DAILY ASSESSMENT FORMAT

Date:	4 June 2020	Name:	Karegowda kn
Course:	LOGIC DESIGN	USN:	4al16ec029
Topic:	• Applications of Programmable logic controllers	Semester & Section:	6th sem & B sec
Github Repository:	Karegowda-courses		

FORENOON SESSION DETAILS

Report –

Applications of Programmable logic controllers:

- • High-Performance Controllers in a Compact, Secure Package Today's industrial applications require faster performance and more reliable connections.
- • Emerson's Programmable Automation Controllers feature an extensive range to support scalable automation and minimize downtime.
- • Redundant by design, these compact controllers use PROFINET for better performance and productivity, and are interoperable with most open industry standards. Rugged, fanless design means more durability and better performance in any environment.
- • In the most basic terms, a programmable logic controller (PLC) is a computer with a microprocessor but has no keyboard, mouse or monitor. It is essentially built to withstand very harsh industrial environments.
- • A PLC performs only a single set or sequence of tasks, with greater reliability and performance, except when it is under real-time constraints. This is in contrast to regular PCs and smartphones that are designed to execute any number of roles simultaneously within the Windows framework.
- • The PLC has a number of features that you don't find in normal computers, such as protection from the open area conditions like heat, dust and cold.

- • It is low cost compared with other microcontroller systems. When you’re using a PLC in various applications, you only need to change the software component for each application

  Attended the Live Bonus Session on “LIVE_SIMPLIFYING THE BRAIN”

Date:	5 June 2020	Name:	Karegowda kn
Course:	Python	USN:	4al16ec029
Topic:	Application 1: Build an interactive English dictionary.	Semester & Section:	6th sem & B sec

AFTERNOON SESSION DETAILS

Report –

1.Tacking into account bad words

```
import json
data = json.load(open("data.json"))
def translate (w):
    if w in data:
      return data(w)
    else:
      return "the word doesn't exist. please double-check it."
word = input("enter word: ")
print (translate (word))
```

2. Implementing case sensitivity

```
import json
data = json.load(open("data.json"))
def translate (w):
    w = w.lower
    if w in data:
      return data(w)
    else:
      return "the word doesn't exist. please double-check it."
word = input("enter word: ")
print (translate (word))
```

3.similarity ratio between two words

Expalin the similarity between two words. Like "rain" and "rainn" it has a similarity of 0.8888888. so we cansider it to rain.