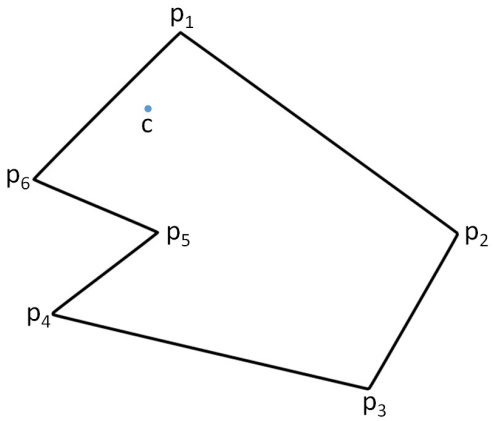
Country recognition

You are given a text file containing a list of countries, together with a description of their borders. Each line of the file contains the name of a country, followed by a tab and a list of an even number of real values that are separated by spaces. The list of $2n$ real values describes a polygon having n successive vertices (listed in clockwise or counterclockwise order) on the border of the country. Each point p on Earth is described by a tuple (p_b, p_l), where $p_b \in \mathbb{R}$ indicates the latitude and $p_l \in \mathbb{R}$ indicates the longitude of the point. Accordingly, each point (p_b, p_l) on the border of a country is described by two successive numbers $p_b, p_l \in \mathbb{R}$ in the list of real values. Below we give an example of a small portion of such a text file.

Afghanistan 37.24 74.92 37.18 74.39 37.03 74.57 36.82 72.56 36.13 71.24 ...
Albania 41.02 19.44 41.80 19.60 41.85 19.37 42.62 19.65 42.56 20.07 ...
Algeria 36.80 2.96 36.89 4.79 36.64 5.33 37.09 6.40 36.94 8.62 ...
American Samoa -14.30 -170.54 -14.29 -170.56 -14.28 -170.54 -14.30 -170.54 ...
Andorra 42.57 1.78 42.51 1.73 42.60 1.45 42.57 1.78
...

Given a text file that is formatted as described above, your task is to determine the country for each given point on Earth. If we assume that a polygon describing the border of a country is plane figure, all we need to able to do is determine whether or not a given point $c = (c_b, c_l)$ is inside or outside the given n-gon (a polygon with n sides) with successive vertices $p_1, p_2, \ldots, p_n$.



This can be done by counting the **number** of edges of the polygon for which the following two conditions are **both** fulfilled: $$\begin{cases}
c_l < p_l \textrm{ or } c_l < q_l \textrm{ (but not both)}\\
c_b - q_b < \dfrac{p_b - q_b}{p_l - q_l}(c_l - q_l)
\end{cases}$$ Herewith, each edge is represented by two successive vertices $p = (p_b, p_l)$ and $q = (q_b, q_l)$, and the vertices of the polygon are iterated consistently in clockwise or counterclockwise order. If an odd number of edges fulfills the above conditions, the point c is located inside the polygon. Otherwise it is located outside the polygon.

Assignment

Define a class Polygon that can be used to represent planar polygons in Python. This class must support at least the following methods:

- An initialisation method that takes a list of points in the two-dimensional plane. These points represent the successive vertices of a polygon (in clockwise or counterclockwise order). Each point is described by a tuple $(withx, y)$, $x, y \in \mathbb{R}$.
- A method `contains` that takes a point in the two-dimensional plane. The method must return a Boolean value that indicates whether or not the given point is inside the polygon.

Use the class `Polygon` to define a class `WorldMap`. An object of the class `WorldMap` represents a list of countries, where the border of each country is represented as a planar polygon. The class `WorldMap` must support at least the following methods:

- An initialisation method that takes the location of a text file. This file must contain a list of countries and the description of their borders, using the format as outlined in the introduction of this assignment.
- A method `borders` that takes the name of a country. The method must return a planar polygon (an object of the class `Polygon`) that describes the border of the given country, as defined in the given text file. Herewith, the name of the country must exactly match the name of one of the countries in the given text file. If this is not the case, the method must raise an `AssertionError` with the message `unknown country`.
- A method `country` that takes the location of a point on Earth (a tuple). Based on the given text file, the method must determine in which country this point is located, and return the name of that country. In case the given point is not inside any country contained in the text file, the method must return the value `None`.

Example

In the following interactive session, we assume that the text file countries.txt is located in the current directory.

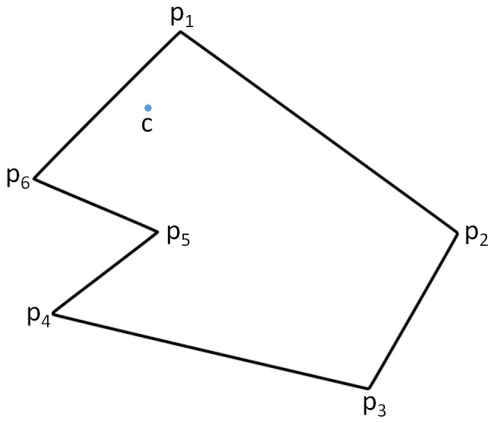
```
>>> polygon = Polygon([(0, 0), (0, 1), (1, 1), (1, 0)])
>>> polygon.contains((0.5, 0.5))
True
>>> polygon.contains((1.5, 0.5))
False
>>> polygon.contains((0.5, 1.5))
False
>>> polygon.contains((-0.5, 0.5))
False
>>> polygon.contains((0.5, -0.5))
False

>>> map = WorldMap('countries.txt')
>>> map.border('Belgium').contains((50.5, 4.2))   # Brussels
True
>>> map.border('France').contains((50.5, 4.2))    # Brussels
False
>>> map.border('Legoland')
Traceback (most recent call last):
AssertionError: unknown country
>>> map.country((50.5, 4.2))     # Brussels
'Belgium'
>>> map.country((45.25, -75.42)) # Ottawa
'Canada'
>>> map.country((-17.5, 31.03))  # Harare
'Zimbabwe'
```

```
>>> map.country((-16.3, -68.09)) # La Paz
'Bolivia'
>>> map.country((-10.0, -20.0)) # ???
```

Gegeven is een tekstbestand met een lijst van landen, samen met de omschrijving van hun landsgrenzen. Elke regel van het bestand bevat de naam van een land, gevolgd door een tab en een lijst van een even aantal reële getallen die van elkaar gescheiden worden door spaties. Hierbij stelt de lijst van $2n$ reële getallen een veelhoek voor die gevormd wordt door n opeenvolgende punten (in wijzerzin of tegenwijzerzin) op de grens van het land. Elk punt p op Aarde wordt beschreven door een tuple (p_b, p_l), waarbij $p_b \in \mathbb{R}$ de breedtegraad en $p_l \in \mathbb{R}$ de lengtegraad van het punt aangeeft. Overeenkomstig hiermee wordt elk punt (p_b, p_l) op de grens van een land, in de lijst van getallen uit het bestand beschreven door twee opeenvolgende getallen $p_b, p_l \in \mathbb{R}$. Hieronder staat alvast een voorbeeld van een deel van zo een tekstbestand.

```
Afghanistan 37.24 74.92 37.18 74.39 37.03 74.57 36.82 72.56 36.13 71.24 ...
Albania 41.02 19.44 41.80 19.60 41.85 19.37 42.62 19.65 42.56 20.07 ...
Algeria 36.80 2.96 36.89 4.79 36.64 5.33 37.09 6.40 36.94 8.62 ...
American Samoa -14.30 -170.54 -14.29 -170.56 -14.28 -170.54 -14.30 -170.54 ...
Andorra 42.57 1.78 42.51 1.73 42.60 1.45 42.57 1.78
...
```

Je opdracht bestaat erin om op basis van het gegeven tekstbestand, voor elk gegeven punt op Aarde te bepalen in welk land het ligt. Als we ervan uitgaan dat een veelhoek die een landsgrens omschrijft een vlakke figuur is, dan moeten we dus kunnen bepalen of een gegeven punt $c = (c_b, c_l)$ binnen of buiten een gegeven n-hoek met opeenvolgende hoekpunten $p_1, p_2, \ldots, p_n$ ligt.



Dit kunnen we doen door het **aantal** zijden van de veelhoek te tellen waarvoor de volgende twee voorwaarden **beide** voldaan zijn: $$\begin{cases}
c_l < p_l \textrm{ of } c_l < q_l \textrm{ (maar niet beide)}\\
c_b - q_b < \dfrac{p_b - q_b}{p_l - q_l}(c_l - q_l)
\end{cases}$$ Hierbij wordt elke zijde bepaald door twee opeenvolgende punten $p = (p_b, p_l)$ en $q = (q_b, q_l)$, en worden de hoekpunten van de veelhoek consistent in wijzerzin of in tegenwijzerzin afgelopen. Als het aantal zijden van de veelhoek dat hieraan voldoet oneven is, dan ligt het punt c binnen de veelhoek. Anders ligt het erbuiten.

Opgave

Definieer een klasse `Veelhoek` waarmee vlakke veelhoeken kunnen voorgesteld worden in

Python. Deze klasse moeten minstens de volgende methoden ondersteunen:

- Een initialisatiemethode waaraan een lijst van punten in het twee-dimensionale vlak moet doorgegeven worden. Deze punten stellen de opeenvolgende hoekpunten van een veelhoek voor (in wijzerzin of in tegenwijzerzin). Elk punt wordt voorgesteld door een tuple (x, y), waarbij $x, y \in \mathbb{R}$.
- Een methode `bevat` waaraan een punt uit het vlak moet doorgegeven worden. De methode moet een Booleaanse waarde teruggeven, die aangeeft of het gegeven punt al dan niet binnen de veelhoek ligt.

Gebruik de klasse `Veelhoek` om een klasse `Wereldkaart` te definiëren. Een object van de klasse `Wereldkaart` stelt een lijst van landen voor, waarbij de landsgrens van elk land wordt voorgesteld door een vlakke veelhoek. De klasse `Wereldkaart` moeten minstens de volgende methoden ondersteunen:

- Een initialisatiemethode waaraan de locatie van een tekstbestand moet doorgegeven worden. Dit bestand moet een lijst van landen en de omschrijving van hun landsgrenzen bevatten, in het formaat zoals uiteengezet in de inleiding van deze opgave.
- Een methode `grens` waaraan de naam van een land moet doorgegeven worden. De methode moet een veelhoek teruggeven (een object van de klasse `Veelhoek`) die de landsgrens omschrijft van het gegeven land, zoals omschreven in het gegeven tekstbestand. Hierbij moet de naam van het land exact overeenkomen met één van de namen van landen in het gegeven tekstbestand. Indien dit niet het geval is, dan moet de methode een `AssertionError` opwerpen met de boodschap `onbekend land`.
- Een methode `land` waaraan een punt op Aarde (een tuple) moet doorgegeven worden. De methode moet op basis van het gegeven tekstbestand bepalen in welk land dit punt gelegen is, en de naam van dit land als resultaat teruggeven. Indien het gegeven punt in geen enkel land ligt dat in het tekstbestand bevat zit, dan moet de methode de waarde `None` teruggeven.

Voorbeeld

Bij onderstaande voorbeeldsessie gaan we ervan uit dat het tekstbestand landen.txt zich in de huidige directory bevindt.

```
>>> veelhoek = Veelhoek([(0, 0), (0, 1), (1, 1), (1, 0)])
>>> veelhoek.bevat((0.5, 0.5))
True
>>> veelhoek.bevat((1.5, 0.5))
False
>>> veelhoek.bevat((0.5, 1.5))
False
>>> veelhoek.bevat((-0.5, 0.5))
False
>>> veelhoek.bevat((0.5, -0.5))
False

>>> kaart = Wereldkaart('landen.txt')
>>> kaart.grens('Belgium').bevat((50.5, 4.2))   # Brussel
True
>>> kaart.grens('France').bevat((50.5, 4.2))    # Brussel
False
>>> kaart.grens('Legoland')
```

```
Traceback (most recent call last):
AssertionError: onbekend land
>>> kaart.land((50.5, 4.2))      # Brussel
'Belgium'
>>> kaart.land((45.25, -75.42)) # Ottawa
'Canada'
>>> kaart.land((-17.5, 31.03))  # Harare
'Zimbabwe'
>>> kaart.land((-16.3, -68.09)) # La Paz
'Bolivia'
>>> kaart.land((-10.0, -20.0)) # ???
```