1. Ask the user for a float value then show her the value:

\[
\cosine \sqrt{x^3}
\]

You do NOT have to convert the number into radians.
Do write the required import(s) that would be needed.

2. What! you didn't write a two functions to do Question 1.?
   You SHOULD have written a function that returns the calculation.
   You SHOULD have written the docstring for this function immediately after writing "def" to help
   you focus on writing the function
   Then you should have written a function that calls raw_input and then calls your calculating
   function – and again you should have written the docstring immediately after writing "def".
   Start over on Q1 and do it the correct way this time (if you didn't)

3. Write the function named oneIsPositive that returns True when any of it's three parameters is
   positive

4. What is the relationship between the return type of a function and the types of the parameters?

5. How many values can a function return to its caller?
   You can answer this one in two ways since you now know about lists

6. How does a programmer communicate with the user?
   How does a programmer communicate with the other programmer that calls her functions?
   Do both mechanisms have to be in every function?
   Is writing a value on the screen communicating with the caller of a function?

7. What are the three ways a programmer writing a function def can get information?

8. What is wrong with this function call (other than it's horribly named)?

   ```python
   def f( nameOfAnimal ):
       ''' returns whether or not the name is in our database '''
       ...
   f( 3 )
   ```

9. What is the parameter passing mechanism in python?
   What does that mean in terms of RAM and referrals?

10. How many copies are there of the data passed into a function are there? (see previous question!)

11. What is the scope of a parameter? of a local variable?
12. Why does this show a misunderstanding? What is being misunderstood

```python
def puny(bird):
    ''' . . .
def func(bird2):
    ''' . . .
```

13. def main must be the first def in a python program (true or false)

14. What is wrong with this code? The correct breakdown is stepONE, stepTWO, stepTHREE.

```python
def main():
    BAD CODE!!!
    stepONE()
    BAD CODE!!!
    BAD CODE!!!

def stepTWO():
    BAD CODE!!!
    BAD CODE!!!
    ''' does step two '''
    BAD CODE!!!
    BAD CODE!!!
    # code to do step two goes here
    BAD CODE!!!
    BAD CODE!!!
    stepTHREE()
    BAD CODE!!!
    BAD CODE!!!

def stepTHREE():
    BAD CODE!!!
    BAD CODE!!!
    ''' does step three '''
    BAD CODE!!!
    BAD CODE!!!
    # code to do step three goes here
    BAD CODE!!!
    BAD CODE!!!

def stepONE():
    BAD CODE!!!
    BAD CODE!!!
    ''' does step one '''
    BAD CODE!!!
    BAD CODE!!!
    # code to do step four goes here
    BAD CODE!!!
    BAD CODE!!!
    stepTWO()

main()
```

15. There is actually a fourth way a programmer can get data in a function. What is it?

16. Does raw_input have a return statement in it?

17. State the four branching problem forms and the four looping problem forms.

18. State the python statements for each of the four branching problem forms and the four looping problem forms from Q17

19. What value must a variable used as a counter be initialized to? When?
    What is the update step for a counter variable? When?

20. Same questions for an accumulator

21. A module must contain a main def.

22. A module must not do any imports