

## CSC1015F Assignment 8: Arrays

### Assignment Instructions

This assignment involves constructing Python programs that manipulate lists, dictionaries and strings. You can use the following built-in functions in Python: `map()`, `math.sqrt()`, `split()`.

**NOTE** Your solutions to this assignment will be evaluated for correctness and for the following qualities:

- Documentation
  - Use of comments at the top of your code to identify program purpose, author and date.
  - Use of comments within your code to explain each non-obvious functional unit of code.
- General style/readability
  - The use of meaningful names for variables and functions.
- Algorithmic qualities
  - Efficiency, simplicity

These criteria will be manually assessed by a tutor and commented upon. In this assignment, up to 10 marks will be deducted for deficiencies.

### Question 1 [10 marks]

Write a Python program, `duplicates.py`, in which the user can enter a list of values and these values are then printed in the same order but without duplicates.

Use the sentinel "DONE" to end the input list.

#### *Sample I/O*

Enter strings (end with DONE):

**the**  
**old**  
**man**  
**and**  
**the**  
**sea**  
**DONE**

Unique list:

the  
old  
man  
and  
sea

**Note:** the user input list is in bold.

### Question 2 [20 marks]

Write a program, `gridio.py`, that allows a user to input integer values and query a 2-dimensional array of size 9x9. Your program should then ask the user for a pair of coordinates,  $(x, y)$ , separated by a space and return the value at the position specified by the given coordinates.

For instance, 0 3 should return the value 7 (the value at row one, column four – bearing in mind that array indices start at zero).

Assume that each integer is a single digit from 1 – 9. Enter -1 for either coordinate to end the program.

#### Sample I/O:

```
Enter an array:
359716482
867345912
413928675
398574126
546281739
172639548
984163257
621857394
735492861

Enter coordinates:
0 3
Value = 7
Enter coordinates:
5 5
Value = 9
Enter coordinates:
8 8
Value = 1
Enter coordinates:
-1 -1
DONE
```

### Question 3 [40 marks]

Write a program called '`vectormath.py`' to do basic vector calculations in 3 dimensions: addition, dot product and normalization.

A vector has 3 component values, such as  $(1, 3, 2)$  and is naturally storable as an array.

- Addition of vectors requires addition of the corresponding elements.
- A dot product is the sum of the products of corresponding elements.
- The norm of a single vector is the square root of the sum of the squares of the elements.

Suppose that we have 2 vectors:  $A=(1, 3, 2)$  and  $B=(2, 3, 0)$ :

- Addition:  
 $A+B = (1+2, 3+3, 2+0) = (3, 6, 2)$
- Dot product:  
 $A.B = 1.2 + 3.3 + 2.0 = 2 + 9 = 11$
- Norm (of A):  
 $|A| = \text{Sqrt}(1^2 + 3^2 + 2^2) = \text{Sqrt}(1+9+4) = \text{Sqrt}(14) = 3.74$
- Norm (of B):  
 $|B| = \text{Sqrt}(2^2 + 3^2 + 0^2) = \text{Sqrt}(4+9+0) = \text{Sqrt}(13) = 3.61$

*Continued*

For the norms, print your answer to 2 decimal positions.

Sample I/O:

Enter vector A:

1 3 2

Enter vector B:

2 3 0

A+B = [3, 6, 2]

A.B = 11

|A| = 3.74

|B| = 3.61

#### Question Four [30 marks]

Write a program called 'election.py' to count the number of votes for each political party in an election. Your program must accept a sequence of names of parties (terminated by the word DONE) and keep track of the votes per party. There is no pre-determined party list so the list must be created as party names are encountered.

At the end of the program, print out the names of the parties and their vote counts, with the parties listed in sorted order (the default order of the sorted function). Format party names in a field with width=10 to create the effect of columns - assume all party names are at most 10 characters wide.

Sample I/O:

Independent Electoral Commission

-----

Enter the names of parties (terminated by DONE):

apples

oranges

oranges

oranges

pears

bananas

bananas

kiwis

oranges

apples

oranges

DONE

Vote counts:

apples - 2

bananas - 2

kiwis - 1

oranges - 5

pears - 1

#### Submission

Create and submit a Zip file called 'ABCXYZ123.zip' (where ABCXYZ123 is YOUR student number) containing duplicates.py, gridio.py, vectormath.py and histogram.py.

**END**

*Continued*