OOP Practice 2

1 Name your Jupyter Notebook as

`TASK1_<your name>_<centre number>_<index number>.ipynb`

A computer game requires players to travel around a mythical island to hunt for and open treasure chests. Each treasure chest contains a mathematical question to be answered by the player. The number of points awarded for correctly answering a question depends on the number of attempts taken by the player to provide the correct answer.

The task is to design the game using object-oriented programming (OOP) techniques.

For each of the sub-tasks, add a comment statement at the beginning of the code, using the hash symbol '#' to indicate the sub-task the program code belongs to, for example:

```
In [1] : # Task 1.1
         Program code

         Output:
```

Task 1.1

In designing the game, the class `TreasureChest` needs to be implemented.

The `TreasureChest` class has the following attributes:

Attribute	Data Type	Description
`Question`	String	the question in a particular treasure chest
`Answer`	Integer	the answer to the question in that treasure chest
`Points`	Integer	the maximum possible points attainable for that treasure chest

The `TreasureChest` class has the following methods:

Method	Description
`getQuestion()`	• the question in the treasure chest
`checkAnswer()`	• takes the user's answer as a parameter • returns True if the answer is correct and False otherwise

`getPoints()`	• takes the number of attempts as a parameter
	• if the number of attempts is
	  ○ 1, returns the value of points.
	  ○ 2, returns the integer value of points divided by 2 (`DIV 2`).
	  ○ 3 or 4, returns the integer value of points divided by 4 (`DIV 4`).
	• if the number of attempts is not 1 or 2 or 3 or 4, it returns the integer 0.

	• For example, a question is worth 90 points and the user took 3 attempts to give the correct answer. The user is awarded 22 points (`90 DIV 4`).

Implement the class `TreasureChest`. [14]

Task 1.2

The text file `TreasureChestData.txt` stores data for five questions, in the order of question, answer and points.

For example, the first three lines of the file are for the first question:

```
2*2
4
100
```

where `2*2` is the question, `4` is the answer and `100` is the points

Write program code for the procedure, `readData()` to:

- read each question, answer and points from the text file
- create an object of type `TreasureChest` for each question
- declare an array named `arrayTreasure` of type `TreasureChest`
- append each object to the array
- use exception handling to output an appropriate message if the file is not found

[8]

Task 1.3

Write program code for the main program to

- call the procedure `readData()`
- ask the user to enter a question number from 1 to 5
- output the question that matches the question number entered by the user
- check if the answer input by the user is correct using the method `checkAnswer()`
- repeat the question until the user inputs the correct answer
- count how many times the user attempted the question
- use the method `getPoints()` to return the number of points awarded
- output the number of points the user is awarded.

[7]

Task 1.4

Test the program three times.

In the first test, select question 1 and answer it correctly the first time.

In the second, test select question 5 and answer it correctly the second time.

In the third test, select a question and answer it correct only at the fifth time.

[3]