## Weekly Test – 06

## **Mini Project**

You are given a Python program that reads the data from two CSV files named with capitals.csv and universities.csv (CSV → Comma Separated Values).

- 1. **Capitals** file contains the various columns saying Country Name, capital, latitude, longitude, country code and continent.
  - a. Country Name represents the name of the country
  - b. Capital represents the capital of the country
  - c. Latitude and Longitude represents the location of the country.
  - d. Country code represents the short form of a code for each country.
  - e. Continent represents the geographical regions of the country.
- 2. Universities file contains the various columns saying Code, Institute Name, country, National Rank, Degrees Offered, Average cost and score.
  - a. Code represents the short form for each institute.
  - b. Institute Name represents the University name.
  - c. Country represents where the university is located.
  - d. National Rank represents the rank of the institute national wide.
  - e. Degree Offered represents what are all the courses Institute is offering.
  - f. Average cost represents the fee of the institute.
  - g. Score represents the max obtained by each institute.

Read both files, process the data and provides various functions to analyze university/institute information based on different criteria. Refer the file named with rank.py this file provides the skeleton of the code and rest of the logic needs to write.

- 1. 'readFiles (UniFileName, capitalsFileName)' Read both files and prepare the required data to process further.
  - a. UniFileName → Input file name/location of the file with name (universities.csv) as the first parameter.
  - b. capitalsFileName  $\rightarrow$  Input file name/location of the file with name (capital.csv) as the second parameter.
- 2. 'findCountryByName (countryName, countries)', find and return the country details by name.
  - a. countryName → Country name is the input parameter and countries is the countries which are available.
- 3. 'getAllCodes (allUnivs)', find and return all the codes of institutes/universities.
- 4. 'getDistinctCountries (allUnivs)', find and return a set of all country names available only in allUnivs dictionary. Not looking at all countries from countries data file.

- 5. 'getDistinctContinents (allUnivs)', find and return set of all continents names available only in allUnivs dictionary. Not looking at all countries from countries data file.
- 6. 'getTopIntRank(countryName, allUnivs)', finds and return the university that has the highest international rank (World rank) within the selected country and return world rank and institute name for within the provided country.
  - a. countryName → Country name is the input parameter and allUnivs is all university/Institute.
- 7. 'getTopNatRank(countryName, allUnivs)', finds the university that has the highest national rank within the selected country and return national rank and institute name for the given country.
  - a. countryName → Country name is the input parameter and allUnivs is all university/Institute.
- 8. 'getAvgScore (countryName, allUnivs)', find and return the average score of the all the universities within the selected country. The value should be rounded with two decimal places.
  - a. countryName → Country name is the input parameter and allUnivs is all university/Institute.
- 'getRelativeScoreContinent(countryName, allUnivs)', find and return relative score to the continent. The ratio between the average score (calculated in question 8) divided by the highest score within the continent. The value should be rounded with two decimal places.
  - a. countryName → Country name is the input parameter and allUnivs is all university/Institute.
- 10. 'getUnivWithCapital (countryName, allUnivs)', find the institute/university code based user input country name.
  - a. countryName → Country name is the input parameter and allUnivs is all university/Institute.
- 11. 'studyInOnePlace (countryName, degrees, budget, allUnivs)', find and return the list of codes of institutes which matches our search criteria.
  - a. countryName → Country name is the input parameter and allUnivs is all university/Institutes.
  - b. degrees → list of degrees user want to search in institutes.
  - c. budget → budget is the input parameter which helps to search the institutes within user budget.
  - d. allUnivs → all institutes/universities.
- 12. 'studyInTwoPlaces (firstCode, firstDegree, secondCode, secondDegree, budget, allUnivs)', find and confirm with True/False whether our search meets the provided input in respective institutes/universities.
  - a. firstCode & firstDegree  $\rightarrow$  first code is the institute code and first degree is the degree name which is offered by institute.
  - b.  $secondCode \& secondDegree \rightarrow second code in the another institute code and second degree is the another degree name which is offered by institute.$
  - c. dudget → budget is the input parameter to search institute fee within the range.

Note: The requirement explanation available in the below link	
> https://voutu.be/B7ltgBY5D8A?si=AF_O0DtTl -BtXdb <	