Problema: TRI
Triangles

romanian

CEOI 2018 Warsaw POLAND

CEOI 2018, ziua 2. Memorie disponibila: 256 MB. *16.08.2018*

Byteland este o tara formata din n ($n \geq 3$) orase, reprezentate prin n puncte distincte in planul 2D. Orasele sunt numerotate de la 1 la n. Ca turist, nu stii pozitiile exacte ale oraselor din Byteland. Dintr-o revista turistica ai aflat ca nu exista trei orase coliniare.

Infasuratoarea convexa a unei multimi de n puncte este poligonul convex de arie minima cu proprietatea ca toate cele n puncte din multime sunt in interiorul sau pe marginea acestuia. Un poligon convex are toate unghiurile mai mici de 180 de grade si nu se poate auto-intersecta.

Sarcina voastra este sa gasitit numarul de puncte de pe marginea infasuratorii convexe a multimii de puncte ce reprezinta orasele din Byteland. Puteti pune intrabari identificate prin triplete de orase **distincte** numerotate cu i, j, k ($1 \leq i, j, k \leq n$). O astfel de intrebare vizeaza un triunghi cu varfurile in orasele i, j, k. Raspunsul la intrebare indica daca traversarea varfurilor triunghiului in ordinea i, j, k este in sensul acelor de ceasornic sau in sens trigonometric.

Interactiune

Programul vostru va folosi o librarie ce permite punerea intrebarilor de forma specificata in enunt si transmiterea raspunsului final.

Libraria (`trilib.h` pentru C si C++) contine urmatoarele functii:

- `int get_n();`
  Returneaza numarul de orase.
- `bool is_clockwise(int a, int b, int c);`
  Returneaza `true` daca varfurile triunghiul a, b, c ($1 \leq a, b, c \leq n, a \neq b \neq c \neq a$) sunt date in sensul acelor de ceasornic si `false` daca sunt date in sens trigonometric.
- `void give_answer(int s);`

Pentru Java, clasa `trilib` contine urmatoarele metode:

- `static public int get_n();`
- `static public boolean is_clockwise(int a, int b, int c);`
- `static public void give_answer(int s);`

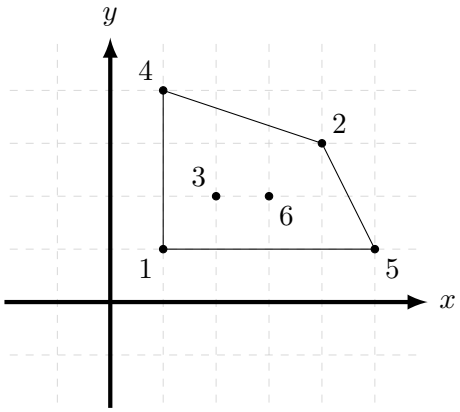
Dupa ce programul vostru apeleaza functia `give_answer` nu se mai permite punerea de intrebari. Programul vostru trebuie sa apeleze functia `give_answer` exact o data.

Nu este permisa citirea de la tastatura sau afisarea pe ecran. Dupa apelarea functiei `give_answer`, programul vostru trebuie sa isi termine imediat executia.

Puteti presupune ca locatiile punctelor sunt stabilite in avans si ca nu se vor schimba pe parcursul executiei programului (cu alte cuvinte, libraria are un comportament perfect determinist). De exemplu, in urmatorul exemplu (vezi mai jos) apelarea `give_answer(4)` si terminarea imediata a executiei va trece testul cu succes. Programului vostru ii este permis sa ghiceasca raspunsul.

Exemplu de interactiune

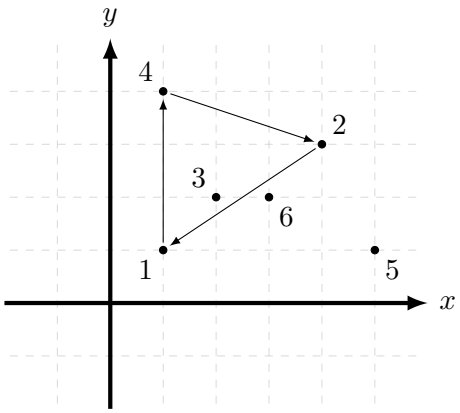
Fie $n = 6$ orase situate in punctele $(1, 1)$, $(4, 3)$, $(2, 2)$, $(1, 4)$, $(5, 1)$, $(3, 2)$ ca in imaginea de mai jos. Infasuratoarea convexa este marcata prin segmentele negre. Ea contine patru varfuri pe marginile sale, deci raspunsul este 4.



Urmatorul tabel prezinta un exemplu de interactiune cu libraria ce corespunde acestui exemplu.

Call	Returned value
`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)`	–

Imaginea de mai jos evidentiaza triunghiul din prima intrebare. Orasele 1, 4, 2 sunt date in sensul acelor de ceasornic, deci valoarea returnata este `true`.



Grading

Setul de teste este impartit in subtask-uri cu restrictii suplimentare, dupa cum urmeaza. Testele din fiecare subtask sunt impartite in una sau mai multe grupe de teste. Fiecare grupa de teste poate contine unul sau mai multe teste.

Pentru toate testele avem ca $3 \leq n \leq 40\,000$. Puteti apela functia `is_clockwise` de cel mult $1\,000\,000$ de ori per test.

Subtask	Restrictii	Nr. puncte
1	$n \leq 50$	15
2	$n \leq 500$	20
3	$n \leq 15\,000$	20
4	cel mult un punct nu este pe marginea infasuratorii convexe	20
5	fara restrictii suplimentare	25

Testare

In directorul `public` va este pusa la dispozitie o librarie model ce va permite testarea corectitudinii formale a solutiei voastre. Libraria citeste de la tastatura setul de puncte ce alcatuiesc Byteland in urmatorul format:

- pe prima linie un numar intreg n, reprezentand numarul de orase,
- pe urmatoarele n linii: doua numere intregi pe fiecare, cele de pe a i-a linie reprezentand coordonatele celui de-al i-lea oras.

Librarie pusa la dispozitie voastra **nu** va verifica corectitudinea solutiei produse de voi. De asemenea, libraria nu valideaza datele de intrare (nu le verifica corectitudinea).

De asemenea, libraria nu este aceeasi cu libraria secreta de pe server-ul de evaluare.

Un exemplu de fisier de intrare pentru librarie va este dat in `tri0.in`.

Dupa ce functia `give_answer` este apelata, libraria va afisa pe ecran raspunsul dat de program, impreuna cu numarul de apeluri ale functiei `is_clockwise` folosite.

Pentru compilarea solutiei impreuna cu libraria model puteti folosi una dintre urmatoarele comenzi de compilare:

- **C**: `gcc -O2 -static trilib.c tri.c -lm -std=gnu99`
- **C++**: `g++ -O2 -static trilib.c tri.cpp -lm -std=c++11`

Pentru Java nu va trebui sa folositi vreo comanda speciala de compilare. Fisierele solutiei si libraria trebuie sa fie in acelasi director.