Háromszögek

Byteland egy szép ország, ahol n város van. A városok a sík n különböző pontjával adottak. Nem ismerjük a városok koordinátáit, de tudjuk, hogy semelyik három pont nem esik egy egyenesre.

Kiszámítandó a városokat tartalmazó konvex burok csúcsainak száma! A feladat megoldásához kérdéseket tehetsz fel. Egy kérdés három különböző város sorszámát tartalmazza:. A válasz a három pont által meghatározott háromszög körüljárási irányát adja meg. Igaz értéket ad, ha a pontok sorrendje órajárással megegyező, egyébként hamisat.

Kommunikáció

A programodnak egy könyvtári modult kell használnia, amely a kérdéseket és a válaszadást valósítja meg.

A modul (`trilib.h` C és C++ nyelvekre) a következő függvényeket tartalmazza:

- `int get_n();` A városok számát adja.
- `bool is_clockwise(int a, int b, int c);` `true` értéket ad, ha az `a`, `b`, `c` ($1 \le a,b,c \le n$, és páronként különbözőek) pontokkal meghatározott háromszög körüljárása órajárás szerinti, és `false` értéket ad, ha órajárással ellentétes.
- `void give_answer(int s);` Ezzel kell közölnöd a kiszámítandó eredményt, azaz `s` a konvex burok csúcsainak száma.

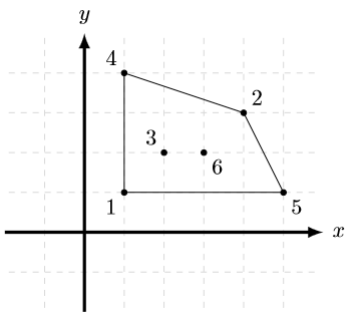
Ebben a feladatban a programod nem olvashat a standard bemenetről, és nem írhat a standard kimenetre!

A `give_answer` hívása után programodnak azonnal terminálnia kell!

Feltételezheted, hogy a pontok koordinátái előre meghatározottak, és nem változnak a program végrehajtása során (vagyis a modul teljes mértékben determinisztikusan viselkedik). Például, ha az alábbi példa tesztesetben a programod végrehajtja a `give_answer(4)` utasítást és rögtön terminál, helyes értékelést kapsz. Megengedett, hogy a program tippeljen a válasz kiszámítása nélkül.

Példa

Legyen n =6 város a következő pontokban: (1,1), (4,3), (2,2), (1,4), (5,1), (3,2), amit az alábbi ábra mutat! A konvex burkot vonalakkal jelöltük, amelynek négy csúcsa van, tehát az eredmény 4.

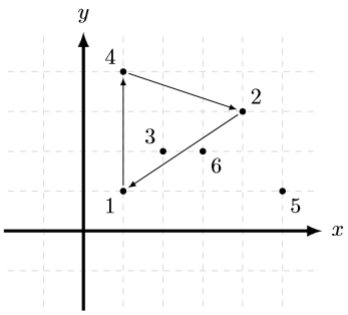


Az alábbi táblázat egy lehetséges interakciót tartalmaz a fenti példához.

Művelet	Kapott érték
`get_n()`	6
`is_clockwise(1,4,2)`	`true`

`is_clockwise(4,2,1)`	`true`
`is_clockwise(1,2,4)`	`false`
`is_clockwise(3,6,5)`	`true`
`give_answer(4) -`	

Az alábbi háromszög az első kérdéshez tartozik. Az 1,4,2 pontok által meghatározott háromszög körüljárási iránya az óramutató járásával megegyezik, tehát a kapott érték true.



Korlátok

Időlimit: 1 mp

Memórialimit: 256 MB

Pontozás

Minden tesztesetben $3 \leq n \leq 40\,000$.

Tesztesetenként az `is_clockwise` függvényt legfeljebb $1\,000\,000$-szor hívhatod meg.

1. tesztcsoport (32 pont): $n \leq 50$

2. tesztcsoport (32 pont): $n \leq 500$

3. tesztcsoport (32 pont): $n \leq 15000$

4. tesztcsoport (65 pont): legfeljebb egy pont nem esik a konvex burok oldalaira

5. tesztcsoport (32 pont): nincs egyéb feltétel

Gyakorlás

A `public` mappában kapsz egy példa könyvtári modult, amellyel tesztelheted a programod formális működését. A modul a *`standard inputról`* olvassa be az adatokat az alábbi formában:

- • Az első sor a városok N számát tartalmazza,
- • a következő N sor mindegyike egy város x és y koordinátáját tartalmazza.

A példa modul nem ellenőrzi a megoldás helyességét. A bemenet helyességét sem ellenőrzi. Természetesen az értékelő szerver nem ezt a modult használja.

Egy példa bemenet található a `tri0.in` fájlban.

A `give_answer` hívás végrehajtása után a modul kiírja a *`standard kimenetre`* az általad adott válasz értékét, és a végrehajtott `is_clockwise` hívások számát.

A megoldásod lefordítására a példa modullal, a következő parancsokat használhatod:

• C: `gcc -O2 -static trilib.c tri.c -lm -std=gnu99`

• C++: `g++ -O2 -static trilib.c tri.cpp -lm -std=c++11`

A megoldásodnak és a modult tartalmazó fájloknak ugyanabban a könyvtárban kell lennie.