

CE151 ASSIGNMENT 2 2018

Set by: Mike Sanderson & Giovanni Stracquadanio

Credit: 20% of total module mark

Deadline: 11.59.59, 10 December 2018

Submission of this assignment will be via FASER; you will not be required to demonstrate your programs in the lab.

It is expected that marks and feedback will be returned by the middle of week 17.

You should refer to sections 5 and 7 of the Undergraduate Students' Handbook for details of the departmental policy regarding late submission and plagiarism; the work handed in must be entirely your own.

This assignment comprises two exercises, to be written as separate programs in files called **ex1.py** and **ex2.py**. Each will contribute 45% of the mark. The remaining 10% of the total mark for the assignment will be awarded for programming style and documentation. Sample data files to be used for testing the programs will be supplied.

Exercise 1

The attributes of a football player are name, last name, position, team and salary.

In the file **ex1.py** write a function which takes a string argument, and creates and returns a tuple containing details of the player specified by the string. The string should be assumed to have the format

```
Chelsea Right-back Davide Zappacosta 20,000,000
```

The tuple will contain the fields in the following order: team, position, name, lastname and salary. The tuple should contain exactly 5 items.

Next, write a function that will print the details of a player on a single output line. It should take a tuple as its argument and print the details in fixed-width fields using a layout such as

```
Zappacosta, Davide 20,000,000 Right-back Chelsea
```

The name must be displayed in a single fixed-width field using the format shown above so you will need to create a string containing the name in this format and print it in a fixed- width field. You may assume that no name will contain more than 30 characters when displayed in this format, position will contain more than 15 characters, salary will contain at most 8 digits.

In the main body of the program, write code which prompts the user for a filename and attempts to open the file whose name is supplied. If the file cannot be opened an error message should be output and the program should terminate; otherwise the program should read each line from the file and supply it to the player tuple-creation function, storing the tuples returned by this function in a list. You may assume that the file contains lines that conform to the format described for the argument to the function.

After the data has been input the program should display details of all the players in the list in a neat table (using the function already written) then enter a loop in which the user should be given the option of requesting the display of (a) full details of (i) the player with a given lastname or (ii) all players with a salary in a particular range (e.g. 20000 to 30000) or (b) the first and last names of all players of a team or quitting the program.

The user should then be asked to supply the position, team or lower and upper bounds of the salary range (as appropriate), the list should then be searched, and the appropriate output displayed (using the function written earlier when full details are required). The salary output should be sorted numerically in ascending order of salary. Appropriate messages should be displayed if a search produces no results. The user should be told what has to be typed in order to select each option; the input options should be concise (e.g. '1', '2' etc. or 'j', 's' etc.) in order to allow quick testing of the program.

Exercise 2

The attributes of a city are name, latitude, longitude and number of citizens.

In the file **ex2.py** write a function which takes a string argument, and creates and returns a tuple containing details of the city specified by the string. The string should be assumed to have the format

```
Rome 111000 20000 1200000
```

The items in the tuple will be name, latitude, longitude, and number of citizens. You can assume that latitude and longitude are 2D coordinates in KM, not degrees. The tuple should contain exactly 4 items.

Next, write a function that will print the details of a city on a single output line. It should take a tuple as its argument and print the details in fixed-width fields using a layout such as

```
Rome n. citizens: 1,200,000
```

The name must be displayed in a single fixed-width field using the format shown above so you will need to create a string containing the name in this format and print it in a fixed-width field. You may assume that no name will contain more than 30 characters when displayed in this format, the number of citizens will contain at most 8 digits.

In the main body of the program, write code which prompts the user for a filename and attempts to open the file whose name is supplied. If the file cannot be opened an error message should be output and the program should terminate; otherwise the program should read each line from the file and supply it to the player tuple-creation function, storing the tuples returned by this function in a list. You may assume that the file contains lines that conform to the format described for the argument to the function.

After the data has been input the program should display details of all the cities in the list in a neat table (using the function already written) then enter a loop in which the user should be given the option of requesting the display of (a) full details of (i) the city with a given name or (ii) all the cities with a population in a particular range or (b) the city, if any, within 10Km from the coordinates specified by the user.

The user should then be asked to input 2 cities names, and the program should be able to compute the distance between the cities using the information about longitude and latitude.