QUESTION 1 (35 MARKS)
Write a program that fulfills the functionalities of a mathematical quiz with the four basic arithmetic operations, i.e., addition, subtraction, multiplication and integer division. A sample partial output of the math quiz program is shown below.
The user can select the type of math operations 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.

```
1. Addition
2. Subtraction
3. Multiplication
4. Integer Division
5. 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 math operation for the quiz. A sample partial output is shown below.

```
1. Addition
2. Subtraction
3. Multiplication
4. Integer Division
5. Exit

Enter your choice: 1
Enter your answer
15 + 2 = 17
Correct.

Enter your choice: 7
Invalid menu option.
Please try again: 3
Enter your answer
20 * 19 = 280
Correct.

Enter your choice: 2
Enter your answer
13 - 17 = -5
Incorrect.

Enter your choice:
```
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 5) 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. 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.
Enter your choice: 4
Enter your answer
2 / 18 = 1
Incorrect.
Enter your choice: 1
Enter your answer
8 + 15 = 23
Correct.
Enter your choice: 5
Exit the quiz.
------------------------
You answered 8 questions with 7 correct.
Your score is 87.5%. Thank you.
```
Your program must give the correct output in the same format as the outputs in the above examples.

There are two assumptions made for the above simple math quiz program, i.e.,

- The numbers in the questions are randomly generated between 1 and 20 (both inclusive);
- The calculation on division is an integer division, which the result is truncated to the integer value only, e.g., $25/7=3$.

The \texttt{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 != 5:
        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:

- \texttt{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 \texttt{"\*"} might be useful to repeat the same symbol a number of times.

- \texttt{def display_menu()}: This function prints out the actual menu options with associated index values.

- \texttt{def display_separator()}: This function prints out the separation line which consists of 24 \texttt{"-"} symbols.

- \texttt{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. If the input is outside the range of 1-5.
(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 20 (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.

Submission: Include this exercise in a module (file), named 'YourUPIA2Q1.py', e.g., abcd001A2Q1.py.

Mark Allocation:
The mark allocations of the eight functions to be implemented are as follows, where the total is 35.

- def display_intro() - 3 marks
- def display_menu() - 3 marks
- def display_separator() - 3 marks
- def get_user_input() - 6 marks
- def get_user_solution(problem) - 5 marks
- def check_solution(user_solution, solution, count) - 5 marks
- def menu_option(index, count) - 5 marks
- def display_result(total, correct) - 5 marks