**Final Project**

**Background**

The acronym API is short for “Application Programming Interface.” It is usually a collection of functions, methods or classes that supports the interaction of an application program (the program you write as a developer) with other programs that run on a remote server. Google has several API's available for developers [(https://developers.google.com/)](https://developers.google.com/apis-explorer/) to query information from services google provides such as google map. We will be using the map API for purposes of finding distances between cities in the U.S. known as Google Map Distance Matrix API [(https://developers.google.com/maps/documentation/distance-matrix/)](https://developers.google.com/maps/documentation/distance-matrix/). It is based on creating a query using the http protocol of the web. Essentially, you send a web address (with the query embedded in the web address) and a web page is returned with the requested information, albeit in a rather odd form (It is not HTML format).

The general form of a distance query is:

[http://maps.googleapis.com/maps/api/distancematrix/*json*?*parameters*](http://maps.googleapis.com/maps/api/distancematrix/json?parameters)

where parameters denotes a series of parameters. The parameters we will use are:

* origins: the origin city and state[[1]](#footnote-2)
* destinations: the destination city and state
* sensor: false, which tells the server that your application does not use a GPS locator
* mode: one of ‘driving’, ‘biking, or ‘walking’

There are no spaces in the query. Parameter assignments are separated using the ampersand (&) symbol and words in the city are separated using the plus (+) sign.

Some examples are listed next. Copy and paste each one into a browser to see what is returned. Then create your own examples to see how changing parameter values changes the response you get back. What happens, for instance, if you make up some city names and/or state abbreviations? What happens if there is not roadway between the origin and destination?

<http://maps.googleapis.com/maps/api/distancematrix/json?origins=New+York+NY&destinations=Lansing+MI&mode=driving&sensor=false>

<http://maps.googleapis.com/maps/api/distancematrix/json?origins=Lansing+MI&destinations=Sacramento+CA&mode=bicycling&sensor=false>

**Parsing the Result**

As distance calculation is part of the map API, and not its own API, the entire result is returned as a single string, which must be parsed. The string returned by the first query above, which finds the driving distance between New York, NY and Lansing, MI, is as follows:

{

"destination\_addresses" : [ "Lansing, MI, USA" ],

"origin\_addresses" : [ "New York, NY, USA" ],

"rows" : [

{

"elements" : [

{

"distance" : {

"text" : "1,092 km",

"value" : 1091921

},

"duration" : {

"text" : "10 hours 12 mins",

"value" : 36711

},

"status" : "OK"

}

]

}

],

"status" : "OK"

}

You will need to parse this string to extract the information required. In this response, you will need to extract the value 1091921, which is the number of meters from Lansing to New York.

In testing queries, you should have discovered that the server “guesses” a location if it cannot exactly match the values indicated for the cities. Your program does not need to check that the response is for the correct location—just assume that the distance returned in a response is correct. But your program *does* need to handle the situation where the response does not contain a distance. (See below.)

**Project Description / Specification**

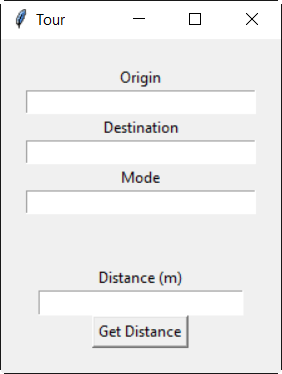
Create a class called Tour. An instance of this class is to be instantiated with two US cities, and is used to fetch information from the web. Specifically, the class must define:

1. \_\_init\_\_ The constructor takes two strings as arguments, each giving a city name and state abbreviation, indicating the origin and destination. For example: Tour(“New York, NY”, “Los Angeles, CA”)represents a tour that starts in New York city and ends in Los Angeles.
2. distance This method takes a single (optional) argument indicating a mode – one of the strings ‘driving’ (default), ‘biking, or ‘walking’. It returns the total distance (in meters) covered by the tour for the indicated mode. This method is where you will use urllib functions to get data from the web to find the distances between two locations in the tour and calculate the total distance. If a response does not contain a distance value, the method should raise a ValueError exception.

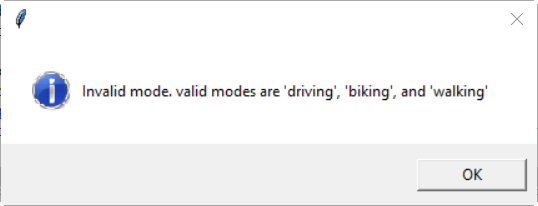
**Graphical User Interface Description:**

Create a class called TourGui. An instance of this class is to be instantiated in the main. Specifically the class must define:

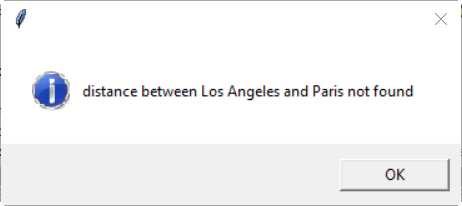
1. \_\_init\_\_ The constructor doesn’t take any arguments. The constructor creates two frames in the window, a top frame and bottom frame. The top frame contains a label and an entry widget for the following fields: origin, destination, and mode. The bottom frame contains a label and text widget for Distance and a button widget for Get Distance. The GUI should like the following:



1. onClick This method is the event handler for the Get Distance Button. When the button is clicked, the origin, destination and mode fields are read, the query is made and the distance between the origin and the destination is displayed. If the user enters an invalid mode, a message box shows up indicating invalid mode was entered:



If the distance was not found between the origin and the destination, a message box shows up indicating the distance was not found:



**Deliverables**

Create a zipfile of the project folder and submit on Canvas. You must name your project as ‘FirstnameLastnameFinalProject”.

1. The parameter names are plural because you can request a distance matrix, indicating distances between a set of potential origins and a set of potential destinations. However, we will just use a single city for each. [↑](#footnote-ref-2)