

COMP500 / ENSE501: Week 7 – Exercise:

EXERCISE NAME: *Letter Guessing Game*

This program is a letter guessing game where a simple AI opponent generates a secret letter for the user to guess. The user must be able to take turns guessing the secret letter, with the AI responding to the user's guesses. After successfully guessing, the user must be allowed to play again. The AI must count how many turns the user has taken.

Here is an example of a game in progress where the human user has incorrectly guessed the letter 'M' on their first turn:

```
BUZZ, HUM, ZAP... Powering up the P1-AI-Bot...

AI: Welcome to "The 'Basic' Letter Guessing Game"!
AI:
AI: Here we go! ...abcdefghijklmnopqrstuvwxyz...
AI: I'm thinking of a secret letter, can you guess it...
AI: Turn 1, what is your guess?

Human: M

AI: Wrong... my secret letter comes before M in the alphabet...
AI: Turn 2, what is your guess?

Human:
```

When the program starts, it must print the following:

```
BUZZ, HUM, ZAP... Powering up the P1-AI-Bot...

AI: Welcome to "The 'Basic' Letter Guessing Game"!
AI:
```

After this, the AI must print the following messages:

```
AI: Here we go! ...abcdefghijklmnopqrstuvwxyz...
AI: I'm thinking of a secret letter, can you guess it...
```

Carefully review the first line of the simple AI's message output above, it contained the following:

...abcdefghijklmnopqrstuvwxyz...

This simple AI is somewhat poor, as when it outputs the lower-case alphabet, it also highlights what its randomly chosen secret letter is by making it upper-case! Notice the upper-case 'L'! A keen player may notice this (and a programmer can also use this knowledge when testing their implementation).

Next the game must begin, where the simple AI will prompt the user with the current turn number and for the user to input their guess. The user must then be able to input their single character guess:

```
AI: Turn 1, what is your guess?  
Human: M
```

Ensure the program accepts both upper- and lower-case input from the user in all input scenarios.

When the user's response is wrong, the AI must respond appropriately. If the guess comes before the secret letter in the alphabet, the simple AI must respond as follows:

```
AI: Wrong... my secret letter comes after M in the alphabet...
```

If the guess comes after the secret letter in the alphabet, the AI must respond as follows:

```
AI: Wrong... my secret letter comes before M in the alphabet...
```

After this, the user must then be prompted to take their next turn, and make another guess:

```
AI: Turn 2, what is your guess?  
Human:
```

This pattern of play must continue until the user guesses correctly.

When the user guesses correctly, the AI must respond as follows and prompt the user whether they want to play again, or not, in the following style:

```
Human: L  
  
AI: Well done!  
AI: You took 12 turns! Game over!  
AI: Play again (y/n)?
```

Ensure the "You took" some number of "turns!" is sensitive to plurals, and appropriately includes or excludes the 's' on turns depending on the number of turns taken!

When the user selects “yes” they want to play again, a new round must start:

```
AI: Play again (y/n)?
Human: y

AI: Here we go! ...abcdefghijklmnopqrstuvwxyz...
AI: I'm thinking of a secret letter, can you guess it...
AI: Turn 1, what is your guess?
Human:
```

When the user selects “no”, the following message is displayed before the program exits:

```
AI: Play again (y/n)?
Human: n

AI: Thanks for playing 2 rounds with me!
AI: G
AI:  o
AI:  o
AI:   d
AI:   b
AI:    y
AI:    e
AI:
AI:     H
AI:     u
AI:     m
AI:     a
AI:     n
AI:     !

Powering down the P1-AI-Bot... FIZZ, POP, BANG!
```

Ensure the final “Thanks for playing” some number of “rounds with me” is sensitive to plurals, and appropriately includes or excludes the 's' on rounds depending on the number of rounds played!

The program must also validate the user's input.

When the user inputs a digit, the program must respond as follows:

```
AI: Turn 1, what is your guess?  
Human: 5  
  
AI: Wrong... 5 is a digit, that is not a letter!  
AI: Turn 2, what is your guess?  
Human:
```

Another example is as follows:

```
AI: Turn 8, what is your guess?  
Human: 7  
  
AI: Wrong... 7 is a digit, that is not a letter!  
AI: Turn 9, what is your guess?  
Human: 0  
  
AI: Wrong... 0 is a digit, that is not a letter!  
AI: Turn 10, what is your guess?  
Human:
```

When the user input is not a letter or a digit, the program must respond as follows:

```
AI: Turn 10, what is your guess?  
Human: %  
  
AI: Wrong... wow, % is not a letter!  
AI: Turn 11, what is your guess?  
Human: *  
  
AI: Wrong... wow, * is not a letter!  
AI: Turn 12, what is your guess?  
Human: [  
  
AI: Wrong... wow, [ is not a letter!  
AI: Turn 13, what is your guess?  
Human:
```

An example of the completed program, with user input:

```
BUZZ, HUM, ZAP... Powering up the P1-AI-Bot...

AI: Welcome to... "The 'Basic' Letter Guessing Game"!
AI:
AI: Here we go! ...abcdefghijklmnopqrstuvwxyz...
AI: I'm thinking of a secret letter, can you guess it...
AI: Turn 1, what is your guess?

Human: r

AI: Wrong... my secret letter comes after R in the alphabet...
AI: Turn 2, what is your guess?

Human: W

AI: Wrong... my secret letter comes before W in the alphabet...
AI: Turn 3, what is your guess?

Human: u

AI: Wrong... my secret letter comes before U in the alphabet...
AI: Turn 3, what is your guess?

Human: t

AI: Well done!
AI: You took 4 turns! Game over!
AI: Play again (y/n)?

Human: n

AI: Thanks for playing 1 round with me!
AI: G
AI:  o
AI:  o
AI:  d
AI:  b
AI:  y
AI:  e
AI:
AI:      H
AI:      u
AI:      m
AI:      a
AI:      n
AI:      !

Powering down the P1-AI-Bot... FIZZ, POP, BANG!
```

Another example of the completed program, with user input:

```
BUZZ, HUM, ZAP... Powering up the P1-AI-Bot...

AI: Welcome to... "The 'Basic' Letter Guessing Game"!
AI:
AI: Here we go! ...Abcdefghijklmnopqrstuvwxyz...
AI: I'm thinking of a secret letter, can you guess it...
AI: Turn 1, what is your guess?

Human: A

AI: Well done!
AI: You took 1 turn! Game over!
AI: Play again (y/n)?

Human: y

AI: Here we go! ...abcdeHijklmnopqrstuvwxyz...
AI: I'm thinking of a secret letter, can you guess it...
AI: Turn 1, what is your guess?

Human: H

AI: Well done!
AI: You took 1 turn! Game over!
AI: Play again (y/n)?

Human: n

AI: Thanks for playing 2 rounds with me!
AI: G
AI:  o
AI:   o
AI:    d
AI:     b
AI:      y
AI:       e
AI:
AI:        H
AI:         u
AI:          m
AI:           a
AI:            n
AI:             !

Powering down the P1-AI-Bot... FIZZ, POP, BANG!
```

One final requirement is that the simple AI must politely insult the player if they have taken more than 25 guesses at the secret letter as follows:

```
AI: Turn 24, what is your guess?
Human: a
AI: Wrong... my secret letter comes after A in the alphabet...
AI: Turn 25, what is your guess?
Human: b
AI: Wrong... my secret letter comes after B in the alphabet...
AI: There are only 26 letters in the alphabet... are you okay?
AI: Turn 26, what is your guess?
Human: c
AI: Wrong... my secret letter comes after C in the alphabet...
AI: There are only 26 letters in the alphabet... are you okay?
AI: Turn 27, what is your guess?
Human: d
AI: Wrong... my secret letter comes after D in the alphabet...
AI: There are only 26 letters in the alphabet... are you okay?
AI: Turn 28, what is your guess?
Human:
```

You must choose a **different polite insult** to occur after 25 guesses – do not use the example: “There are only 26 letters in the alphabet... are you okay?” – choose your own unique message!

Based upon the program requirements set out above, start by creating a pseudo code design.

Declare and define at least five functions in addition to the **main** function. Do not use global variables or **goto**!

Once you have your pseudo code design, implement your design using C. Finally, robustly test your implementation to ensure it matches the requirements.

Follow good programming standards for code layout whitespace, naming and commenting. Ensure your C source code can successfully compile. Test your program with a variety of input and ensure the resulting program output is as described above.