Dayananda Sagar College of Engineering

Department of Electronics and Communication Engineering

Assignment

Program:	B.E.	**Semester:**	6
Course :	Python Programming	**Section:**	A
Course Code:	19CE6IEPYP	**Date:**	June 20, 2022

A Report on

Analysis & Automation of Data and display in Graphical Representation

Submitted by:

USN	Name
1DS19EC003	Abhishek Singh
1DS19EC010	Anish Ingale
1DS19EC012	Anmol Singh
1DS19EC015	Arpit Kumar

Faculty In-charge

Prof. Deepa NP

Signature of Faculty In-charge

Analyze & Automate Data and display in Graphical Representation (Use Case 15)

Work Breakdown:

S.No.	Task	Input	Processing Step	Output
1	Graphical Representation Memory consumption Vs Time with HMI Views loaded	Log file	identify no. of HMI views loaded identify memory consumption for each HMI view loaded and plot graph with time	Graphical Display Refer below graph

Keywords for referring log file

Keywords
View Displayed: view loaded **Memory consumption:** ebUIMemoryReport**(Consider Stack, Heap, Total)** **IOD (Information On Demand) View =** ebUIIodView

More Details

Input: Log File (.txt file) – Attached in mail
Output: Graphical Representation – Refer image below

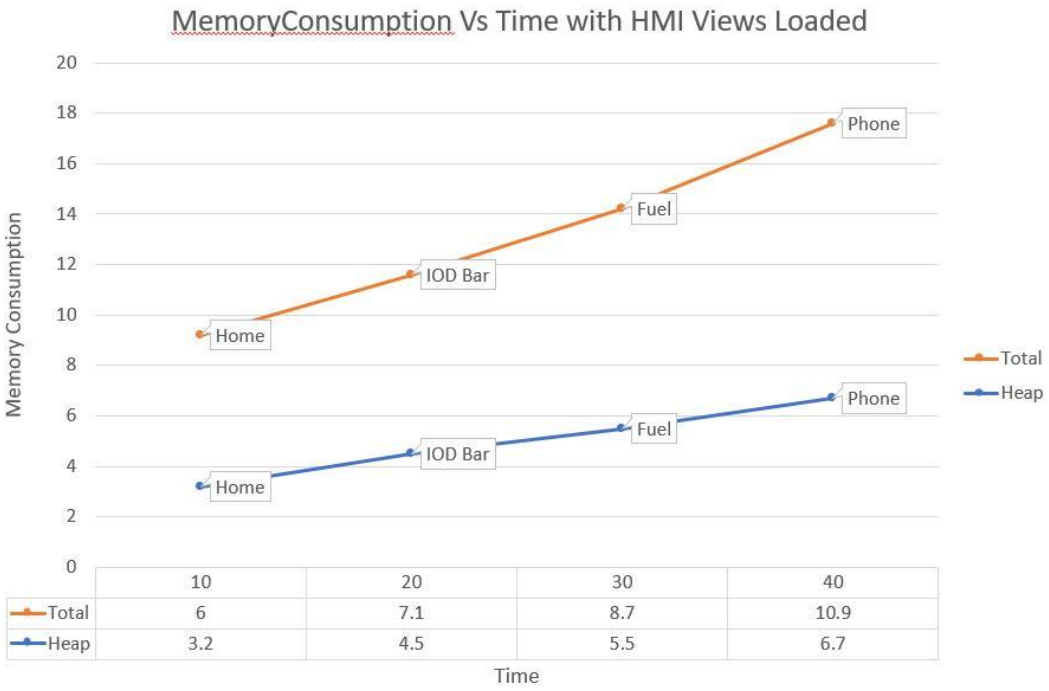
Sample Graph:

X-Axis – Time
Y-Axis - Memory Consumption (Heap and Total)

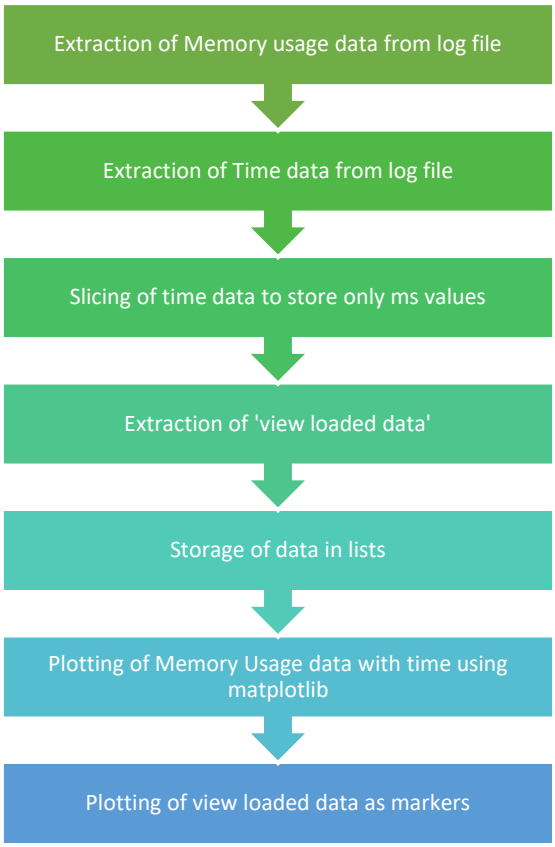
Note:

Consider Heap and Total for Memory Consumption
Plot two lines in one graph with different colors one each for Heap and Total
Heap & Total graph lines should display the HMI views/popup/IOD view loaded

Graphical Representation should be similar to below one

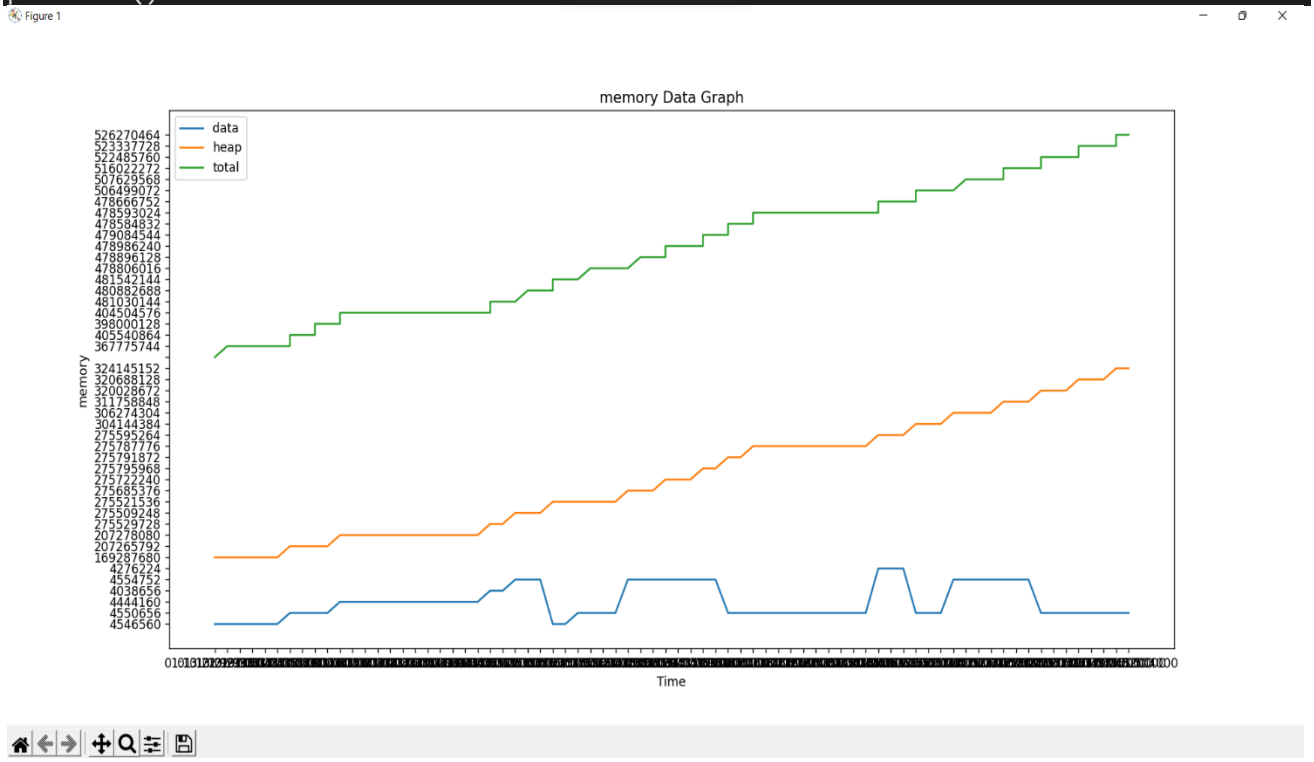

MemoryConsumption Vs Time with HMI Views Loaded
Memory Consumption
Time
Home
IOD Bar
Fuel
Phone
Total
Heap
10
20
30
40
Total
6
7.1
8.7
10.9
Heap
3.2
4.5
5.5
6.7

Flowchart:


Extraction of Memory usage data from log file
Extraction of Time data from log file
Slicing of time data to store only ms values
Extraction of 'view loaded data'
Storage of data in lists
Plotting of Memory Usage data with time using matplotlib
Plotting of view loaded data as markers

```
from cProfile import label
import re
import matplotlib.pyplot as plt
import numpy as np

inline="C:\\Users\\abhis\\OneDrive\\Desktop\\project\\Sample_Logs_PythonUseCase_ToSh
are.txt"#taking input
time_data=[]
time_total=[]
time_heap= []
data=[]
total=[]
heap=[]
vL=set()
pas_data=0
pas_heap=0
pas_total=0
pas_vL=0
key=""
value_data=""
value_heap=""
value_total=""
value_vL=""

with open(inline) as f:
    f = f.readlines() #reading line from log file
for line in f:
    words=line.split() #spliting words from line
    for word in words:
        if re.search("^2000-01-01T01", word):# searching for word starting from
2000-01-01T01
            word=word[11::]
            word=word.rstrip('Z') #striping 'Z' from time stamp
            key=word   #storing word as key
        elif (pas_data):
            pas_data=0
            value_data=word  #taking word next to data=
        elif (pas_heap):
            pas_heap=0
            value_heap=word #taking word next to heap=
        elif (pas_total):
            pas_total=0
            value_total=word #taking word next to total=
        elif (pas_vL):
            pas_vL=0
            value_vL=word   #taking word next to view loaded
        elif re.search("^loaded:",word):
```

```
                pas_vL=1
            elif (word == ("data=")):
                pas_data=1
            elif (word == ("heap=")):
                pas_heap=1
            elif (word == ("total=")):
                pas_total=1
        if(value_data !=" "):
            time_data.append(key)
            data.append(value_data)
        if(value_heap !=" "):
            time_heap.append(key)
            heap.append(value_heap)
        if (value_total !=" "):
            time_total.append(key)
            total.append(value_total)
        if(value_vL !=" "):
            vL.add(value_vL)

print(vL)
vLL=list(vL)
plt.plot(time_data,data, label="data")
plt.plot(time_total,marker=vLL)
plt.plot(time_heap,heap, label="heap")
plt.plot(time_total,total, label="total")
plt.plot(time_total,value_vL, label="loaded")

plt.legend()
plt.show()
```



References:

Barrett, Paul & Hunter, J. & Miller, J.T. & Hsu, J.-C & Greenfield, P.. (2005). matplotlib -- A Portable Python Plotting Package.

N. Ari and M. Ustazhanov, "Matplotlib in python," 2014 11th International Conference on Electronics, Computer and Computation (ICECCO), 2014, pp. 1-6, doi: 10.1109/ICECCO.2014.6997585.

https://www.w3schools.com/

https://stackoverflow.com/

https://www.python.org/