ISD Term 1 Coursework 2

Dr David Weston, Birkbeck, University of London

- Deadline for submission: **Friday 10th January at 17:55**. Please read the *Submitting the assignment* section at the end of this document.
- Second deadline (mark will be capped to pass mark unless there are mitigating circumstances) is Friday 24th January 17:55.
- Any submissions uploaded after the second deadline will be capped at zero marks. No mitigation will be accepted after this date. Mitigation forms must be submitted one week after the first deadline at the latest.

Preliminaries - Important Please Read Carefully

1. Download from Moodle - Week 10:
   `term1cw2.py`.
2. Read carefully the plagiarism statement at:
   `http://www.bbk.ac.uk/registry/policies/documents/assessment-offences-policy.pdf`
   Once you have read it, add to the top of `term1cw2.py` as a multi-line comment, the following:

   ```
   I have read and understood the sections of plagiarism in the College Policy
   on assessment offences and confirm that the work is my own,
   with the work of others clearly acknowledged.
   I give my permission to submit my work to the plagiarism testing database
   that the College is using and test it using plagiarism detection software,
   search engines or meta-searching software.
   ```

   You will get **0 marks** if this is not included in your submission
3. Do not use Github or any publicly accessible site to store your work. This will be regarded as a plagiarism offence.
4. Do not be tempted to use contractor websites to find somebody to do the coursework for you.

The Coursework

Introduction to Software Development has 4 courseworks. Each coursework contributes equally to the overall coursework mark.

By doing this coursework you will get experience in writing functions, using loops and lists; and comparing iteration with recursion.

The Questions

There are 4 questions. This coursework is marked out of 50. All questions must be answered using Python 3 programming language.

Do **not** use any Python functionality that has not been shown in the lectures, i.e. no exception handling, dictionaries, sets, etc.
Do **not** use `import` in your code.

Always keep backup copies of all assignments. If your assignment gets lost, a backup copy will make things easier for you.

1. Review Question 3 from Coursework 1 and my sample solution.

   (a) (15 marks)
   Write a function called `DiamondQuality`, this function has one argument. This argument is called `properties` and it is a `list of 3` elements that correspond to *table*, *pavillion* and *depth* in this order.
   The function returns a `list` containing 2 `strings`.
   The first element of the returned list is a feedback string, which is one of the following:

   ```
   "Inputs are all valid"
   "The number of properties must equal 3"
   "At least one input is incorrect"
   ```

   The second element of the returned list is an empty string if the input is not valid, otherwise it is one of the following:

   ```
   "This diamond is bad"
   "This is a good diamond"
   ```

   Use the lists provided in `term1cw2.py`. This will mean you can reduce the number of conditional statements you need to hardcode by replacing them with a loop. The simpler your program (i.e. the less hardcoded the decisions), the more marks you will receive.
   Examples: (Note quotation marks may need to be replaced if you wish to copy to Python.)
   Running the following lines:

   ```
   print(DiamondQuality(["53","42","60"]))
   print(DiamondQuality(["53","10","60"]))
   print(DiamondQuality(["fifty","42","60"]))
   print(DiamondQuality(["fifty","42","60","w"]))
   ```

   Will output the following:

   ```
   ['Inputs are all valid', 'This is a good diamond']
   ['Inputs are all valid', 'This diamond is bad']
   ['At least one input is incorrect', '']
   ['The number of properties must equal 3', '']
   ```

(b) (5 marks)
Write a function called `main()` that accepts three inputs that will be used in a call to your `DiamondQuality` function. If the user enters 'q' in any of the entries then your program terminates. If the user enters anything other than the ending sentinel, call your `DiamondQuality` function with the values entered by the user. If the values are legitimate, display the resulting category and your program should end. If the values are not legitimate, display the feedback message returned from your `DiamondQuality` function and loop to accept another round of user entered data.

Your program should replicate the following behaviour:

```
Please input, table 53
Please input, pavillion 42
Please input, depth 60
This is a good diamond
```

An example where the user enters incorrect values then quits:

```
Please input, table fifty
Please input, pavillion 42
Please input, depth 60
At least one input is incorrect
Please input, table fifty
Please input, pavillion q
Please input, depth 60
```

Use the list provided in `term1cw2.py` to simplify your code.

(c) 5 additional marks will be awarded for the clarity of your code for the above two functions, this includes the following:

- Comments. See function comments in Section 5.2
- Appropriate variable, function and argument names.

2. (10 marks)
Write a function called `ExpandIt` that is a refactor of the following code. Your function must not use recursion.

```
## Repeat each letter in a string.
#  @param text the text to update.
#  @return string containing the doubled letters.
#
def Expand(text):
   if len(text) == 0:
      return ''
   else:
      return text[0] + text[0] + Expand(text[1 : ])
```

Hint: what does `Expand('abcd')` return?

3. The following are distinction level questions.

   (a) (5 marks)
   Write a function called `ExpandCentreIt`. This function is an amended version of `ExpandIt`. It doubles up all the letters but places them in a different order. The first character is in the centre and the remaining characters are placed on either side.
   For example
   `ExpandcentreIt("a")` should equal `"aa"`
   `ExpandcentreIt("ab")` should equal `"baab"`
   `ExpandcentreIt("abcd")` should equal `"dcbaabcd"`
   `ExpandcentreIt("abc")` should equal `"cbaabc"`

   (b) (5 marks)
   Write a function called `ExpandCentre`. This is a refactored version of `ExpandCentreIt`. Your function must use recursion.

4. (5 marks)
   For correctly following the submission instructions below.

Submitting the assignment:

1. Do not forget the plagiarism declaration comment at the top of your file. You will get **0 marks** if it is not there.

2. Include a further comment directly after the plagiarism declaration that has your name, the name of the programme you are taking (e.g., MSc IT, etc.), and the submission date.

3. Submit **only** your `term1cw2.py` do **not** zip your file. Do not rename this file.

4. Submission is only through Moodle.

5. To submit your file, follow the Upload Submission for Term1 - Coursework2 link on the ISD Moodle page.