Programming Fundamentals

Tutorial 8: Strings

Note: You are not allowed to copy-paste code snippets. You need to have a basic understanding of following concepts at the end of the Tutorial

- If – else
- While/FOR
- String functions

1. Use Pycharm integrated development environment (IDE) to write and run the following programs. If it not installed, you can use Python IDLE.

2. You are asked to develop a game called GUESS THE WORD. First, read the requirement carefully and try to understand.

   What is GUESS THE WORD?
   Here the program hids a specific word and the user needs to guess the hidden word However it is not possible to guess the whole word without guessing each character right? So the user will be given several turns to predict the word by guessing character by character. If the user guesses all characters correctly or the number of turns/attempts runs out, the program terminates.

   Initial Step
   1. First, Store the secret word in a variable
   2. Set the number of turns/attempt. Example: set as 6
   3. Create an empty variable to store the guess. Initialize it as “”
   4. Introduce the game to the user
   5. Initially introduce the number of characters in the secret as follows
      a. Example if the length of the word is 6, displays it as **_ _ _ _ _ _ .** This will help the user to understand the number of characters in the secret. Also, store this pattern in a variable called “**storage**”. This is the variable you are modifying in the next step
      HINT: You are allowed to use * operator with python strings

Sample output:

```
Let's play Guess the Word
You have 6 turns to guess the word!
_ _ _ _ _ _ _ _ _ _ _ #if the length of the secret is 11
```

Stage 1

1. Ask the user to enter a character
2. Check whether the secret contains the entered character
3. If it contains, finds the location of the character in the **storage** and, replaces the "_" with the specific character.
4. Otherwise, no modifications

Hint: You can use a for loop to iterate through the string

```
for letter in secret:
    if the guess == letter
            #code
```

Further, if you want to modify the **storage** string, use string slicing concept

Hint:
storage=storage[:location_of_character] + guess+ storage[location_of_character + 1:]

Sample output:

```
Let's play Guess the Word
You have 6 turns to guess the word!
_ _ _ _ _ _ _ _ _ _ _
Guess a letter: w
w _ _ _ _ _ _ _ _ _ _
```

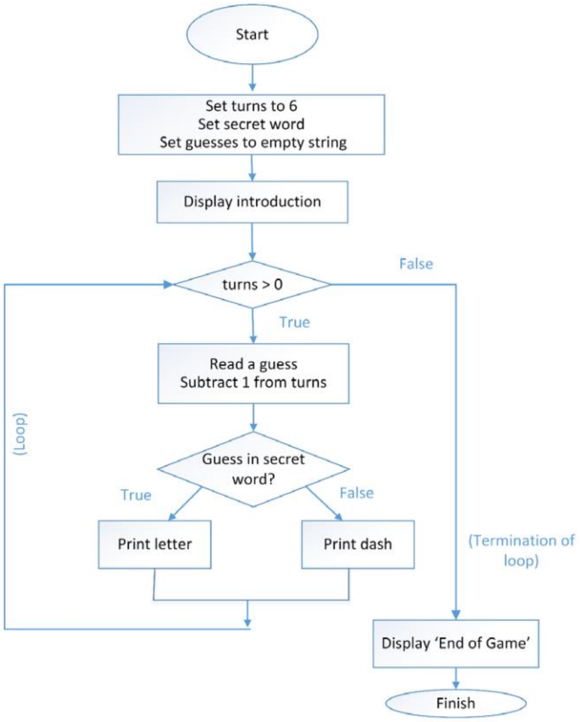
Stage 2

1. Now, the user needs to be able to enter multiple guesses. Use a loop with an appropriate condition to limit the number of turns/attempts.
   Sample output:

```
Let's play Guess the Word
You have 6 turns to guess the word!

_ _ _ _ _ _ _ _ _ _ _
Guess a letter: e
_ e _ _ _ _ _ _ _ e _
Guess a letter: i
_ e _ _ _ i _ _ _ e _
Guess a letter: w
w e _ _ _ i _ _ _ e _
Guess a letter: x
w e _ _ _ i _ _ _ e _
Guess a letter: y
w e _ _ _ i _ _ _ e _
Guess a letter: z
w e _ _ _ i _ _ _ e _
```

The following flowchart explains the flow of the program. Please examine it carefully before you start writing the code.



Additional

1. The program should base the number of turns based on the length of the word

2. What if the string contains multiple occurrences of a character (several **e** or **a**) and the, program does not consider replacing all placement using a single entry. What changes need to be done?