Data Analysis

Aim:

1. To read the data from a converge engine data file.
2. Take column numbers as input and plot the respective plots
3. Each file should be saved by the name of the column.
4. Labels should be created automatically for the plots requested.
5. Exit if not proper extension file is imported to read.
6. Calculate area under the PV diagram.
7. Power output of the engine at 1500 RPM.
8. Calculate specific fuel consumption if the engine consumes 10 micrograms of fuel per 1 stroke cycle.

Solution:

Step 1: Import the mathematical models
Step 2: Request for the file name to be opened.
Step 3: if .out file is not present, close the program.
Step 4: Create empty arrays to store the values after reading the numerical data.
Step 5: Start reading the lines which does not have '#' comments and append the values into the empty arrays.
Step 6: Store all the arrays into a single master array for easy reference.
Step 7: Start reading the lines which contains '#' and accept only 3rd and 4th row,since they contain the legend and units.
Step 8: Now ask for user input in terms of x and y axis.
Step 9: Plot the graph using the numerical data and also the units obtained from commented lines.
Step 10: Use np.trapz to compute the area under a curve and import the formulae to calculate power and specific fuel consumption.

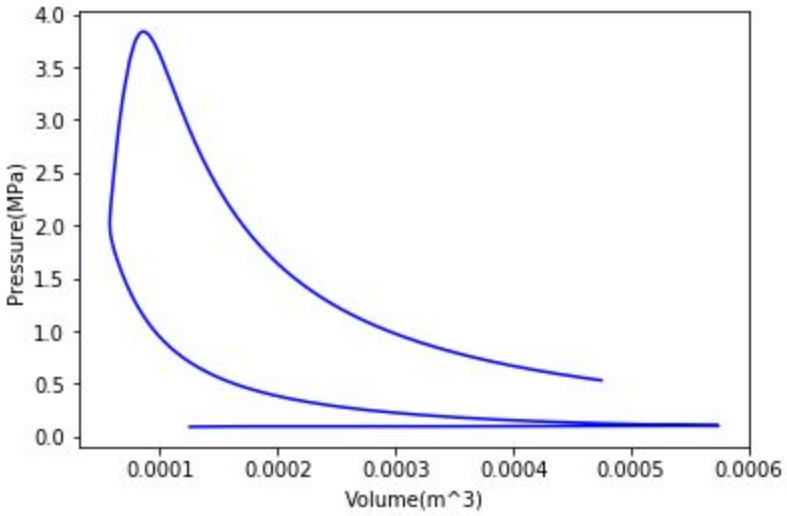
Program:

```
import math
import matplotlib.pyplot as plt
import sys
import numpy as np

name=input("Enter file name:")
if '.out' not in name:
    print ("File not recognized. Please provide a valid CONVERGE
output file")
    sys.exit()
else:
    #empty array for storing the parameters after reading
    Crank=[]
    Pressure=[]
    Max_pres=[]
    Min_pres=[]
    Mean_pressure=[]
    Max_temp=[]
    Min_temp=[]
    Mean_temp=[]
    Volume=[]
    Mass=[]
    Density=[]
    Integrated_HR=[]
    HR_rate=[]
    C_p=[]
    C_v=[]
    Gamma=[]
    Kin_Visc=[]
    Dyn_Visc=[]

for line in open(name):
    if '#' not in line: #eliminating the comments and reading the
data, appending it to the empty arrays

        Crank.append(float(line.split()[0]))
        Pressure.append(float(line.split()[1]))
        Max_pres.append(float(line.split()[2]))
        Min_pres.append(float(line.split()[3]))
        Mean_temp.append(float(line.split()[4]))
        Max_temp.append(float(line.split()[5]))
```

```
        Min_temp.append(float(line.split()[6]))
        Volume.append(float(line.split()[7]))
        Mass.append(float(line.split()[8]))
        Density.append(float(line.split()[9]))
        Integrated_HR.append(float(line.split()[10]))
        HR_rate.append(float(line.split()[11]))
        C_p.append(float(line.split()[12]))
        C_v.append(float(line.split()[13]))
        Gamma.append(float(line.split()[14]))
        Kin_Visc.append(float(line.split()[15]))
        Dyn_Visc.append(float(line.split()[16]))

 #master array to contain all the parameters
parameters=([Crank,Pressure, Max_pres, Min_pres, Max_temp, Min_temp,
Mean_temp, Volume, Mass, Density, Integrated_HR, HR_rate, C_p, C_v,
Gamma, Kin_Visc, Dyn_Visc])
#Now considering comment lines for extracting legends for units
count=1

for line in open(name):
    if '#'in line:
        if count==3: #line 3 contains data header
            legend=(line.split()[0:17])
        if count==4: #line 4contains units
            units=(line.split()[0:17])
        count=count+1
#request for user input
x=int(input('Enter value for x-axis:'))
y=int(input('Enter value for y-axis:'))

 #plotting
plt.plot(parameters[x-1], parameters[y-1], color='blue')  # -1 is
done since python has zero indexing
plt.xlabel(legend[x]+units[x])
plt.ylabel(legend[y]+units[y])

'''Area and Power calculation'''
#to find the area under pv
area=np.trapz(parameters[1],parameters[7])
print('Area under PV = ', area)

rpm=1500
power=(area*rpm*1000)/60  #to convert rpm to rps
```

```
print('Power=',power,'kW')

'''Specific Fuel consumption'''
fuel=(20*pow(10,-6)*3600)/(0.08*power)
print('Specific fuel consumption =',fuel,'g/kWh')
```

Results:




```
Enter file name:engine_data.out

Enter value for x-axis:8

Enter value for y-axis:2
Area under PV =  0.0005005866274680075
Power= 12.514665686700187 kW
Specific fuel consumption = 0.07191562463841637 g/kWh
```

Thus the output PV diagram, Area under PV dia, power and specific fuel consumption are computed.