DT265/B CMPU4060: Lab 6
DIT School of Computing | Dr Andrew Hines

Instructions

Before you begin, take out a pen and paper. Put a title of Lab 6 on it and the date. Based on the information given below write out the function(s) you will code before you start writing any code. What parameters will it/they take? What value(s) will be returned? What libraries will you use? How will you test your function in using a driver (`main()`) function?

You will have more time in the next lab to work on Exercises 2 – 4. **Submit your your program by next Tuesday via webcourses.**

Exercises

1. (*4 points*) In this lab you will develop code to calculate the distance between two points on earth using a direct route or "as the crow flies". The first we need to know how to calculate the distance between two points on Earth, given their co-ordinates (latitude and longitude). As a slight simplification, we will assume the earth is round (as opposed to geoid). Latitude is a measure of degrees north (positive) or south (negative) from the equator. 0° is at the equator, +90° is the north pole and −90° is the south pole. Longitude is a measure of degrees east (positive) or west (negative) of the prime meridian located at Greenwich, England.
As an example DIT, Kevin St. has a location of 53.338 (North), -6.2676448 (West).
The formula for the distance, d, between position 1 and 2 is

$$d_{(1,2)} = arccos(sin\phi_1 sin\phi_2 cos(\theta_1 - \theta_2) + cos\phi_1 cos\phi_2) * r_{earth} \quad (1)$$

ϕ is the angle from pole to position, meaning $90 - latitude$ for points north of the equator and $90 + latitude$ for points south of the equator.
θ is the number of degrees east (positive) or west (negative) from the prime meridian (at Greenwich).
The spherical radius of the earth (r_{earth}) can be approximated as 6371 km.
When carrying out calculations on the angles, the latitude and longitude angles should be converted from degrees into radians. So,

$$\phi_{radians} = \phi^\circ \times \frac{2\pi}{360^\circ} \quad (2)$$

1. You will need to import the math module to get access to constants (e.g. `math.pi`) and trigonometric functions (e.g. `sin, cos, acos`)
2. You should write a function that takes the latitude and longitude for 2 positions and returns the distance between them in kilometres, i.e.
3. You can test your code using the example airport co-ordinates given in Table 1
4. Round the answers to the nearest km. See Table 2 for distances calculated using the formula given and rounded to the nearest km by casting as an `int`

Table 1: Sample Airport Co-ordinates

Code	Airport	Country	Latitude	Longitude
DUB	Dublin	Ireland	53.421333	-6.270075
LHR	Heathrow	United Kingdom	51.4775	-0.461389
JFK	John F Kennedy Intl	United States	40.639751	-73.778925
AAL	Aalborg	Denmark	57.092789	9.849164
CDG	Charles De Gaulle	France	49.012779	2.55
SYD	Sydney Intl	Australia	-33.946111	151.177222

Table 2: Distance between airports (to nearest km)

	JFK	AAL	CDG	SYD	LHR	DUB
JFK	0	5966	5833	16013	5539	5103
AAL	5966	0	1021	16140	913	1096
CDG	5833	1021	0	16944	347	784
SYD	16013	16140	16944	0	17020	17215
LHR	5539	913	347	17020	0	448
DUB	5103	1096	784	17215	448	0

2. (*2 points*) Write a function to find the distance between all airports in Table 1. Read the notes below (titled **Aside: More About Lists**) if you need some help storing the data from Table 1 in your program.

3. (*2 points*) Give two airport codes, calculate the distance between them. Write a function that returns the distance between two airports that takes two codes as input parameters (e.g. `getDistanceBetweenAirports(airportCode1,airportCode2)`). Use lists to store the values in the test table.

4. (*2 points*) Rewrite your function, `getDistanceBetweenAirports`, to run without any loops. You might need to store your data in a different way that allows you to lookup a value given a name.

Aside: More About Lists

A list is a stored sequence of items

- We saw lists of numbers last week
  - e.g. `oddNumbers = [1, 3, 5, 7]`
- They can also contain other types (e.g. strings) of data or even mixtures of items
  - `names = ['Frank', 'Bob', 'Alice']`
  - `miscStuff = [15, 'Bob', -3.14, True]`
- Like loop ranges they are zero indexed

We can carry out a many kinds of operations on lists

```
miscStuff = [15, 'Bob', -3.14, True]
```

0	15
1	'Bob'
2	-3.14
3	True

- Access values in lists via their index, e.g. `myName = miscStuff[1]`
- Find the number of items in a list **`len`**`(miscStuff)` will return 4
- Create a counter for a loop using **`enumerate`**`(miscStuff)` (example in Listing 1)

Listing 1: Creating a numeric index and a code value as you iterate through a list

```
codes=['DUB','LHR','SYD']

for idx1, code1 in enumerate(codes):
    print(idx1, 'of codes is', code1)
```

Output of Listing 1 will be:

```
0 of codes is DUB
1 of codes is LHR
2 of codes is SYD
```

Read more about lists in chapter 14 of Johnson textbook (p.75)