Part A

Read Chapter I (class 1) and Chapter II (class 2) of the text material.

Part B – all codes in Python

1. Compute and print both roots of the quadratic equation x^2 – 5.86 x + 8.5408.

   Use a **for** loop to print the decimal representations of 1/2, 1/3, ..., 1/10, one on each line.

2. Use a for loop to compute the 10th triangular number. The *n*th triangular number is defined as 1+2+3+...+*n*.
   (You can also compute the *n*th triangular number as *n**(*n*+1)/2. Use this formula to double-check that your loop is correct.)
   Hint: This outline is an almost-complete solution. You only must replace each ellipsis by an expression.

3. Use a for loop to compute 10! (the factorial of 10). Recall that the factorial of n is 1*2*3*...*n.
   The first line of your solution will be n = 10. After that, your solution should not use 10 again, though your solution will use n. In other words, your code (after the n = 10 line) should work for any value of n.
   Hint: Your answer will be like your answer to "Problem 3: Triangular numbers".

4. Write code to print the first 10 factorials, in reverse order. In other words, write code that prints 10!, then prints 9!, then prints 8!, ..., then prints 1!. Its literal output will be:

   ```
   3628800
   362880
   40320
   5040
   720
   120
   24
   6
   2
   1
   ```

   The first line of your solution should assign a variable **numlines** to 10, and then the rest of your solution must not use 10 anywhere.
   Hint: Use two nested for loops.

5. Compute the following value:
   1 + 1/1! + 1/2! + 1/3! + 1/4! + ... + 1/10!
   The value should be close to *e* (≈ 2.71828), the base of the natural logarithms.

Hint: The easiest way to solve this is with two nested for loops. It is possible, but tricky, to compute this using only one for loop. That is not necessary for this assignment.
Hint: Copy your solution to "Problem 5: Multiple factorials", then modify it. Rather than printing the factorials, you will add their reciprocals to a running total, then print that total at the end.
Hint: don't try to work the very first "1 +" into your loop; do it outside the loops (either at the very beginning or the very end of the outer loop).

6. Part A: NOT GRADED: import the MS Excel Worksheet **Sample_01** to Python. Do all sort of statistics and graphs.
   Part B: NOT GRADED: Prepare a small database to import the values in **Sample_01**, like we have seen in **5_class_example.py**.

GOOD LUCK!