

Bajtija je prelestna dežela z n ($n \geq 3$) mesti, ki so predstavljena kot n različnih točk v 2D ravnini. Mesta so označena s številkami od 1 do n . Kot turist seveda ne poznate natančnih lokacij posameznih bajtiskih mest, le v turističnem vodiču ste prebrali, da nobena tri mesta ne ležijo na isti premici.

Konveksna ovojnica n točk je konveksni mnogokotnik z najmanjšo možno ploščino, ki vsebuje vseh n točk. Vsebuje pomeni, da so točke lahko v notranjosti mnogokotnika ali na njegovem robu. Mnogokotnik je konveksen, če so vsi njegovi notranji koti manjši od 180 stopinj in se stranice ne sekajo.

Vaša naloga je ugotoviti, koliko točk leži na robu konveksnega mnogokotnika bajtiskih mest. Pri tem lahko postavljate vprašanja o trojicah **različnih** mest i, j in k ($1 \leq i, j, k \leq n$). Vprašanje je dejansko zastavljeno o trikotniku mest i, j in k ter odgovor sporoča, ali je obhod mest iz i v j in od tam v k opravljen v smeri urinega kazalca ali obratno.

Raba knjižnice

Vaš program naj uporablja knjižnico, ki omogoča poizvedbe in sporočanje končnega rezultata.

Za programska jezika C in C++ knjižnica `trilib.h` nudi naslednje funkcije:

- `int get_n()`;
vrne število mest (točk).
- `bool is_clockwise(int a, int b, int c)`;
vrne `true`, če so točke trikotnika a, b, c ($1 \leq a, b, c \leq n, a \neq b \neq c \neq a$) podane v smeri urinega kazalca in `false`, če so podane v nasprotni smeri urinega kazalca.
- `void give_answer(int s)`;

Za programski jezik Java (Gregor!) razred `trilib` nudi naslednje enakovredne metode:

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

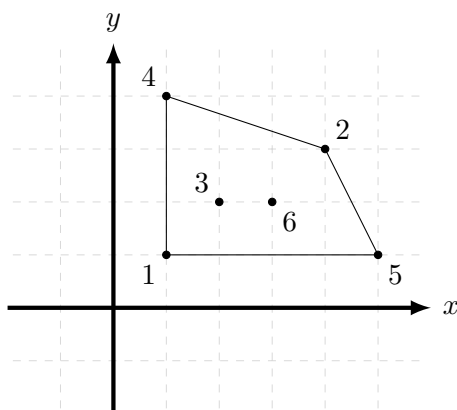
Vaš program ne sme več postavljati vprašanj po tem, ko pokliče `give_answer` ter naj bi poklical `give_answer` natančno enkrat in po klicu takoj končal izvajanje.

Pri tej nalogi ne smete brati s standardnega vhoda in ne pisati na standardni izhod.

Lahko predpostavite, da so točke določene v naprej in se med izvajanjem ne spreminjajo (z drugimi besedami knjižnica je povsem deterministična). Na primer, v spodnjem primeru bi program, ki bi samo klical `give_answer(4)` in takoj za tem zaključil izvajanje, uspešno opravil test. Program lahko namreč poskusi uganiti rezultat, ne da bi bil prepričan vanj.

Primer interakcije

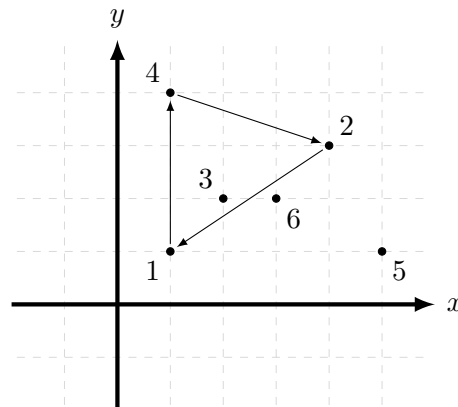
Imejmo $n = 6$ mest na koordinatah $(1, 1)$, $(4, 3)$, $(2, 2)$, $(1, 4)$, $(5, 1)$ in $(3, 2)$, kot je predstavljeno na spodnji sliki. Konveksna ovojnica je označena s črtami in, ker vsebuje štiri točke na svojem robu, je rezultat 4.



Naslednja tabela ponazarja primer interakcije s knjižnico v podanem primeru:

klic	vrnjena vrednost
<code>get_n()</code>	6
<code>is_clockwise(1, 4, 2)</code>	true
<code>is_clockwise(4, 2, 1)</code>	true
<code>is_clockwise(1, 2, 4)</code>	false
<code>is_clockwise(3, 6, 5)</code>	true
<code>give_answer(4)</code>	-

Spodnja slika prikazuje trikotnik iz prve poizvedbe. Ker so mesta 1, 4 in 2 podana v smeri urinega kazalca, je vrnjena vrednost true.



Ocenjevanje

Testni primeri so razdeljeni v sledeče podnaloge z dodatnimi omejitvami. Vsaka podnaloge vsebuje eno ali več skupin testnih primerov. Vsaka skupina prav tako lahko vsebuje enega ali več testnih primerov.

Pri vseh testih velja $3 \leq n \leq 40\,000$. Poleg tega smete klicati `is_clockwise` največ 1 000 000 krat za vsak testni primer.

podnaloge	omejitve	točke
1	$n \leq 50$	15
2	$n \leq 500$	20
3	$n \leq 15\,000$	20
4	največ ena točka ni na robu konveksne ovojnice	20
5	brez dodatnih omejitev	25

Poiskovanje na vašem računalniku

V imeniku `public` imate testno knjižnico, ki vam omogoča testiranje formalne pravilnosti vaše rešitve. Knjižnica prebere opis Bajtije s standardnega vhoda v naslednji obliki:

- v prvi vrstici je n , število mest;
- naslednjih n vrstic vsebuje po dve celi števili, ki predstavljata koordinate i -tega mesta.

Testna knjižnica **ne testira** pravilnosti vaše rešitve. Knjižnica tudi ne preverja pravilnosti vhodnih podatkov. Vsekakor pa je različna od (tajne) knjižnice na strežniku.

Primer vhodnih podatkov za testno knjižnico je v datoteki `tri0.in`.

Po klicu `give_answer` knjižnica izpiše na standardni izhod odgovor in število klicev funkcije/metode `is_clockwise`.

Za prevajanje vaše rešitve z uporabo testne knjižnice lahko uporabite naslednja ukaza:

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

Pri Javi za prevajanje s testno knjižnico (razredom) ne potrebujete nobenega posebnega ukaza.

Datoteke z vašo rešitvijo in knjižnico naj bi bile v istem imeniku.