1) While Loop Practice

In practice, while loops and for loops are used for two very different types of iteration. One being a definite number of iterations and the other being an uncertain number. However, knowing how to write various loops both ways is a useful way to practice. The function ‘forLoop’ operates on a list. Read the code to find out what it does, then write a function called ‘whileLoop’ that does the same thing using a while loop. The original list should not be modified.

```python
def forLoop(alist):
    new_list = []
    for each in alist:
        if type(each) is type("string"):  
            new_list.append(each[1:])
    return new_list
```
2) Sentinel Values

a. What does the program below do? (Hint: What is the output if the user enters 1.5 then 2.0 and then 2.5 and then 0.)

```python
def myfunction():
    t = 0
    d = 0
    c = True
    while c:
        x = float(input('Enter a floating point value (0 when done): '))
        if x != 0:
            t = t + x
            d = d + 1
        else:
            c = False
    a = t / d
    return(a)

print(myfunction())
```

b. Write a function called ‘getAges’, that continuously prompts the user for ages of some set of individuals. When the user wants to stop entering ages, there should be an appropriate value they can enter to stop the prompting. The function should return a list of the entered ages.
3) Validating Input

a. Write a function called get_five_letter_word() that prompts the user for a word, checks that the word has five letters, and returns the word if it does. If not, it prompts the user to try again until the correct input is entered.

b. What could be a problem with the function above?

c. How can we fix it? Write the code.
4) Putting it all together:

a. Write a function called isbetween_bounds that tests if a passed integer value is between two bounds (inclusive). The function should take one integer value \( x \) and one list of two integers with the first integer being the lower bound and the second being the upper bound as parameters. The function should return a Boolean value.

b. Write a function called get_valid_user_guess() that takes a list of two integers as a parameter and prompts the user for a number guess between a lower bound and an upper bound, if the user enters an integer that is not between those bounds it prompts the user for a valid guess and continues to do so until the user enters a valid value. You should use the function you wrote above.
c. Write a guess_the_number program:

Use the functions written above and use a while loop for the main part of the game. The program picks a secret number and prompts the user to try to guess the number. The program provides feedback as to whether the users guess was too high or too low. The user has a maximum of three guesses to correctly guess the number. The program should not count invalid guesses as one of the player’s three guesses.