CompSci 101 - Assignment 02

Due: 4:30pm, Tuesday 2nd May 2017.

Worth: This assignment is marked out of 25 and is worth 2.5% of your final mark.

Topics covered:
- Functions
- If statements
- While loops

NOTE (VERY IMPORTANT): You must not use any for loop statements in your code.

NOTE (VERY IMPORTANT): Your programs for this assignment must all have the following format:

```
# add the functions that main() calls

def main():
    # statements in the main function of the program

main()
```

You will be adding your own functions to the program. **There must not be any variables used outside any of the functions.**

NOTE (VERY IMPORTANT): You must include a docstring at the top of your file containing your name, UPI, ID number and a description of the program.

Submit the file containing your exercises using the Assignment Dropbox:

https://adb.auckland.ac.nz/Home/
**QUESTION 1 (25 MARKS)**

Write a program that fulfills the functionalities of a mathematical quiz with the five basic arithmetic operations, i.e., addition, subtraction, multiplication, integer division and modulo operation. A sample partial output of the math quiz program is shown below.

```
************************
** A Simple Math Quiz **
************************
1. Addition
2. Subtraction
3. Multiplication
4. Integer Division
5. Modulo Operation
6. Exit
------------------------
Enter your choice:
```

The user can select the mathematical operation that he/she would like to proceed with. Once a choice (i.e., menu option index) is entered, the program generates a question and asks the user for an answer. A sample partial output is shown below. Note that user inputs are denoted in bold and blue in the examples.

```
************************
** A Simple Math Quiz **
************************
1. Addition
2. Subtraction
3. Multiplication
4. Integer Division
5. Modulo Operation
6. Exit
------------------------
Enter your choice: 1
Enter your answer
15 + 2 =
```

After receiving an answer, the program then checks whether the user’s answer is correct or not, according to the result of the actual computation. The program continues with asking the user to choose another mathematical operation for the quiz. A sample partial output is shown below.

```
************************
** A Simple Math Quiz **
************************
1. Addition
2. Subtraction
3. Multiplication
4. Integer Division
5. Modulo Operation
6. Exit
------------------------
Enter your choice: 1
Enter your answer
15 + 2 = 17
```
Note that if the user enters an invalid menu option index, the program responds with an error message and prompts the user to enter a correct menu option (until it obtains one), as shown in the above example.

The program will continue running with more questions to answer until the user selects the “Exit” option on the menu. Once the exit option (i.e., index 6) is entered, the program stops and prints out the result of the quiz, in terms of the total number of questions answered, the number of questions that are correct and the score in a percentage form. A sample output is shown below.

```
************************
** A Simple Math Quiz **
************************
1. Addition
2. Subtraction
3. Multiplication
4. Integer Division
5. Modulo Operation
6. Exit
------------------------
Enter your choice: 1
Enter your answer
12 + 13 = 25
Correct.
Enter your choice: 2
Enter your answer
9 - 8 = 1
Correct.
Enter your choice: 3
Enter your answer
7 * 17 = 119
Correct.
Enter your choice: 4
Enter your answer
8 // 12 = 0
Correct.
Enter your choice: 2
Enter your answer
13 - 17 = -4
Correct.
Enter your choice: 3
Enter your answer
2 * 11 = 22
Correct.
```
Your program must give the correct output in the same format as the outputs in the above examples. There are three assumptions made for the above simple math quiz program, i.e.,

- The numbers in the questions are randomly generated between 1 and 30 (both inclusive);
- The calculation for division is an integer division, where the result is truncated to the integer value only, e.g., \( 25 \div 7 = 3 \).
- The calculation for the modulo operation uses the \% operator, where the result is the remainder after the division, e.g., \( 17 \% 6 = 5 \).

The `main()` function is given below and it should not be changed in any way. Copy this code into your program.

```python
def main():
    display_intro()
    display_menu()
    display_separator()

    option = get_user_input()
    total = 0
    correct = 0
    while option != 6:
        total = total + 1
        correct = menu_option(option, correct)
        option = get_user_input()

    print("Exit the quiz.")
    display_separator()
    display_result(total, correct)

main()
```

Complete the following eight functions so that the completed program runs as described above:
• **def display_intro():**

This function prints out the heading part of the menu, i.e., the line of stars, the name of the quiz program and another line of stars. There are 24 star symbols in one line of stars. Note that the string repeat operator “*” might be useful to repeat the same symbol a number of times.

• **def display_menu():**

This function prints out the actual menu options with associated index values.

• **def display_separator():**

This function prints out the separation line which consists of 24 “-” symbols.

• **def get_user_input():**

This function obtains an integer from the user as the selection of the menu index. You can assume that the user is always going to enter a positive integer value. However, you do need to check whether the integer value entered by the user is a valid menu option index or not. If the input is outside the range of 1-6 (inclusive), the function will print an error message and ask for the correct input, and it will continue to do so until a valid menu index is obtained. Finally, the function returns the user input choice.

• **def get_user_solution(problem):**

This function instructs the user to enter an answer for the math question which is stored in the parameter variable ‘problem’. The function returns the input value as the user’s answer.

• **def check_solution(user_solution, solution, count):**

This function checks whether the user answer (in the parameter variable ‘user_solution’) is the same as the actual solution of the question (in the parameter variable ‘solution’). The parameter variable ‘count’ stores the total number of questions correctly answered by the user so far in the quiz. Based on the correctness of the user answer, the function prints out a corresponding message (i.e., either correct or incorrect). If the user answer is correct, the function also increases the total number of correct answers by 1. Finally, the function returns the total number of correctly answered questions so far.

• **def menu_option(index, count):**

This function handles a specific menu option choice, which consists of the following steps.

  o Generate two random numbers between 1 and 30 (both inclusive);
  o Generate the math question and its actual solution according to the user selected menu option (value in the parameter variable ‘index’);
  o Obtain the user answer by calling the ‘get_user_solution’ function and passing the problem as a parameter;
  o Check the correctness of the user answer by calling the ‘check_solution’ function (where the parameter variable ‘count’ stores the number of questions correctly answered so far and it should be passed to the function call);
  o Finally, the function returns the total number of correctly answered questions so far which was obtained from the ‘check_solution’ function.
• def display_result(total, correct):
  
  This function displays the result of the quiz. The parameter variable ‘total’ stores the total number of questions, and the variable ‘correct’ stores the number of questions answered correctly. Based on these two values, the function works out the percentage score (in two decimal places) and prints out the result. Note that if the user did not answer any question and chose to exit, your program should print the score as “0%”, as shown below.

  ************************
  ** A Simple Math Quiz **
  1. Addition
  2. Subtraction
  3. Multiplication
  4. Integer Division
  5. Modulo Operation
  6. Exit
  ------------------------
  Enter your choice: 6
  Exit the quiz.
  ------------------------
  You answered 0 questions with 0 correct.
  Your score is 0%. Thank you.

  Hint: The use of if and while statements for the menu control handling was discussed in lectures. Please refer to the L11 slide 14 and L12 slides 14 for detailed examples.

  Save this exercise in a module (file), named 'YourUPIA2Q1.py', e.g., abcd001A2Q1.py, for submission.

  Mark Allocation:

  The mark allocations for each of the eight functions are as follows, where the total is 25.

  • display_intro – 3 marks
  • display_menu – 3 marks
  • display_separator – 2 marks
  • get_user_input – 3 marks
  • get_user_solution – 3 marks
  • check_solution – 3 marks
  • menu_option – 5 marks
  • display_result – 3 marks