

Tapşırıq: TRI

Triangles - Üçbucaqlar

CEOİ 2018, day 2. Məsələyə ayrılmış yaddaş: 256MB

Byteland $n(n \geq 3)$ şəhəri olan gözəl bir ölkədir. Bu şəhərlər 2D müstəvidə müxtəlif n nöqtə şəklində göstərilə bilər. Şəhərlər 1 dən n -ə kimin nömrələnmişdir. Siz orada turist olduğunuz üçün Bytelanddakı şəhərlərin dəqiq yerini bilmirsiniz. Bir turist jurnalından öyrəndiniz ki, burada heç bir üç şəhər kollinear deyil (bir düz xətt üzərində yerləşmir).

n sayda nöqtələr çoxluğu üçün qabarıq gövdə onların hamısını içində və ya kənarları üzərində saxlayan sahəsi ən kiçik qabarıq çoxbucaqlıdır. Qabarıq çoxbucaqlının bütün bucaqları 180 dərəcədən azdır və onun tərəfləri təpə nöqtələrindən başqa heç bir yerdə kəsişə bilməz.

Sizin tapşırığınız bu qabarıq çoxbucaqlının kənarları üzərində qalan Byteland şəhərlərinin sayını tapmaqdır. Bunun üçün siz üç **müxtəlif** şəhəri i, j, k ($1 \leq i, j, k \leq n$) ehtiva edən sorğular soruşa bilərsiniz. Belə bir sorğu i, j, k şəhərlərini (nöqtələrini) birləşdirən üçbucağı nəzərdə tutur. Bu tip sorğuya verilən cavab sizə nöqtələri verilmiş ardıcılıqla gəzdirdiyiniz zaman dönmənin saat əqrəbi istiqamətində yoxsa əksinə olduğunu bildirir.

Kitabxanadan istifadə

Sizin proqramınız sorğular soruşmaq və yekun cavabınızı qaytarmaq üçün lazım olan funksiyaları dəstəkləyən kitabxanadan istifadə etməlidir.

Kitabxana (C və C++ üçün **trilib.h**) aşağıdakı funksiyaları dəstəkləyir:

- **int get_n()** – Şəhərlərin sayını qaytarır
- **bool is_clockwise(int a, int b, int c)** – a, b, c nöqtələri saat əqrəbi istiqamətində üçbucaq əmələ gətirirsə **true**, əks halda **false** qaytarır.
- **void give_answer(int s)**

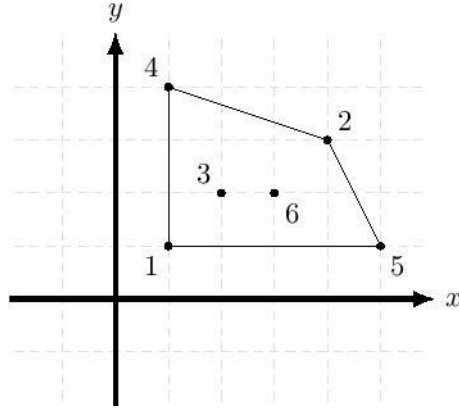
Proqramınız **give_answer** funksiyasını çağırdıqdan sonra daha sorğular soruşmağa icazə verilmir. **give_answer** funksiyasını yalnız bir dəfə çağırmalısınız.

Bu məsələdə siz standart input/outputdan istifadə edə bilməzsiniz. Proqramınız **give_answer** funksiyası çağırıldıqdan dərhal sonra dayanmalıdır.

Qəbul edə bilərsiniz ki, nöqtələrin yerləri qabaqcadan seçilmişdir və proqramın icrası zamanı dəyişməyəcək (yəni ki, kitabxana tamamilə deterministik davranır). Məsələn, aşağıdakı test nümunəsində **give_answer(4)** çağırılıb proqramı dərhal dayandırarsanız testdən keçəcəksiniz. Sizə cavabı dəqiq tapmadan təxmin etməyə icazə verilir.

Nümunə:

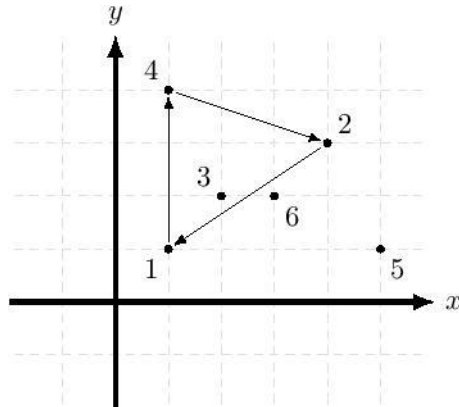
Aşağıdakı şəkildə göstərildiyi kimi $n = 6$ sayda $(1, 1)$, $(4, 3)$, $(2, 2)$, $(1, 4)$, $(5, 1)$, $(3, 2)$ nöqtələrində yerləşən şəhərlərdən ibarət nümunəni nəzərdən keçirək. Qabarıq çoxbucaqlı düz xəttlərlə işarələnmişdir. Onun kənarları üzərindəki nöqtələrin sayı dörd olduğundan, cavab 4 olur.



Aşağıdakı cədvəl yuxarıda verilmiş nümunəyə uyğun kitabxana ilə qarşılıqlı əlaqəni göstərir.

Çağırılmış funksiya	Qaytarılan dəyər
<i>get_n()</i>	6
<i>is_clockwise()</i>	true
<i>is_clockwise()</i>	true
<i>is_clockwise()</i>	false
<i>is_clockwise()</i>	true
<i>give_answer(4)</i>	-

Aşağıdakı şəkil birinci sorğudakı üçbucağı göstərir. 1, 4, 2 şəhərləri saat əqrəbi istiqamətində olduğu üçün qaytarılan dəyər **true** olur.



Qiymətləndirmə

Bütün testlər üçün $3 \leq n \leq 40\,000$ və *is_clockwise* funksiyasını maksimum 1 000 000 dəfə çağırmaq olar. Testlər əlavə məhdudiyyətlərlə aşağıdakı alt-tapşırıqlara ayrılıb. Bir alt-tapşırıqdakı testlər bir və ya bir neçə ayrı test qruplarından ibarətdir. Hər bir test qrupu bir və ya bir neçə test halından ibarət ola bilər.

Alt-tapşırıq	Məhdudiyyətlər	bal
1	$n \leq 50$	15
2	$n \leq 500$	20
3	$n \leq 15\,000$	20
4	<i>ən çoxu bir nöqtə qabarıq çoxbucaqlının kənarı üzərində olmaya bilər</i>	20
5	<i>Ümumi məhdudiyyətlər</i>	25