CSC1015F Assignment 6: Arrays

Assignment Instructions

This assignment involves constructing Python programs that manipulate lists, dictionaries and strings.

Question 1

This question concerns the development of an automated technical support system[€]. Users can enter technical queries and the system will suggest solutions.

As it stands, the tech support system consists of the program '`nohelp.py`'. Running this program produces the following behaviour:

```
Welcome to the automated technical support system.
Please describe your problem:
My computer keeps crashing
Curious, tell me more.
I don't think it has a driving license
Curious, tell me more.
Really?
Curious, tell me more.
Oh forget it!
Curious, tell me more.
quit
```

The program just prints out the same response again and again.

1.1 Some help [20 marks]

Using `nohelp.py` as the basis, create a new program called '`somehelp.py`' that randomly selects a response from a list. The list should contain the following responses in the following order (so that your work can be automarked):

1. `Have you tried it on a different operating system?`
2. `Did you reboot it?`
3. `What colour is it?`
4. `You should consider installing anti-virus software.`
5. `Contact Telkom.`
6. `I'd get that looked at if I were you.`

Here's an example of expected behaviour:

```
Welcome to the automated technical support system.
Please describe your problem:
My computer keeps crashing
What colour is it?
Blue, why?
You should consider installing anti-virus software.
Oh
You should consider installing anti-virus software.
You said that
```

[€]Assignment based on series of exercises from "Objects First with Java" by Barnes and Kölling.

```
Did you reboot it?
No
Contact Telkom.
quit
```

NOTE: To enable automarking, your program must:

1. Use a 'main()' function.
2. Use a list containing the given responses in the given order.
3. Use the following conditional to ensure it is not run when imported:
   ```
   if __name__=='__main__':
       main()
   ```
4. Randomly select responses by using the Python `random.randint()` function. (you must import `random`.)

   The function '`randint(a, b)`' returns a random integer N such that a <= N <= b.

1.2 Support [20 marks]

Using `somehelp.py` as the basis, create a new program called `support.py`.

Modify `support.py` so that it keeps a dictionary of responses indexed by keywords.

- Assume that the user only ever inputs a single word at a time.
- Given a word entered by the user, the program will look for that entry in the dictionary and will print the associated response.
- If there is no entry for that word the program will output '`Curious, tell me more.`'.

The dictionary should contain the following keyword-response pairs:

keyword	***response***
`crashed`	`Are the drivers up to date?`
`blue`	`Ah, the blue screen of death. And then what happened?`
`hacked`	`You should consider installing anti-virus software.`
`bluetooth`	`Have you tried mouthwash?`
`windows`	`Ah, I think I see your problem. What version?`
`apple`	`You do mean the computer kind of apple don't you?`
`spam`	`You should see if your mail client can filter messages.`
`connection`	`Contact Telkom.`

Here's an example of expected behaviour:

```
Welcome to the automated technical support system.
Please describe your problem:
crashed
Are the drivers up to date?
```

CONTINUED

```
yes
Curious, tell me more.
blue
Ah, the blue screen of death. And then what happened?
hacked
You should consider installing anti-virus software.
quit
```

NOTE: To enable automarking, your program must use a '`main()`' function and must contain the exact set of keyword and response pairs listed above.

1.3 Tech Support [30 marks]

Using `support.py` as the basis, create a new program called `techsupport.py`. Modify `techsupport.py` so that it splits a query up into a list of words and then, taking each in turn, searches the dictionary for a match.

- Once it finds a match it should print the associated response.
- It should print only one response per query.
- If none of the words can be matched then the program should output '`Curious, tell me more.`'.

Here's an example of expected behaviour:

```
Welcome to the automated technical support system.
Please describe your problem:
My computer crashed
Are the drivers up to date?
No
Curious, tell me more.
I don't have an internet connection
Contact Telkom.
I did
Curious, tell me more.
They told me to use bluetooth
Have you tried mouthwash?
quit
```

NOTE:

- You may assume that user input is not punctuated.
- To enable automarking, your program must use a '`main()`' function and must contain the exact set of keyword and response pairs listed above.

HINT: A string can be split into a list of constituent words by using the '`split()`' method. For example:

```
>>> 'the rain in spain falls mainly on the plain'.split()
['the', 'rain', 'in', 'spain', 'falls', 'mainly', 'on', 'the',
'plain']
>>>
```

CONTINUED

Question 2 [30 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| = Sqrt(1^2 + 3^2 + 2^2) = Sqrt(1+9+4) = Sqrt(14) = 3.74
- Norm (of B):
  |B| = Sqrt(2^2 + 3^2 + 0^2) = Sqrt(4+9+0) = Sqrt(13) = 3.61

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
```

Submission

Create and submit a Zip file called '`ABCXYZ123.zip`' (where ABCXYZ123 is YOUR student number) containing `somehelp.py`, `support.py`, `techsupport.py`, and `vectormath.py`.

END