All Problems of PyCampaign20 in one Sheet

1. Print the following poem:

   The Zen of Python, by Tim Peters

   Beautiful is better than ugly.
   Explicit is better than implicit.
   Simple is better than complex.
   Complex is better than complicated.
   Flat is better than nested.
   Sparse is better than dense.
   Readability counts.
   Special cases aren't special enough to break the rules.
   Although practicality beats purity.
   Errors should never pass silently.
   Unless explicitly silenced.
   In the face of ambiguity, refuse the temptation to guess.
   There should be one-- and preferably only one --obvious way to do it.
   Although that way may not be obvious at first unless you're Dutch.
   Now is better than never.
   *Although never is often better than *right* now.*
   If the implementation is hard to explain, it's a bad idea.
   If the implementation is easy to explain, it may be a good idea.
   Namespaces are one honking great idea -- let's do more of those!

2. Create a Myself paragraph (7 lines). It will not be a regular paragraph; it will be a general version of myself paragraph. You have to ask user for their name, age, class, favorite hobby (4 lines), manipulate the code for "Zhamela" users, so that all the casing remain perfect. then concate the user's input and make a paragraph which have nice beginning and ending lines (3 lines). then print the answer and send me the code.

3. Suppose you are a salesman in a shopping mall. You have to do certain tasks:
   1. Ask the user how much he spent on the shopping.
   2. Add 10percent vat with that shopping charge.
   3. If the shopping charge is above 500 then give 20percent discount (over vat added charge)
   4. Otherwise give 5percent discount (same as above)
   5. Print the final charge in a string by concatenating it. Exp: your charge is 567.
   6. Add some extra lines in if-else condition or without condition. Exp: have a nice day.

   * Extra point- complete the task using Booleans.

4. Draw 5 pentagons inside a big pentagon.

5. Print Fibonacci series using for loop and while loop.

6. Complete following tasks:
   1. Create an empty guest list. 2. Ask the user for guest names until they input done (use while loop)
   3. Append the inputs to the guest list (without done)
   4. After the entire input is completed, iterate through the list and print all the elements.

7. Complete following tasks:
   1. ask user for total Testcases.
   2. for each testcase,
   3. take two lines of numbers as an input from the user
   4. make each lines of input into a list
   5. then print the numbers which are common in the list (as a line of numbers -string)

8. create a calculator using function. Your calculator must have basic operations like + - x /

9. Remember problem 8, you have to make a basic calculator, now you have to handle errors in that calculator program.

10. What do you understand by the probability of coin flipping is 1/2? It doesn't mean if you toss the coin 2 times once you will get heads and the others tails. It means if you continuously toss the coin eventually the ratio of heads and tails become "near" 50:50. Create a function which will describe the same theory in code. You will have to -
    1. Use random module
    2. Toss the coin at a large amount of times.
    3. Return the number of times when the toss supports heads and tails
    4. Also return the ratio.