

Code Analysis:

`main`

Loop εκτέλεσης:

1. **Έλεγχος απλότητας:** Αν το P δεν είναι simple, τερματίζει..
2. **Έλεγχος y-μονοτονίας:** Αν δεν είναι y-μονοτονικό, τερματίζει..
3. **Τριγωνοποίηση:** Καλεί την συνάρτηση `triangulate_monotone_P`, τυπώνει τα τρίγωνα στο τερματικό.
4. **Οπτικοποίηση:** Καλεί την συνάρτηση `draw_polygon_and_triangles`.

5. **In a loop:**

Ρωτάει το χρήστη αν θέλει νέο πολύγωνο για τριγωνοποίηση.

Αν ναι, διαβάζει το πλήθος κορυφών και τις συντεταγμένες, επαναλαμβάνει τους ελέγχους 1,2.

Σε λάθος είσοδο διακόπτει ή ζητά νέα εισαγωγή συντεταγμένων.

`check_if_y_monotone(P)` // P: list κορυφών (x,y).

Ελέγχει αν το πολύγωνο είναι **y-μονοτονικό**, δηλ. κάθε οριζόντια γραμμή τέμνει το σύνορο του P το πολύ σε δύο σημεία.

Εντοπίζει την κορυφή με maximum y (`i_max`) και με minimum y (`i_min`).

Δημιουργεί δύο “αλυσίδες”- σύνορα :

αριστερή: ακολουθεί τους “pointers” από `i_max` προς `i_min` με αύξουσα σειρά.

δεξιά: από `i_max` προς `i_min` με φθίνουσα.

Ελέγχει κάθε αλυσίδα ξεχωριστά ότι οι τιμές y φθίνουν σε κάθε βήμα.

Αν και οι δύο φθίνουν, το πολύγωνο είναι y-μονοτονικό (True), αλλιώς όχι (False).

is_simple_polygon(P) // P: list κορυφών (x,y).

Ελέγχει αν ένα πολυγωνικό περίγραμμα χωρίς επαναλαμβανόμενες ακμές είναι **simple** (δεν αυτοτομείται).

Για κάθε ακμή p_1q_1 (in a loop διατρέχει όλες τις κορυφές), συγκρίνει με κάθε άλλη p_2q_2 .

Αγνοεί ακμές που έχουν κοινές κορυφές ή είναι adjacent.

Αν εντοπίσει **segments_intersect(p1,q1,p2,q2)**, τότε επιστρέφει False.

Αν δεν βρεθεί καμία τομή, επιστρέφει True.

Comment: Το **%n** διασφαλίζει τον κυκλικό τρόπο λειτουργίας (η τελευταία κορυφή θα ξαναενωθεί με την 1η). Έτσι ακριβώς λειτουργεί η κυκλική ουρά στις δομές.

segments_intersect(p1, q1, p2, q2)

Ελέγχει αν (p_1q_1) και (p_2q_2) τέμνονται.

Υπολογίζει τις orientation τιμές:

$o_1 = \text{orientation}(p_1, q_1, p_2)$

$o_2 = \text{orientation}(p_1, q_1, q_2)$

$o_3 = \text{orientation}(p_2, q_2, p_1)$

$o_4 = \text{orientation}(p_2, q_2, q_1)$

General περίπτωση τομής:

Αν o_1 και o_2 : διαφορετικά πρόσημα **και** o_3 και o_4 : διαφορετικά πρόσημα
==> υπάρχει τομή. (Returns True, else returns False)

on_segment(p, q, r) // $p=(p_x,p_y)$, $q=(q_x,q_y)$, $r=(r_x,r_y)$

Ελέγχει αν το σημείο **q** βρίσκεται πάνω στο ευθύγραμμο τμήμα **pr**.

(1) Ελέγχει αν x-coordinate του q είναι μεταξύ των x-coordinates p και r.

(2) Ελέγχει αν y-coordinate του q είναι μεταξύ των y-coordinates p και r.

Επιστρέφει **True** μόνο αν ισχύουν και τα δύο((1) & (2))

orientation(p,q,r) // $p=(p_x,p_y)$, $q=(q_x,q_y)$, $r=(r_x,r_y)$

Υπολογίζει αν η διαδρομή $p \rightarrow q \rightarrow r$ κάνει αριστερή ή δεξιά στροφή.

(1) Υπολογίζει το διάνυσμα $p \rightarrow q$ και το διάνυσμα $p \rightarrow r$.

(2) Το cross product των δύο διανυσμάτων δίνει:

(2.1) θετικό αν είναι αριστερή στροφή (Counter clock wise)

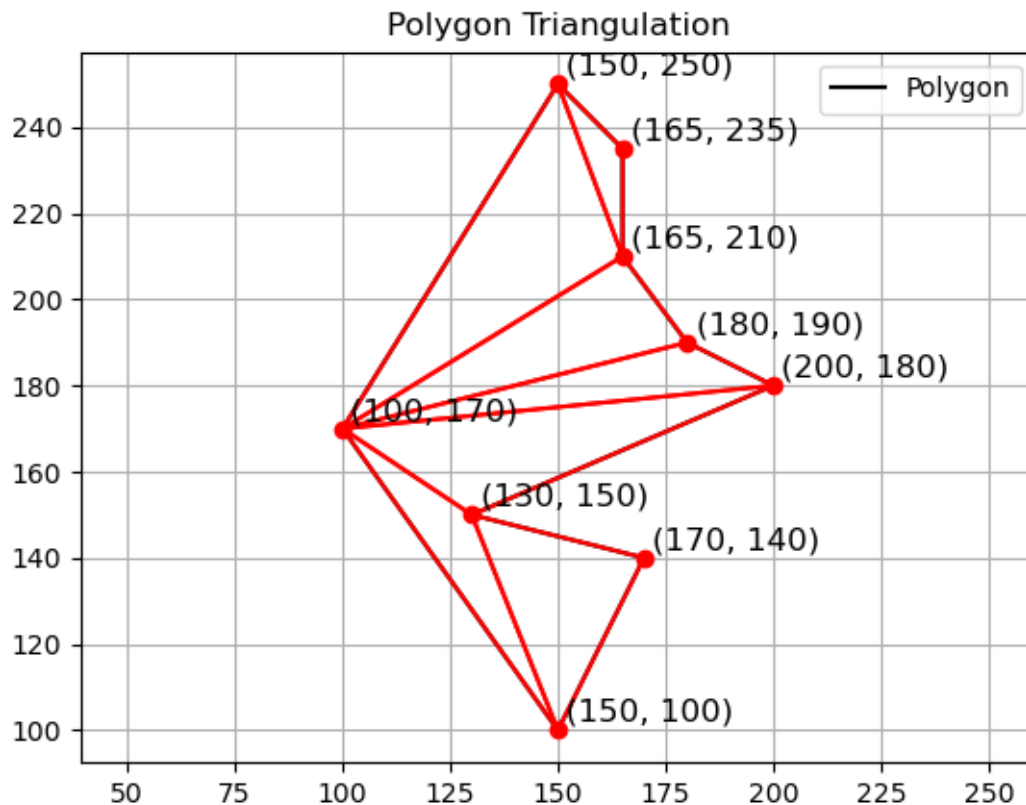
(2.2) αρνητικό αν είναι δεξιά (Clock wise),

(2.3) 0 αν είναι collinear.

Επιστρέφει: τον αριθμό του cross product

Run the code:

```
jim@jim-desktop:~/CompGeo$ python3 triangulation_matplot.py
===== Program to Triangulate simple polygons. =====
Triangles each one contains 3 vertices):
((150.0, 250.0), (165.0, 235.0), (165.0, 210.0))
((100.0, 170.0), (200.0, 180.0), (180.0, 190.0))
((100.0, 170.0), (180.0, 190.0), (165.0, 210.0))
((100.0, 170.0), (165.0, 210.0), (150.0, 250.0))
((130.0, 150.0), (100.0, 170.0), (200.0, 180.0))
((130.0, 150.0), (170.0, 140.0), (150.0, 100.0))
((100.0, 170.0), (130.0, 150.0), (150.0, 100.0))
```



Do you want to insert a new Polygon? (Yes/No): y

How many vertices do you want for P? 4

The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 1 (x y): 1 0

The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 2 (x y): 10 10

The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 3 (x y): 20 0

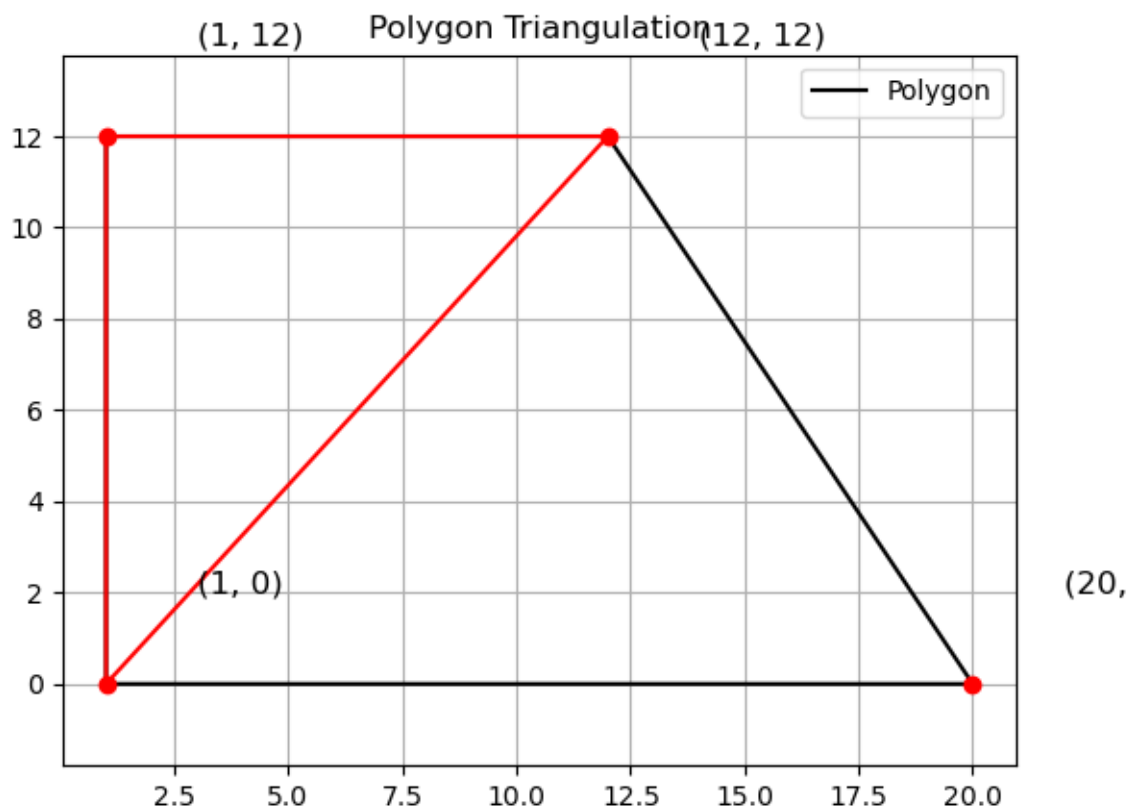
The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 4 (x y): 10 -10

Triangles each one contains 3 vertices):

((20.0, 0.0), (1.0, 0.0), (10.0, 10.0))

((10.0, -10.0), (20.0, 0.0), (1.0, 0.0))



Do you want to insert a new Polygon? (Yes/No): y

How many vertices do you want for P? 3

The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 1 (x y): 20 0

The coordinates should be given in Counter Clock wise or Clock wise turn!

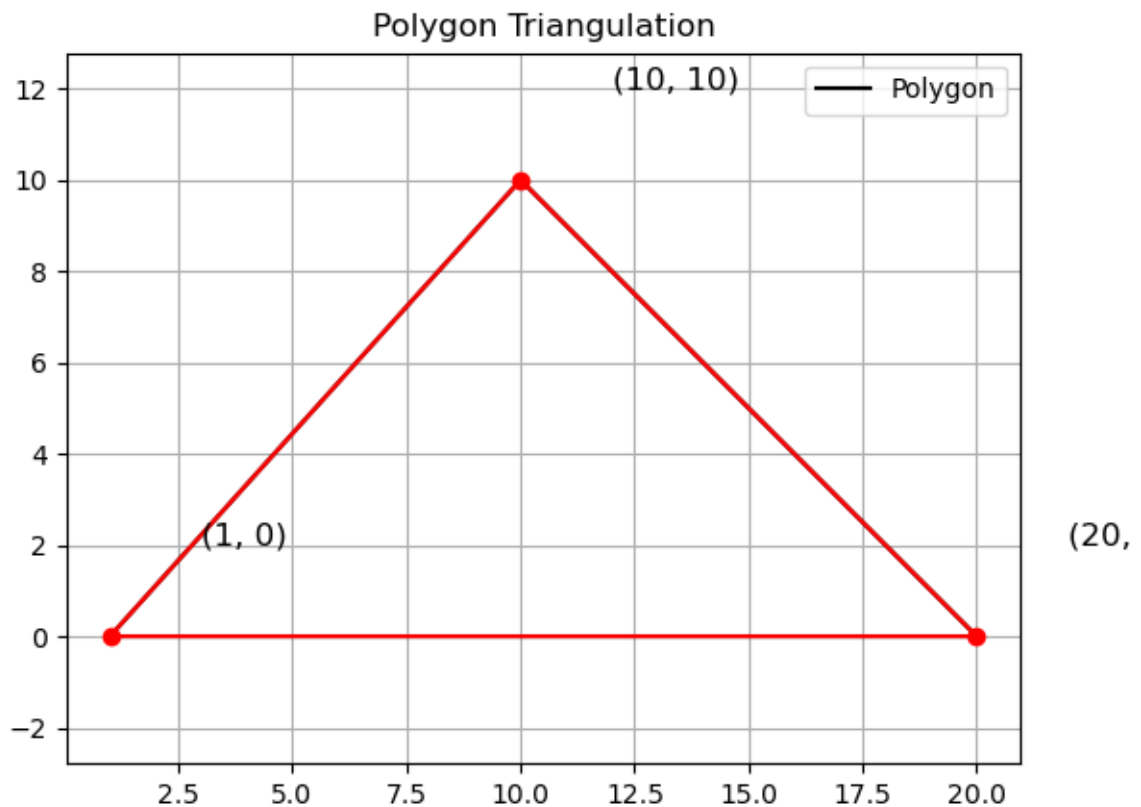
Please give coordinates 2 (x y): 10 10

The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 3 (x y): 1 0

Triangles each one contains 3 vertices):

((20.0, 0.0), (1.0, 0.0), (10.0, 10.0))



Do you want to insert a new Polygon? (Yes/No): y

How many vertices do you want for P? 4

The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 1 (x y): 1 0

The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 2 (x y): 10 10

The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 3 (x y): 20 0

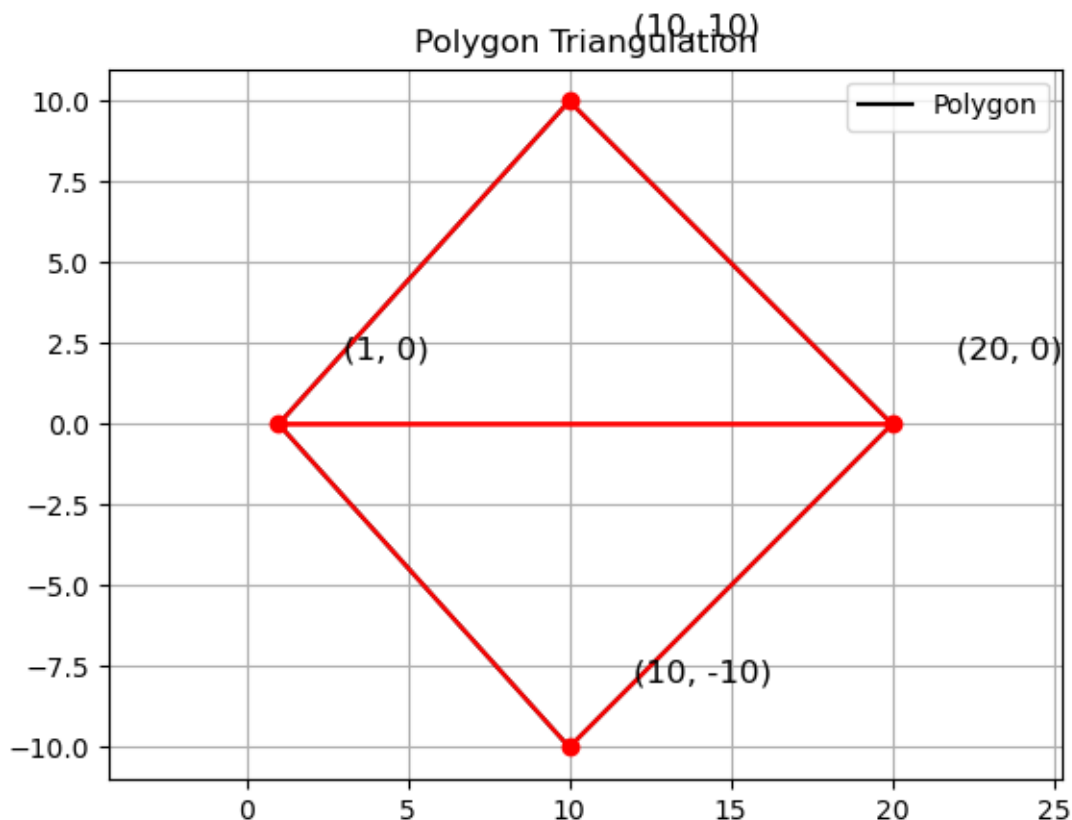
The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 4 (x y): 10 -10

Triangles each one contains 3 vertices):

((20.0, 0.0), (1.0, 0.0), (10.0, 10.0))

((10.0, -10.0), (20.0, 0.0), (1.0, 0.0))



Do you want to insert a new Polygon? (Yes/No): Yes

How many vertices do you want for P? 5

The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 1 (x y): 5 0

The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 2 (x y): 10 10

The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 3 (x y): 20 10

The coordinates should be given in Counter Clock wise or Clock wise turn!

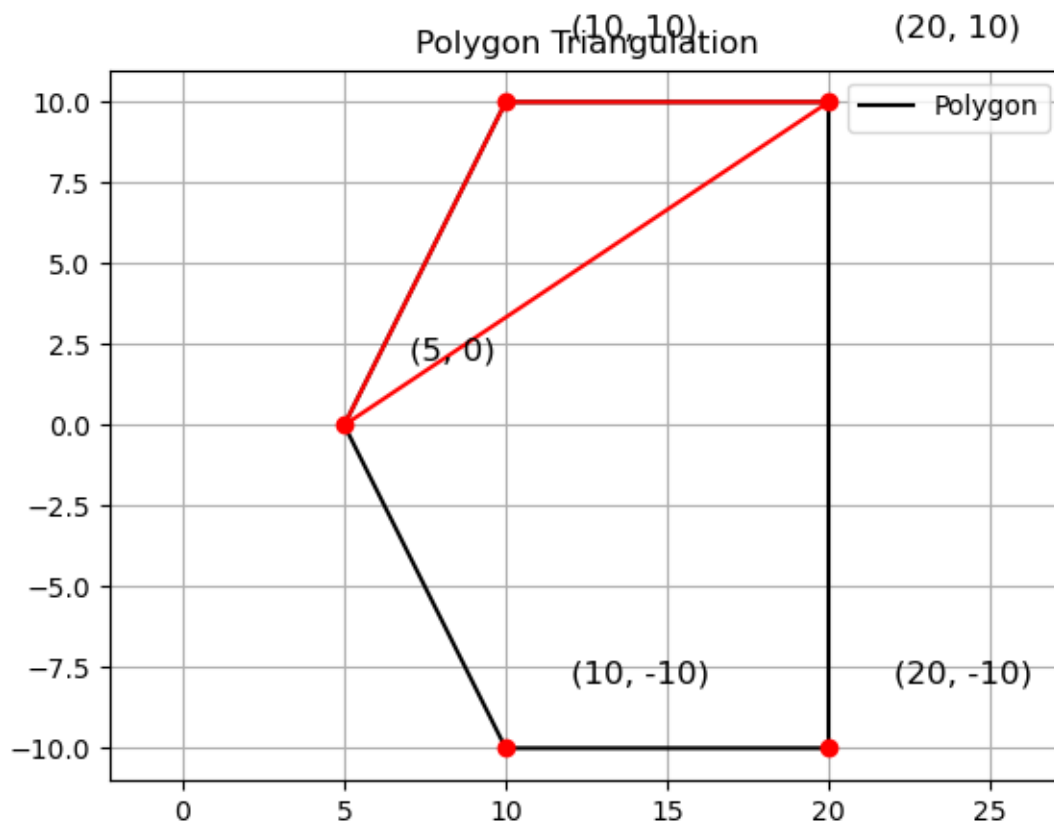
Please give coordinates 4 (x y): 20 -10

The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 5 (x y): 10 -10

Triangles each one contains 3 vertices):

((5.0, 0.0), (20.0, 10.0), (10.0, 10.0))



Do you want to insert a new Polygon? (Yes/No): Yes

How many vertices do you want for P? 5

The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 1 (x y): 5 0

The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 2 (x y): 10 10

The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 3 (x y): 20 10

The coordinates should be given in Counter Clock wise or Clock wise turn!

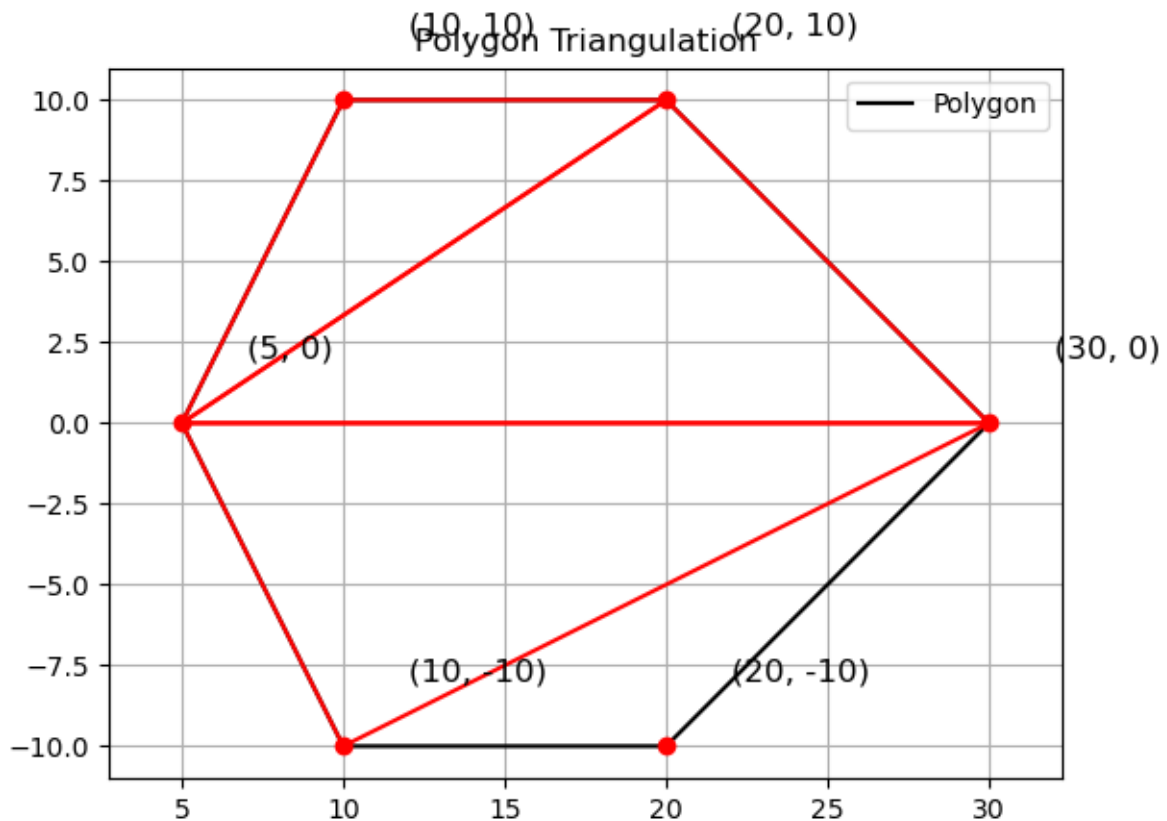
Please give coordinates 4 (x y): 20 -10

The coordinates should be given in Counter Clock wise or Clock wise turn!

Please give coordinates 5 (x y): 10 -10

Triangles each one contains 3 vertices):

((5.0, 0.0), (20.0, 10.0), (10.0, 10.0))



Note: Σχετικά με την λειτουργία του κώδικα, ο χρήστης πρέπει να δίνει κορυφές με συγκεκριμένο τρόπο.

- (1) Να σχηματίζουν απλό πολύγωνο. Δηλαδή να μην υπάρχουν αυτοτεμνόμενες ακμές.
- (2) Το πολύγωνο πρέπει να είναι y-μονότονο.
- (3) Χωρίς επαναλήψεις κορυφών (δηλ. η τελευταία κορυφή δεν πρέπει να είναι ίδια με την 1η)

Οι συνθήκες (1), (2), (3) ελέγχονται στον κώδικά.

- (4) Με Counter Clockwise 'η Clockwise τρόπο (not random), διαφορετικά δεν θα δωθεί σωστό output(Improvement to do!)