaj 10 decimalnih mest (glej tehnična navodila pod "Handouts", če ne veš kako).

Za izhod mora veljati:

- Koordinate točk v izhodu morajo biti med vključno -10^7 in 10^7.
- Vsaka oblika v izhodu ima lahko največ 100 točk.
- V vsaki operaciji mora biti število oblik k med vključno 1 in 100.
- Število operacij ne sme preseči 2000.
- Skupno število točk v vseh oblikah v izhodu ne sme preseči 20 000.
- Po koncu operacij mora ostati natanko ena neuničena oblika, in ta oblika mora biti ekvivalentna obliki T.
- Vse operacije morajo biti narejene po opisanih pravilih: njihovo veljavnost bo preverjal ocenjevalni sistem, ki je tudi **na voljo za prenos in lastno zaganjanje**.

Preneseš lahko datoteko `scissors-checker`, jo spremeniš v executable (`chmod a+x scissors-checker`) in preveriš svoj izhod s "`./scissors-checker input your_output`".

Rešitve bodo sprejete tudi s precej velikimi računskimi napakami: ocenjevalni program preverja, če so relativne ali absolutne vrednosti napak pri preverjanju pogojev pod 10^{-3}.

Ocenjevanje

Obliki pravimo **negovan pravokotnik**, če ima koordinate $((0,0),\ (x,0),\ (x,y),\ (0,y))$ za neki pozitivni celi števili x in y.

Obliki pravimo **negovan kvadrat**, če dodatno velja še $x = y$.

Obliki A pravimo, da je strogo konveksna, če so vsi notranji koti mnogokotnika $P(A)$ strogo manjši kot 180 stopinj.

Podnaloga 1 (5 točk): S in T sta negovana pravokotnika. Vse koordinate so cela števila med vključno 0 in 10.

Podnaloga 2 (13 točk): S je negovan pravokotnik z $x > y$ in T je negovan kvadrat.

Podnaloga 3 (12 točk): S in T sta negovana pravokotnika

Podnaloga 4 (14 točk): S je trikotnik in T je negovan kvadrat

Podnaloga 5 (10 točk): S in T sta trikotnika

Podnaloga 6 (16 točk): S je strogo konveksen mnogokotnik in T je negovan pravokotnik

Podnaloga 7 (11 točk): T je negovan pravokotnik

Podnaloga 8 (19 points): brez dodatnih omejitev

Primeri

standardni vhod	standardni izhod

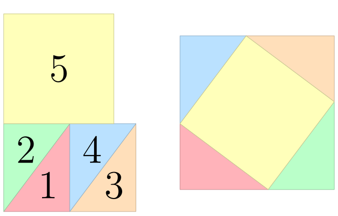
`6 0 0 6 0 6 4 5 4 5 9 0 9`
`4 0 0 7 0 7 7 0 7`
	`scissors`
	`0 5`
	`3 0 0 3 0 3 4`
	`3 3 4 0 4 0 0`
	`3 3 0 6 0 6 4`
	`3 6 4 3 4 3 0`
	`4 0 4 5 4 5 9 0 9`
	`tape`
	`5 1 2 5 3 4`
	`3 0 3 0 0 4 0`
	`3 4 0 7 0 7 4`
	`4 0 3 4 0 7 4 3 7`
	`3 7 4 7 7 3 7`
	`3 3 7 0 7 0 3`
	`4 0 0 7 0 7 7 0 7`

`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`

standardni vhod	standardni izhod

`4 0 0 9 0 9 1 0 1`
`4 0 0 3 0 3 3 0 3`
	`scissors`
	`0 2`
	`4 1.47000000000 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 1`
	`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`

Komentar



Slika na levi ilustrira prvi primer. Na levi je začetna oblika po uporabi operacije `scissors`, na desni pa so pripadajoči C_i, ko koščke zlepimo skupaj.

Drugi primer kaže, da je dovolj, da je končna oblika le ekvivalentna ciljni obliki; ni potrebno, da sovpadata.

Spodnja slika kaže tri korake tretjega primera. Najprej razrežemo vhodni pravokotnik na dva manjša pravokotnika, nato pa večjega od teh razrežemo še enkrat. Stanje po teh razrezih je prikazano v zgornjem levem kotu.

Potem zlepimo novonastala pravokotnika v šestkotnik, in nato tega razrežemo na tri pravokotnike velikosti 2×1 in še enega manjšega. To je prikazano v levem spodnjem kotu.

Na koncu vzamemo pravokotnik, ki ga imamo iz prvega koraka, in nove štiri pravokotnike ter jih zlepimo v željen 3×3 kvadrat.