Internal Load Distributor Tool User Guide - Version 1.0

TABLE OF CONTENTS

1. INTRODUCTION

The purpose of the Load Distributor Tool is to propose an optimal load distribution of assays/tests to analyzers on a site to meet the following goals:

- Achieve balanced load distribution among instruments on a site
- Reduce the number of quality control tests
- Not to exceed maximum capacity constraints on instruments
- Satisfy the assay merge rules:
  - Measles IgG and Mumps IgG
  - Toxo IgG and Toxo IgM
  - HSV-1 IgG and HSV-2 IgG
  - CMV IgG and CMV IgM
  - EBV IgG (EA IgG), EBV IgM and EBNA IgG

  Merge rules target to reduce labor, supply, QC and disposal costs by combing assays.
- Group the assays that require clean procedure (IgM assays) on the same instruments
  - IgM assays that require clean procedure: Toxo IgM, CMV IgM, EBV IgM, Rubella IgM

  IgM assays require Clean procedure to be run on the instruments. IgG assays do not require clean procedure. Therefore grouping IgM assays on the same instrument allows minimizing the number of instruments that run Clean procedure.

The user inputs of the program are:

- The number of analyzers on the site
- The maximum daily test capacity on instruments
- The load balancing upper bound
- The volume of billable tests for each assay type

Each of these inputs will be explained in more detail in the following sections. The rest of this document describes step-by-step instruction on how to use this tool.

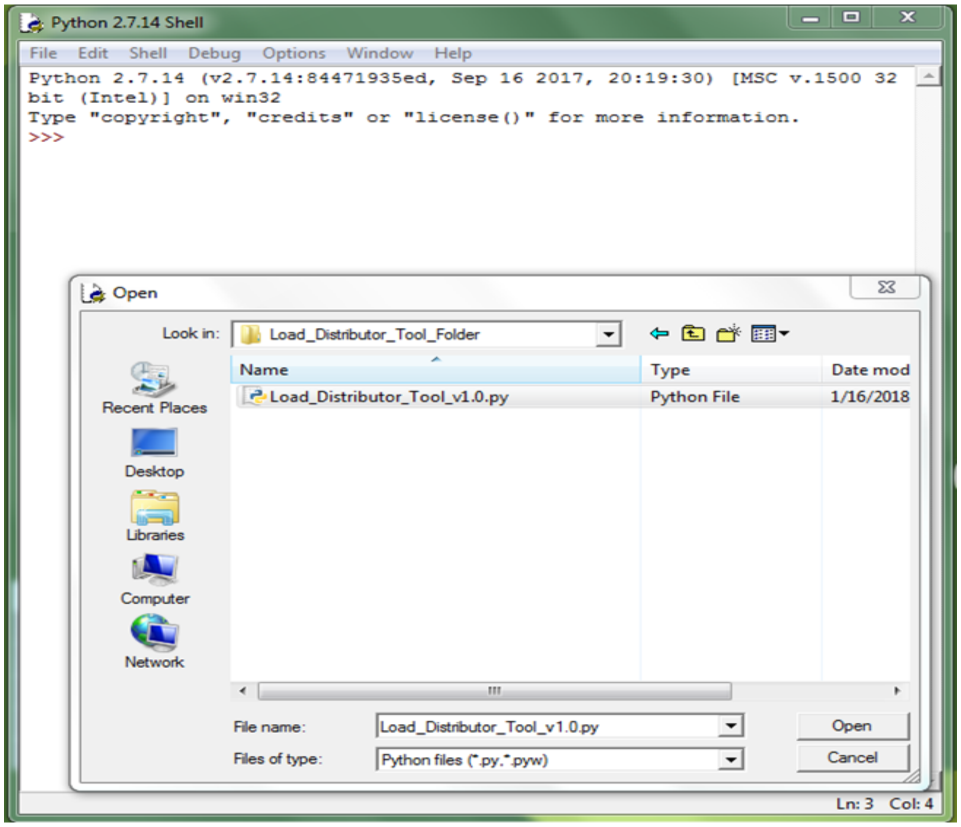
2. USER INSTRUCTIONS

The steps described in the chapter refer to the procedures, after the installation of Python 2.7 environment and Python libraries is completed on the computer, as explained in the Appendix section of this document.

2.1 Launching the Load Distributor Script

Step-1: Place the “Load_Distributor_Tool.py” file into any location in your computer.

Step-2: Launch Python GUI on your Windows computer as described in the Appendix section. Then go to "File → Open" and select the "Load_Distributor_Tool.py" file on the Python GUI shown in the screenshot below.


```
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:19:30) [MSC v.1500 32
bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

Look in: Load_Distributor_Tool_Folder

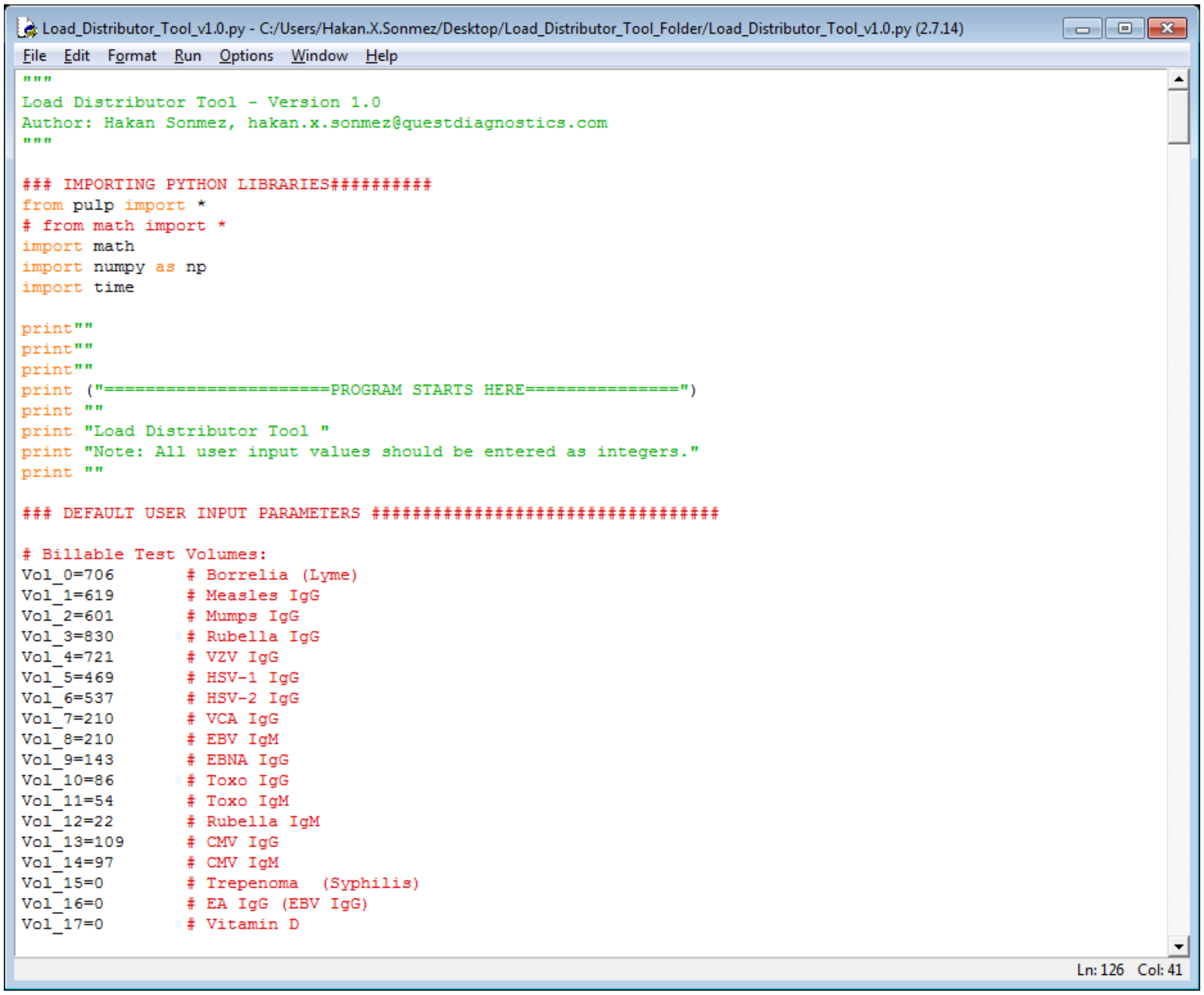
Name	Type	Date mod
Load_Distributor_Tool_v1.0.py	Python File	1/16/2018

File name: Load_Distributor_Tool_v1.0.py
Files of type: Python files (*.py,*.pyw)

Figure: Opening the script on the Python GUI

In the example screenshot above, the "Load_Distributor_Tool.py" file is placed in a folder named "Load Distributor Tool Folder".

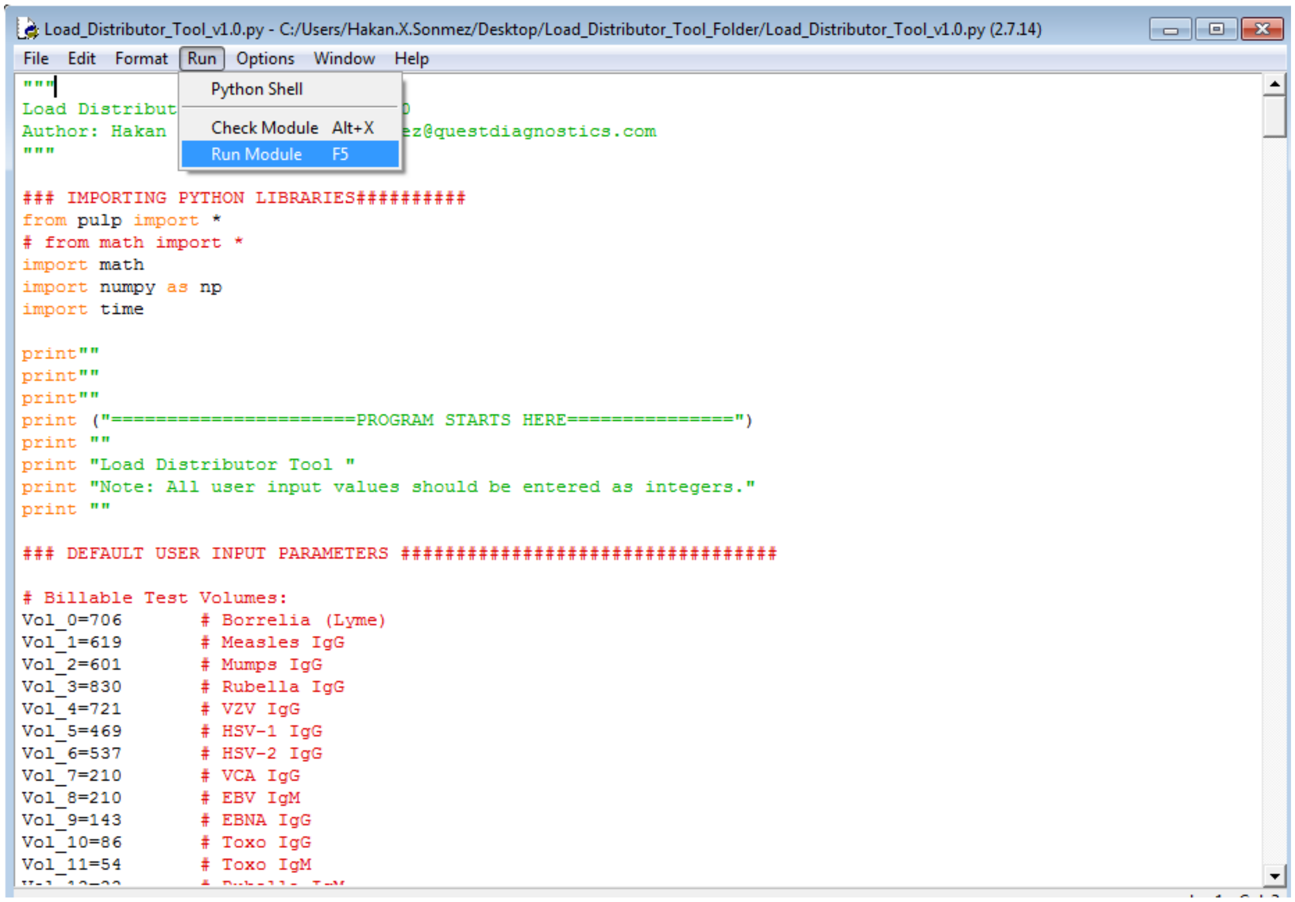
Step-3: After you open the" Load_Distributor_Tool.py file", you should see the Python script loaded on the Python GUI screen similar to the screenshot below.


Load_Distributor_Tool_v1.0.py - C:/Users/Hakan.X.Sonmez/Desktop/Load_Distributor_Tool_Folder/Load_Distributor_Tool_v1.0.py (2.7.14)
File Edit Format Run Options Window Help
Ln: 126 Col: 41

```
"""
Load Distributor Tool - Version 1.0
Author: Hakan Sonmez, hakan.x.sonmez@questdiagnostics.com
"""

### IMPORTING PYTHON LIBRARIES##########
from pulp import *
# from math import *
import math
import numpy as np
import time

print""
print""
print""
print ("======================PROGRAM STARTS HERE===============")
print ""
print "Load Distributor Tool "
print "Note: All user input values should be entered as integers."
print ""

### DEFAULT USER INPUT PARAMETERS ##################################

# Billable Test Volumes:
Vol_0=706       # Borrelia (Lyme)
Vol_1=619       # Measles IgG
Vol_2=601       # Mumps IgG
Vol_3=830       # Rubella IgG
Vol_4=721       # VZV IgG
Vol_5=469       # HSV-1 IgG
Vol_6=537       # HSV-2 IgG
Vol_7=210       # VCA IgG
Vol_8=210       # EBV IgM
Vol_9=143       # EBNA IgG
Vol_10=86       # Toxo IgG
Vol_11=54       # Toxo IgM
Vol_12=22       # Rubella IgM
Vol_13=109      # CMV IgG
Vol_14=97       # CMV IgM
Vol_15=0        # Trepenoma  (Syphilis)
Vol_16=0        # EA IgG (EBV IgG)
Vol_17=0        # Vitamin D
```

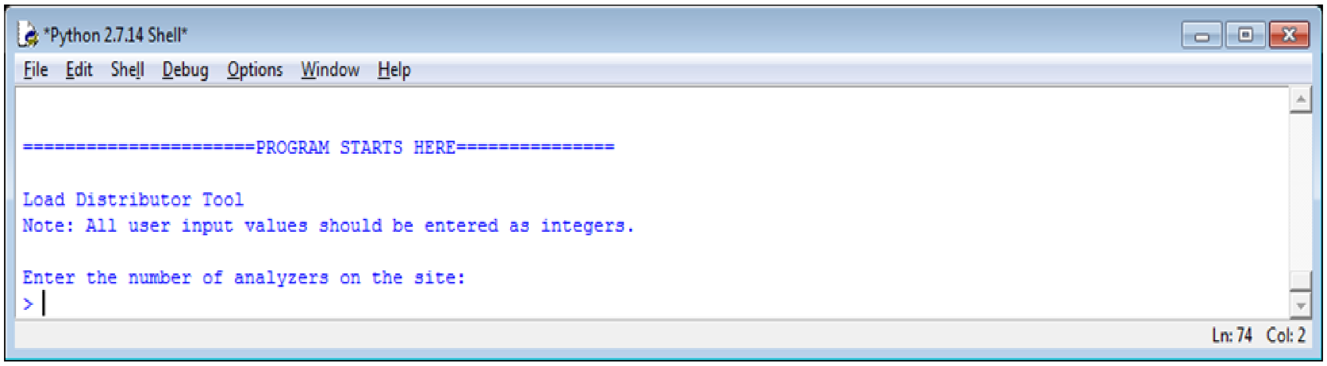
Figure: The script loaded on the Python GUI

Step-4: Go to" Run → Run Module" icon to run the script as shown in the screenshot below.


Load_Distributor_Tool_v1.0.py - C:/Users/Hakan.X.Sonmez/Desktop/Load_Distributor_Tool_Folder/Load_Distributor_Tool_v1.0.py (2.7.14)
File Edit Format Run Options Window Help
Python Shell
Check Module Alt+X
Run Module F5

```
"""
Load Distribut
Author: Hakan                  ez@questdiagnostics.com
"""

### IMPORTING PYTHON LIBRARIES##########
from pulp import *
# from math import *
import math
import numpy as np
import time

print""
print""
print""
print ("=====================PROGRAM STARTS HERE===============")
print ""
print "Load Distributor Tool "
print "Note: All user input values should be entered as integers."
print ""

### DEFAULT USER INPUT PARAMETERS ##################################

# Billable Test Volumes:
Vol_0=706       # Borrelia (Lyme)
Vol_1=619       # Measles IgG
Vol_2=601       # Mumps IgG
Vol_3=830       # Rubella IgG
Vol_4=721       # VZV IgG
Vol_5=469       # HSV-1 IgG
Vol_6=537       # HSV-2 IgG
Vol_7=210       # VCA IgG
Vol_8=210       # EBV IgM
Vol_9=143       # EBNA IgG
Vol_10=86       # Toxo IgG
Vol_11=54       # Toxo IgM
```

After this step, the program will start asking the inputs from the user as described in the next section.

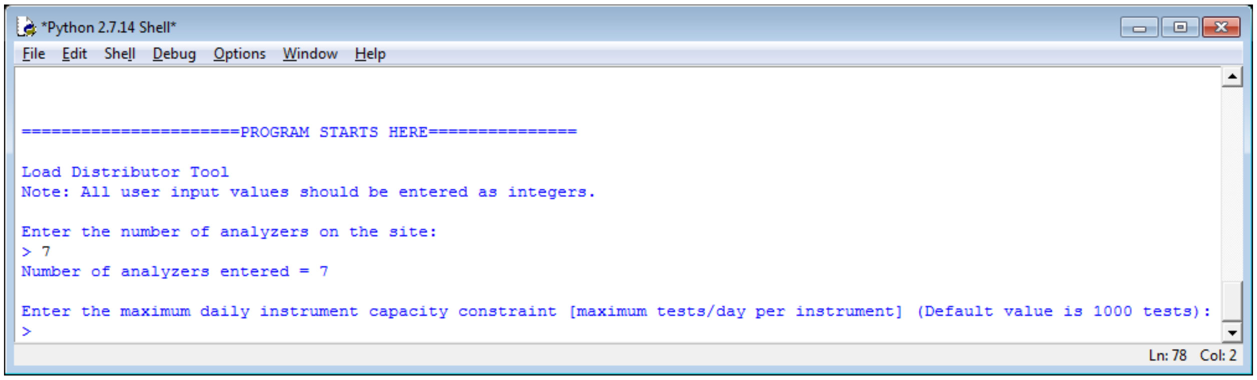


```
=====================PROGRAM STARTS HERE===============

Load Distributor Tool
Note: All user input values should be entered as integers.

Enter the number of analyzers on the site:
>
```

2.2. Entering Inputs

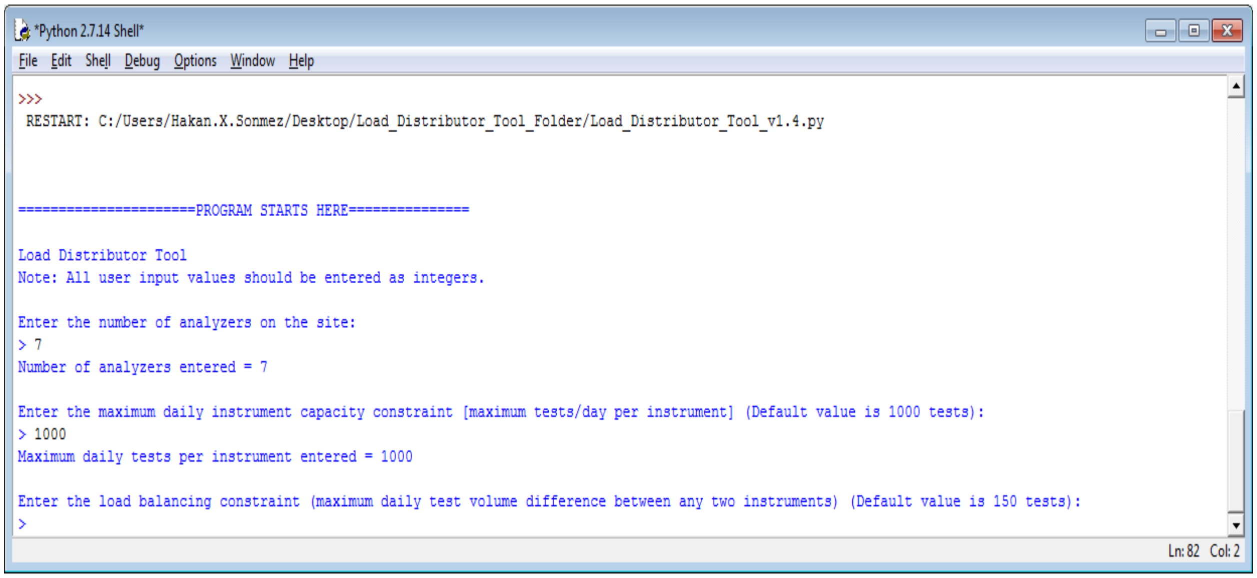
This section describes the instructions to enter user inputs and the explanation of each input. All inputs to the program should be entered as integer values.
If a non-integer value is entered by mistake (for example a decimal value or a string character), the program will exit and the user should go to the Step-4 of Section 2.1 and rerun the program.

Step-1: Entering the number of analyzers on the site
The user should enter the number of analyzers (instruments) available on the site. The load will be distributed over the number of analyzers that is entered in this step.



```
======================PROGRAM STARTS HERE===============

Load Distributor Tool
Note: All user input values should be entered as integers.

Enter the number of analyzers on the site:
> 7
Number of analyzers entered = 7

Enter the maximum daily instrument capacity constraint [maximum tests/day per instrument] (Default value is 1000 tests):
>
```

Step-2: Entering the maximum daily instrument capacity constraint
The user should enter the number of maximum daily tests allowed on one analyzer. This constraint is used to limit the number of tests that can be assigned to a single analyzer.



```
>>>
 RESTART: C:/Users/Hakan.X.Sonmez/Desktop/Load_Distributor_Tool_Folder/Load_Distributor_Tool_v1.4.py



=====================PROGRAM STARTS HERE==============

Load Distributor Tool
Note: All user input values should be entered as integers.

Enter the number of analyzers on the site:
> 7
Number of analyzers entered = 7

Enter the maximum daily instrument capacity constraint [maximum tests/day per instrument] (Default value is 1000 tests):
> 1000
Maximum daily tests per instrument entered = 1000

Enter the load balancing constraint (maximum daily test volume difference between any two instruments) (Default value is 150 tests):
>
```

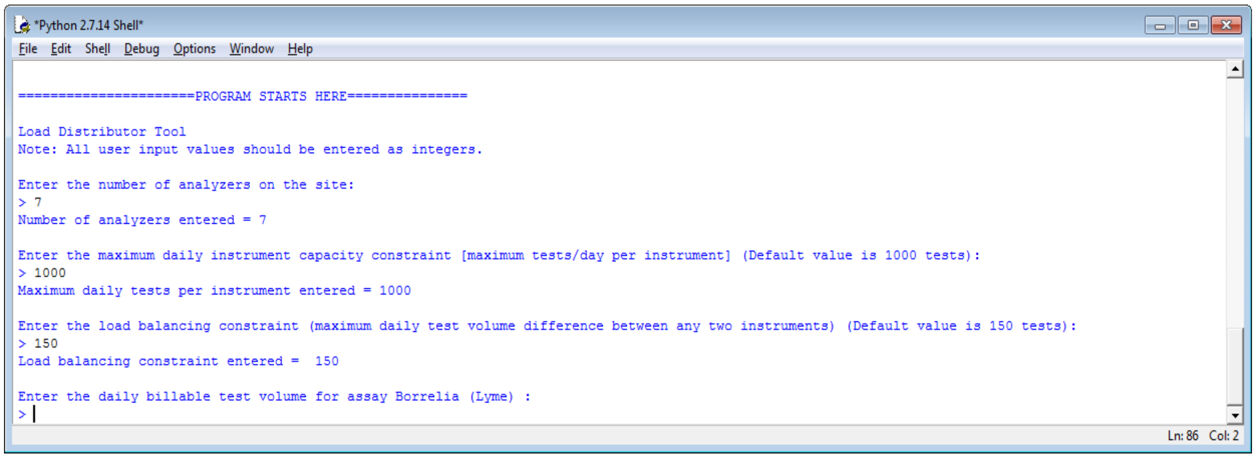
Step-3: Entering the load balancing constraint

This constraint is set to achieve load balancing among instruments. The value entered here will determine the maximum difference of daily test volumes between any two analyzers on the site.

For example, if 150 is entered for this constraint, the daily test volume difference between the instrument with the highest load and the instrument with the lowest load cannot be higher than 150 tests per day.

The load balancing constraint value entered by the user is critically important in the determination of an optimal solution and reducing the quality control (QC) test count. There is a trade-off between achieving load-balancing among instruments and reducing the QC counts.
If the load-balancing constraint is set too narrow, the total QC count will usually increase. This is due to the fact that the reduction of QC test count is achieved by consolidating the tests of an assay on the same instrument. Relaxing the load balancing constraint will help consolidation of test volumes and lowering total QC counts.

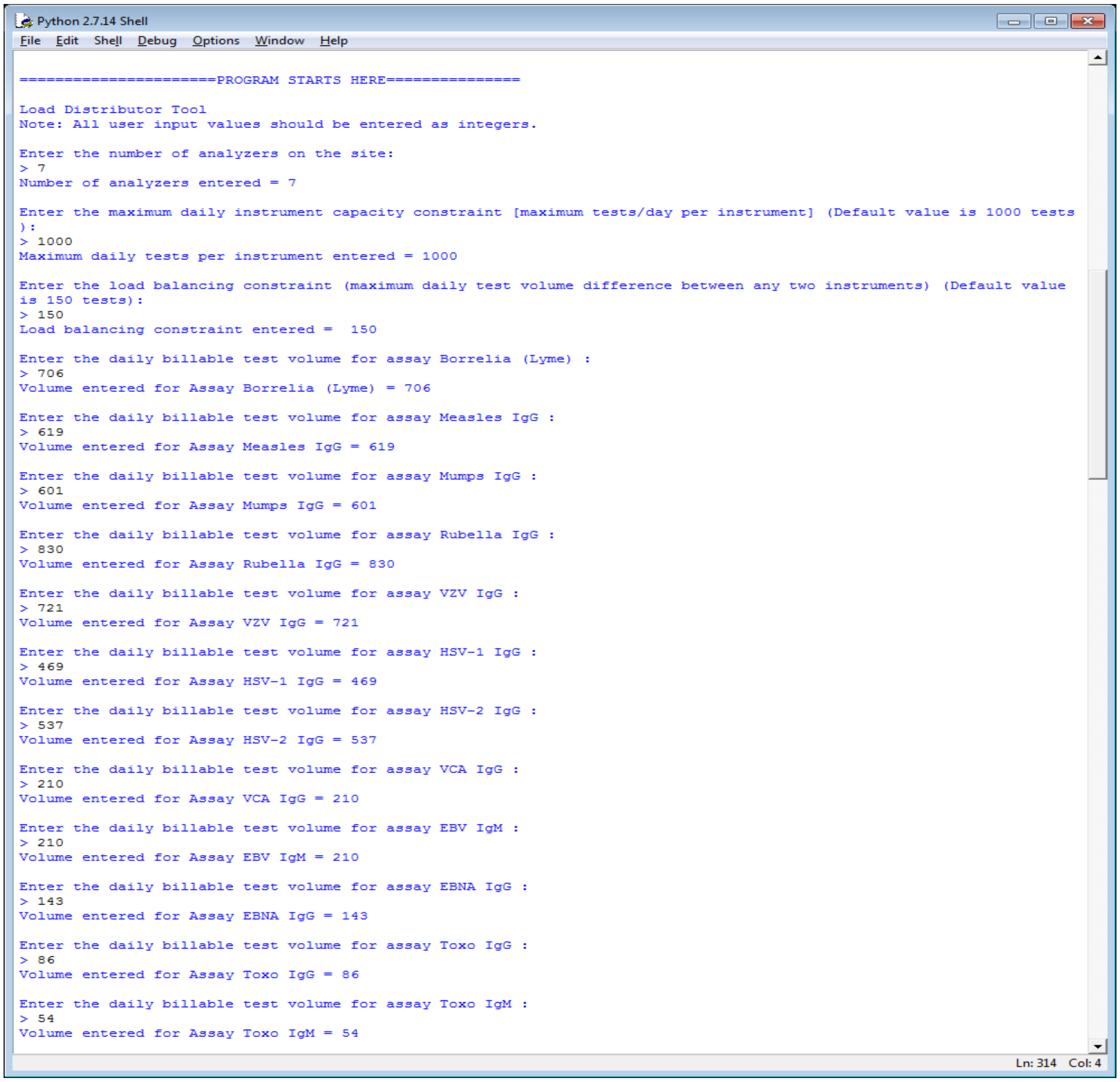
The suggested load balancing constraint is 150 tests/day or higher to achieve more consolidation and lower QC counts.



```
======================PROGRAM STARTS HERE===============

Load Distributor Tool
Note: All user input values should be entered as integers.

Enter the number of analyzers on the site:
> 7
Number of analyzers entered = 7

Enter the maximum daily instrument capacity constraint [maximum tests/day per instrument] (Default value is 1000 tests):
> 1000
Maximum daily tests per instrument entered = 1000

Enter the load balancing constraint (maximum daily test volume difference between any two instruments) (Default value is 150 tests):
> 150
Load balancing constraint entered =  150

Enter the daily billable test volume for assay Borrelia (Lyme) :
> 
```

Step-4: Entering Billable Test Volumes for Each Assay

The user enters the daily billable test volume for each assay one by one as shown in the screenshot below. If there is no volume for a specific assay (i.e. the particular assay is not tested on the site) , then “0” (zero) should be entered for that assay.



```
=====================PROGRAM STARTS HERE===============

Load Distributor Tool
Note: All user input values should be entered as integers.

Enter the number of analyzers on the site:
> 7
Number of analyzers entered = 7

Enter the maximum daily instrument capacity constraint [maximum tests/day per instrument] (Default value is 1000 tests
):
> 1000
Maximum daily tests per instrument entered = 1000

Enter the load balancing constraint (maximum daily test volume difference between any two instruments) (Default value
is 150 tests):
> 150
Load balancing constraint entered =  150

Enter the daily billable test volume for assay Borrelia (Lyme) :
> 706
Volume entered for Assay Borrelia (Lyme) = 706

Enter the daily billable test volume for assay Measles IgG :
> 619
Volume entered for Assay Measles IgG = 619

Enter the daily billable test volume for assay Mumps IgG :
> 601
Volume entered for Assay Mumps IgG = 601

Enter the daily billable test volume for assay Rubella IgG :
> 830
Volume entered for Assay Rubella IgG = 830

Enter the daily billable test volume for assay VZV IgG :
> 721
Volume entered for Assay VZV IgG = 721

Enter the daily billable test volume for assay HSV-1 IgG :
> 469
Volume entered for Assay HSV-1 IgG = 469

Enter the daily billable test volume for assay HSV-2 IgG :
> 537
Volume entered for Assay HSV-2 IgG = 537

Enter the daily billable test volume for assay VCA IgG :
> 210
Volume entered for Assay VCA IgG = 210

Enter the daily billable test volume for assay EBV IgM :
> 210
Volume entered for Assay EBV IgM = 210

Enter the daily billable test volume for assay EBNA IgG :
> 143
Volume entered for Assay EBNA IgG = 143

Enter the daily billable test volume for assay Toxo IgG :
> 86
Volume entered for Assay Toxo IgG = 86

Enter the daily billable test volume for assay Toxo IgM :
> 54
Volume entered for Assay Toxo IgM = 54
```

After volumes for all assays are entered, the program automatically starts calculating a feasible optimal solution and starts outputting the results on the screen. The outputs of the program will be explained in the next section.

2.3. Interpreting Outputs

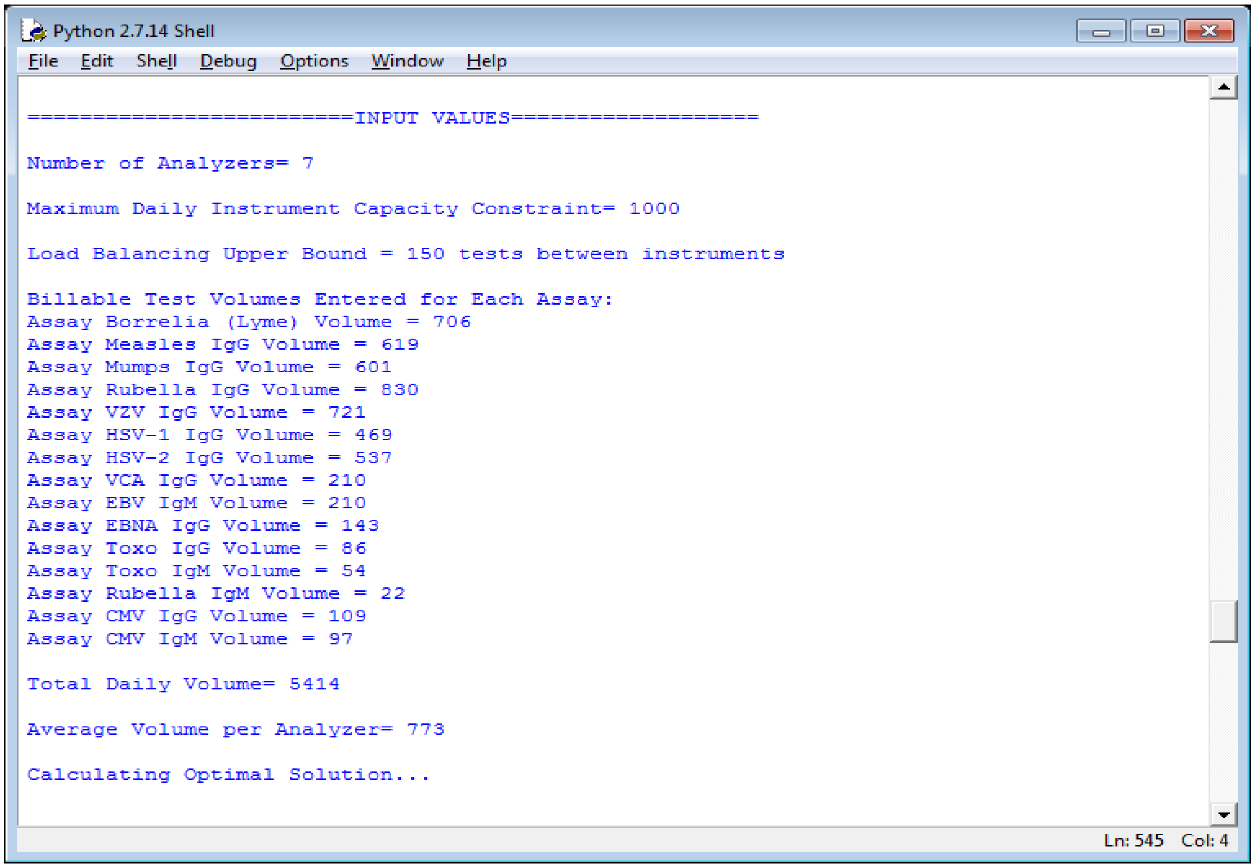
Step-1: Interpreting Input Values Summary Section

This section outputs a summary of the user entered inputs on the screen:

- Number of Analyzers
- Maximum Daily Instrument Capacity Constraint
- Load Balancing Upper Bound
- Billable Test Volume Entered for Each Assay
- The program outputs only the assays for which non-zero volume is entered.

In addition to the user entered inputs, the program calculates and outputs the following values based on the input values:

- Total Daily Volume: The program calculates the total daily volume of billable tests by summing up the volumes entered for each assay.
- Average Daily Volume per Analyzer: The program calculates this average value by dividing the Total Daily Volume by the Number of Analyzers on the site.



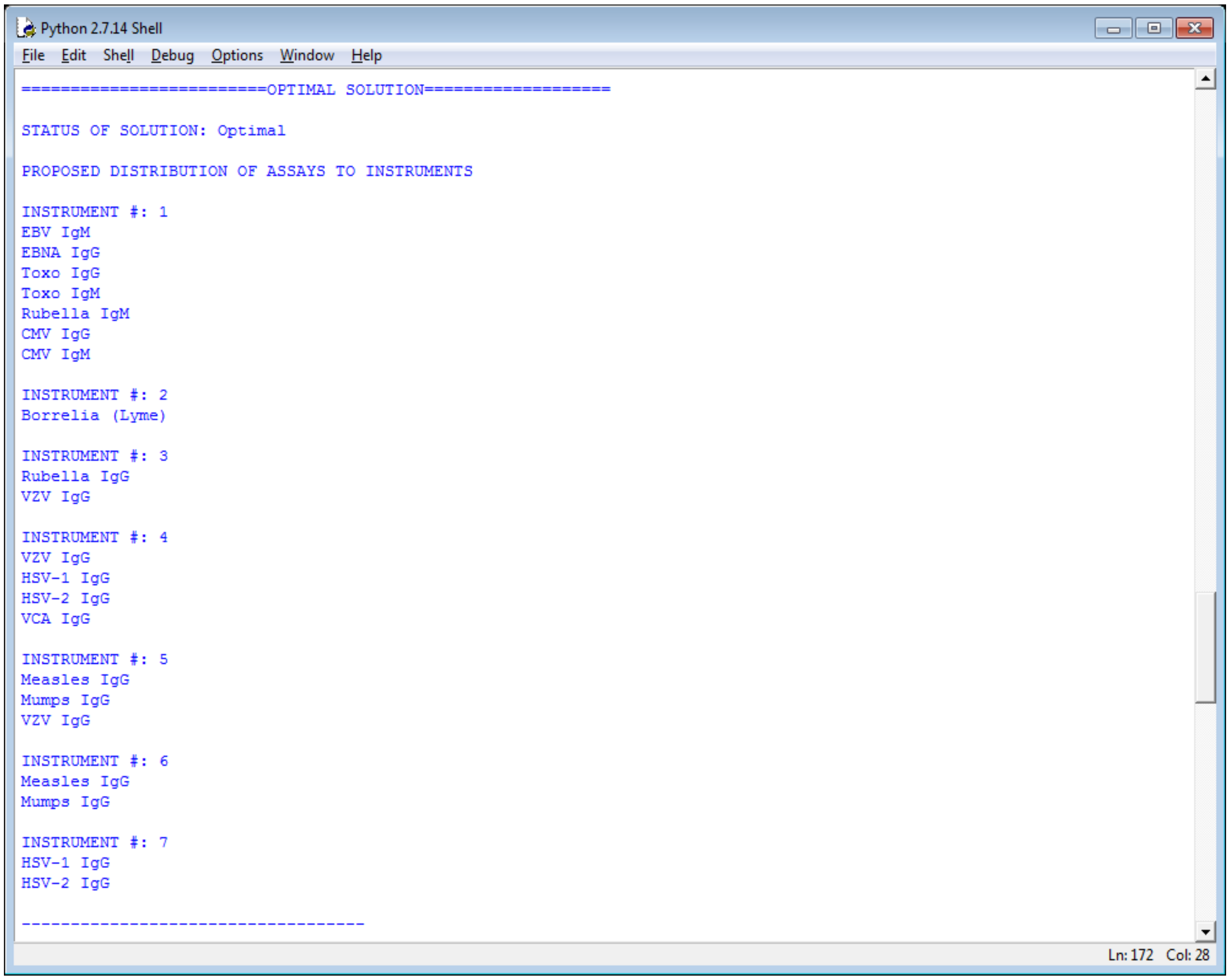
```
=========================INPUT VALUES===================

Number of Analyzers= 7

Maximum Daily Instrument Capacity Constraint= 1000

Load Balancing Upper Bound = 150 tests between instruments

Billable Test Volumes Entered for Each Assay:
Assay Borrelia (Lyme) Volume = 706
Assay Measles IgG Volume = 619
Assay Mumps IgG Volume = 601
Assay Rubella IgG Volume = 830
Assay VZV IgG Volume = 721
Assay HSV-1 IgG Volume = 469
Assay HSV-2 IgG Volume = 537
Assay VCA IgG Volume = 210
Assay EBV IgM Volume = 210
Assay EBNA IgG Volume = 143
Assay Toxo IgG Volume = 86
Assay Toxo IgM Volume = 54
Assay Rubella IgM Volume = 22
Assay CMV IgG Volume = 109
Assay CMV IgM Volume = 97

Total Daily Volume= 5414

Average Volume per Analyzer= 773

Calculating Optimal Solution...
```

Figure: Input Values Summary Sections

Step-2: Interpreting Optimal Solution Section

The program proposes an optimal distribution of assays to instruments following the user given constraints and merge rules. An example screenshot is included below.

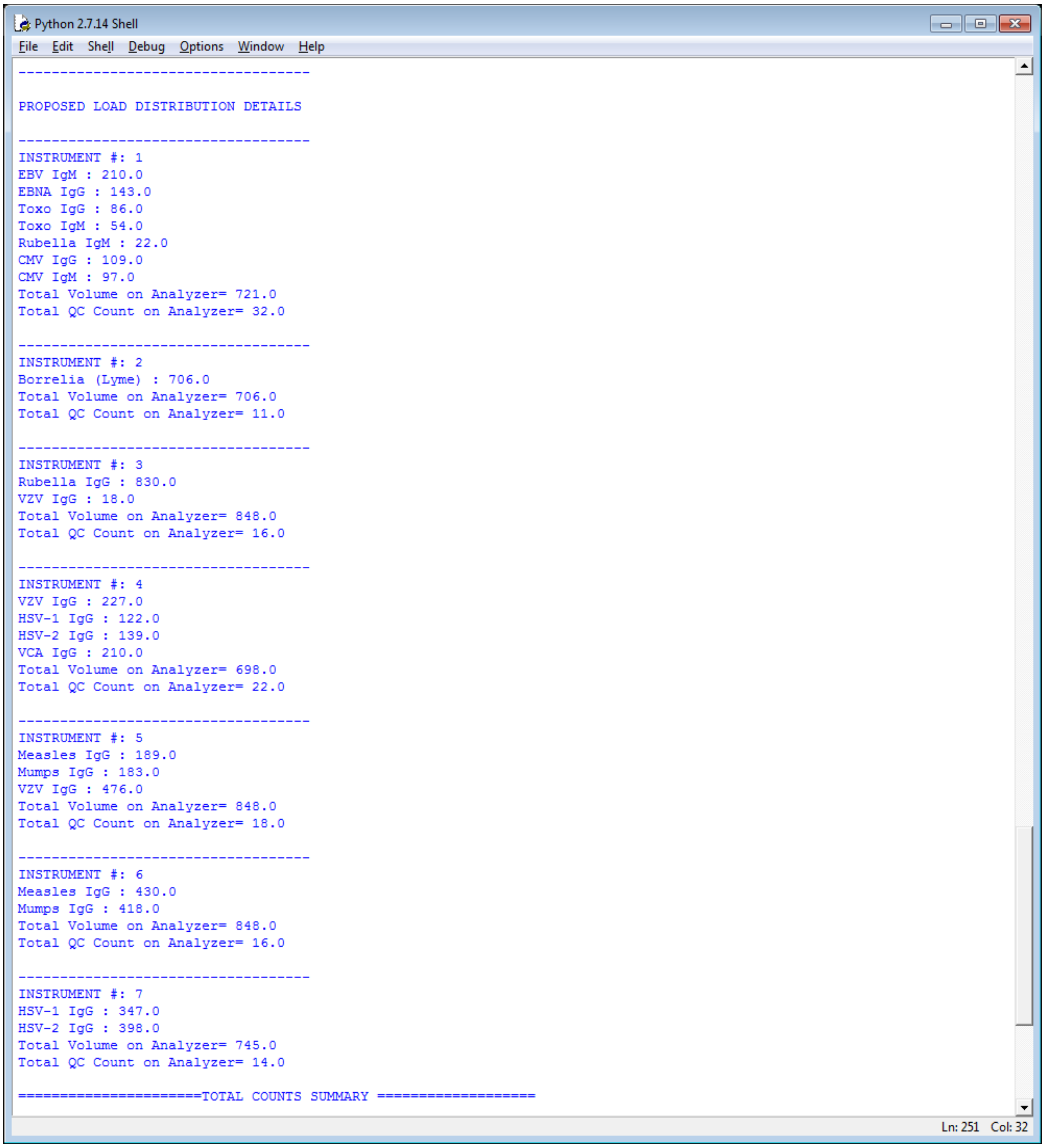
- **Status of Solution Field:** The output of this field can either be "Optimal" or "Infeasible".
  - **"Optimal"** means the program is able to find an optimal solution that satisfies the user given constraints.
  - **"Infeasible"** means the program is not able to find a solution that satisfies the user given constraints. This happens either when the number of total daily volume exceeds the total instruments capacity set by the user or when the load balancing constraint is set too narrow. The user should increase the daily maximum test capacity constraint and rerun the program to find a feasible (optimal) solution, or alternatively the user should increase the load balancing constraint value and rerun the program to find a feasible (optimal) solution.



```
=========================OPTIMAL SOLUTION===================

STATUS OF SOLUTION: Optimal

PROPOSED DISTRIBUTION OF ASSAYS TO INSTRUMENTS

INSTRUMENT #: 1
EBV IgM
EBNA IgG
Toxo IgG
Toxo IgM
Rubella IgM
CMV IgG
CMV IgM

INSTRUMENT #: 2
Borrelia (Lyme)

INSTRUMENT #: 3
Rubella IgG
VZV IgG

INSTRUMENT #: 4
VZV IgG
HSV-1 IgG
HSV-2 IgG
VCA IgG

INSTRUMENT #: 5
Measles IgG
Mumps IgG
VZV IgG

INSTRUMENT #: 6
Measles IgG
Mumps IgG

INSTRUMENT #: 7
HSV-1 IgG
HSV-2 IgG

-----------------------------------
```

Step-3: Interpreting Proposed Load Distribution Details Section

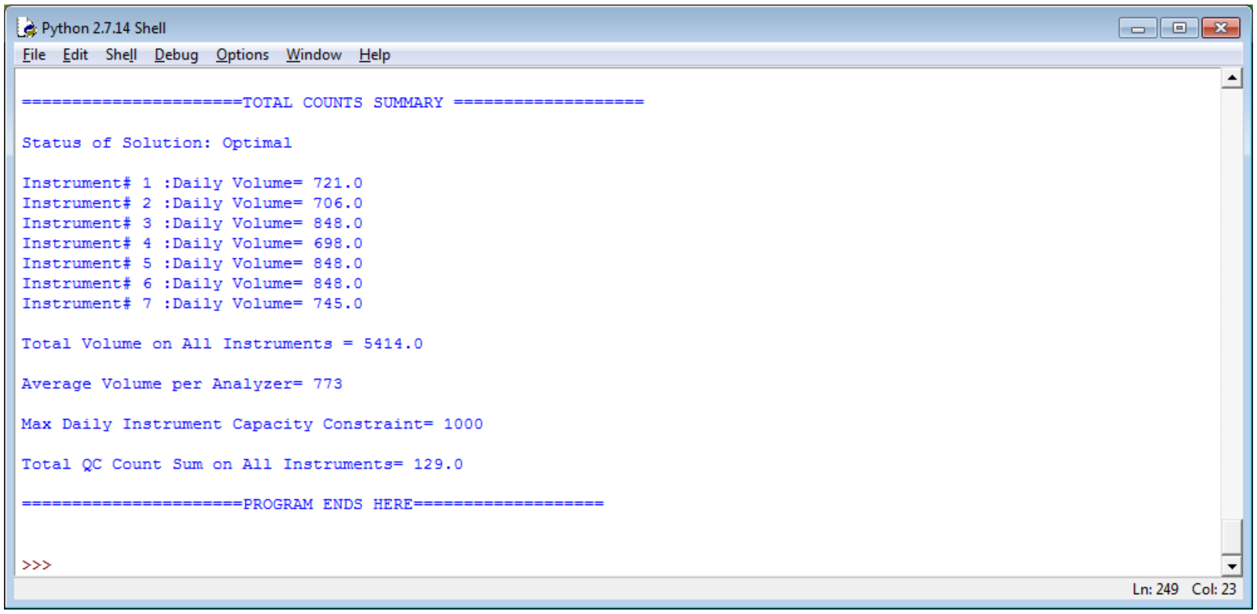
The following output section displays a detailed load distribution of test volumes to instruments with resulting total billable test volumes and the quality control (QC) test counts on each instrument.



```
-----------------------------------

PROPOSED LOAD DISTRIBUTION DETAILS

-----------------------------------
INSTRUMENT #: 1
EBV IgM : 210.0
EBNA IgG : 143.0
Toxo IgG : 86.0
Toxo IgM : 54.0
Rubella IgM : 22.0
CMV IgG : 109.0
CMV IgM : 97.0
Total Volume on Analyzer= 721.0
Total QC Count on Analyzer= 32.0

-----------------------------------
INSTRUMENT #: 2
Borrelia (Lyme) : 706.0
Total Volume on Analyzer= 706.0
Total QC Count on Analyzer= 11.0

-----------------------------------
INSTRUMENT #: 3
Rubella IgG : 830.0
VZV IgG : 18.0
Total Volume on Analyzer= 848.0
Total QC Count on Analyzer= 16.0

-----------------------------------
INSTRUMENT #: 4
VZV IgG : 227.0
HSV-1 IgG : 122.0
HSV-2 IgG : 139.0
VCA IgG : 210.0
Total Volume on Analyzer= 698.0
Total QC Count on Analyzer= 22.0

-----------------------------------
INSTRUMENT #: 5
Measles IgG : 189.0
Mumps IgG : 183.0
VZV IgG : 476.0
Total Volume on Analyzer= 848.0
Total QC Count on Analyzer= 18.0

-----------------------------------
INSTRUMENT #: 6
Measles IgG : 430.0
Mumps IgG : 418.0
Total Volume on Analyzer= 848.0
Total QC Count on Analyzer= 16.0

-----------------------------------
INSTRUMENT #: 7
HSV-1 IgG : 347.0
HSV-2 IgG : 398.0
Total Volume on Analyzer= 745.0
Total QC Count on Analyzer= 14.0

=====================TOTAL COUNTS SUMMARY ==================
```

Step-4: Interpreting Proposed Load Distribution Details Section

The following output section display a summary of total daily test volumes on each instrument and the resulting total QC tests count on all instruments. The user can try to reduce the total QC count by relaxing (increasing) the load balancing constraint and maximum instrument capacity constraint.



```
======================TOTAL COUNTS SUMMARY ===================

Status of Solution: Optimal

Instrument# 1 :Daily Volume= 721.0
Instrument# 2 :Daily Volume= 706.0
Instrument# 3 :Daily Volume= 848.0
Instrument# 4 :Daily Volume= 698.0
Instrument# 5 :Daily Volume= 848.0
Instrument# 6 :Daily Volume= 848.0
Instrument# 7 :Daily Volume= 745.0

Total Volume on All Instruments = 5414.0

Average Volume per Analyzer= 773

Max Daily Instrument Capacity Constraint= 1000

Total QC Count Sum on All Instruments= 129.0

======================PROGRAM ENDS HERE===================


>>>
```

3. REFERENCES

1. DiaSorin LIAISON® XL User Manual - Revision C

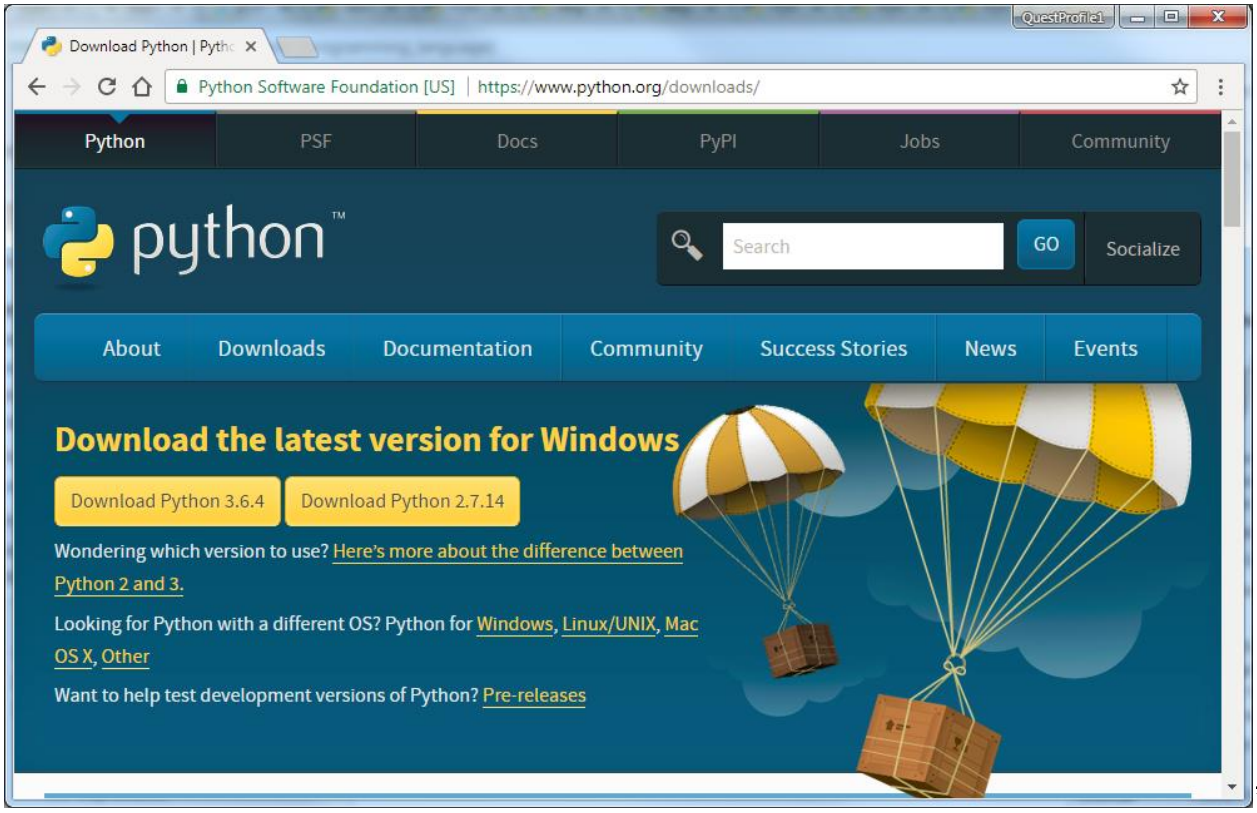
2. DiaSorin LIAISON® Infectious Disease Assay Best Practices Document

4. APPENDIX

4.1 PYTHON INSTALLATION AND ENVIRONMENT SETUP

This tool is written using Python (version 2.7) software language. Python coding language is developed under an OSI-approved open source license, making it freely usable and distributable for commercial use. The Python software tool is free to download from the following website of the Python Software Foundation:
https://www.python.org/downloads/

Step-1: Click on "Download Python 2.7.14" version to start downloading the Windows installation file: "python-2.7.14.msi" to any folder of your choice in your Windows computer.



Download the latest version for Windows

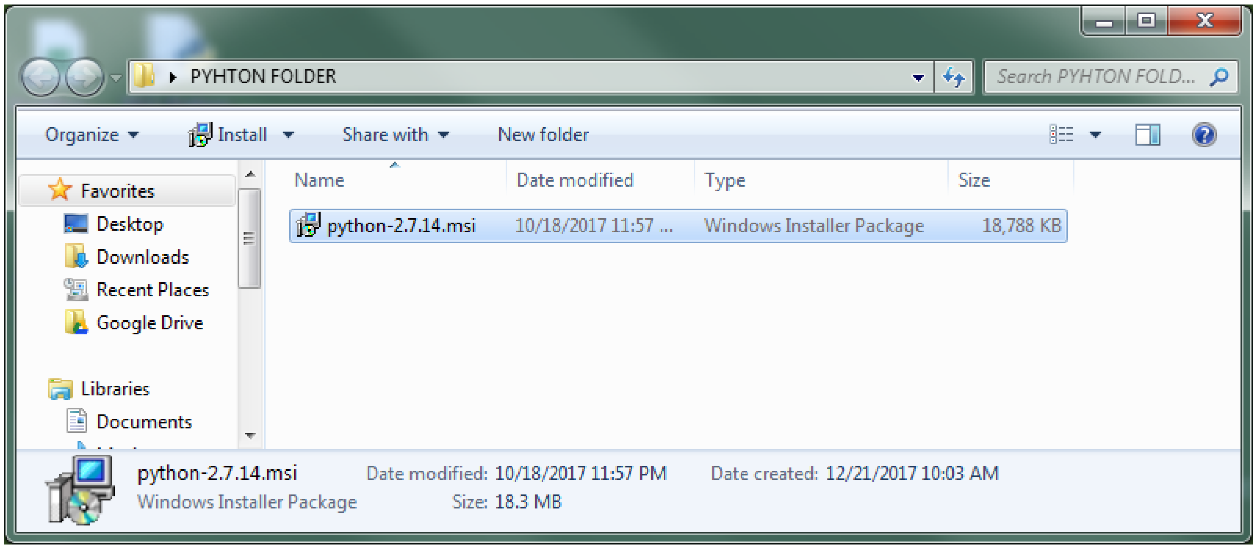
Wondering which version to use? Here's more about the difference between Python 2 and 3.

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Pre-releases

,

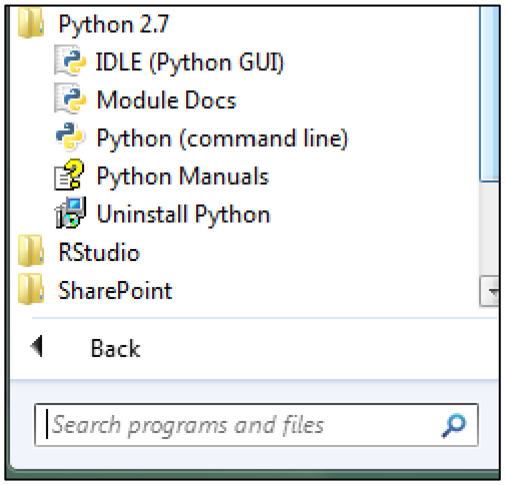
Step-2: Click on the "python-2.7.14.msi" file to start installation of the Python and follow the installation instructions on the screen. The installation should take less than 3 minutes.



Name	Date modified	Type	Size
python-2.7.14.msi	10/18/2017 11:57 ...	Windows Installer Package	18,788 KB

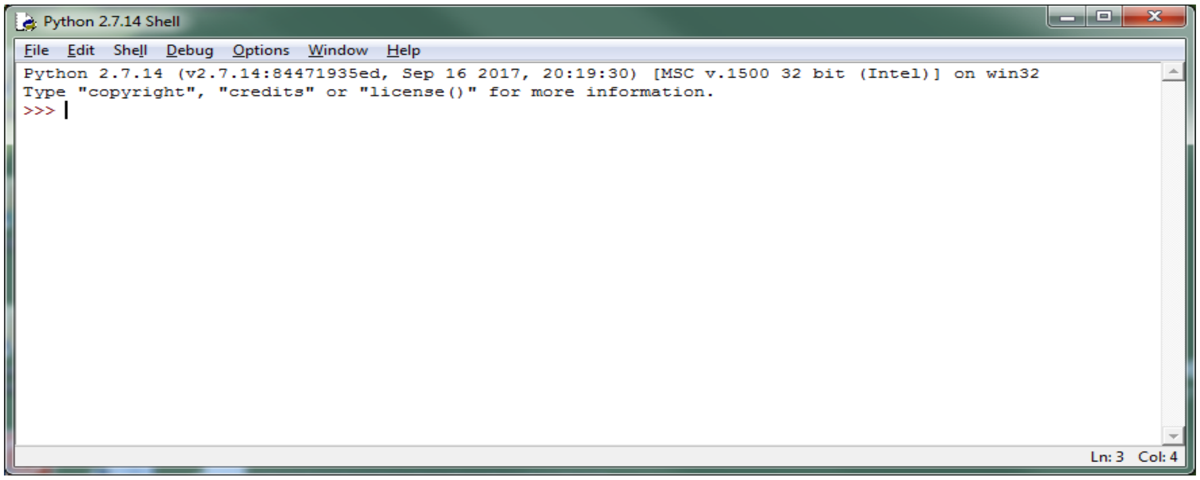
python-2.7.14.msi Date modified: 10/18/2017 11:57 PM Date created: 12/21/2017 10:03 AM
Windows Installer Package Size: 18.3 MB

Step-3: After the successful installation, you should be able to see Python 2.7 folder and icons under the Start Menu > Programs location.



This verifies that the Python program is successfully installed and ready to use at this point.

Step-4: Click on "IDLE (Python GUI) " icon to launch the graphical user interface as shown in the screenshot below.



```
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:19:30) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
```

4.2 INSTALLING PYTHON LIBRARIES

The Load Distributor Tool Python script uses the following open-source public Python packages (libraries) that do not come pre-installed within Python 2.7 distribution.

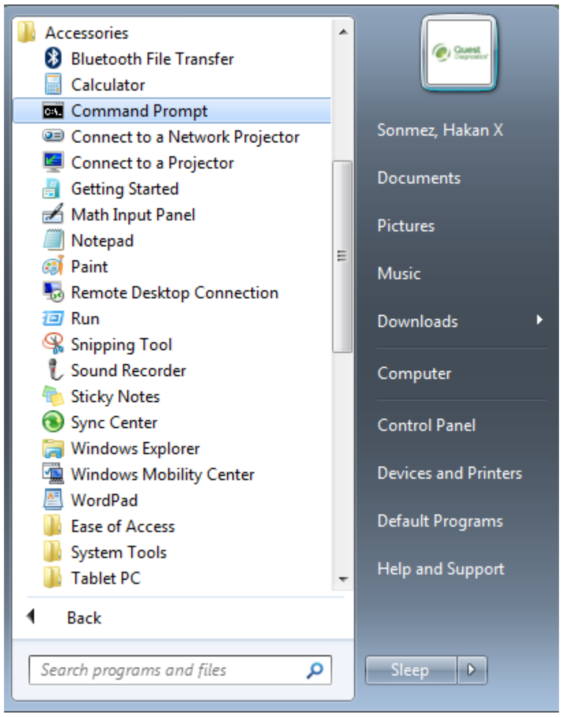
1. PuLP : A linear programming (LP) modeler package for Python.
More info at: http://pythonhosted.org/PuLP/

2. NumPy: A scientific computing package for Python.
More info at: http://www.numpy.org

The PuLP and Numpy libraries should be installed before running the script for the first-time. This is a one-time procedure. Note that other Python libraries used in the script such as "math" library and "time" library come pre-installed within Python 2.7 distribution and they do not require additional installation.

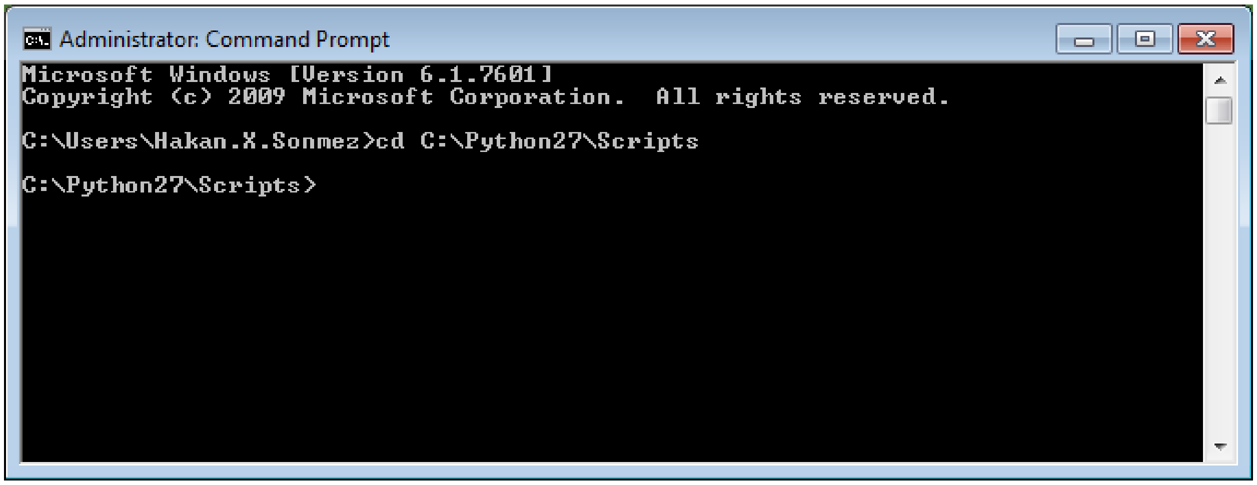
Step-1: Launch Windows Command Prompt (cmd.exe)

Click on the "Command Prompt" icon under the Accessories folder on the Windows Start Menu as shown in the screenshot below.



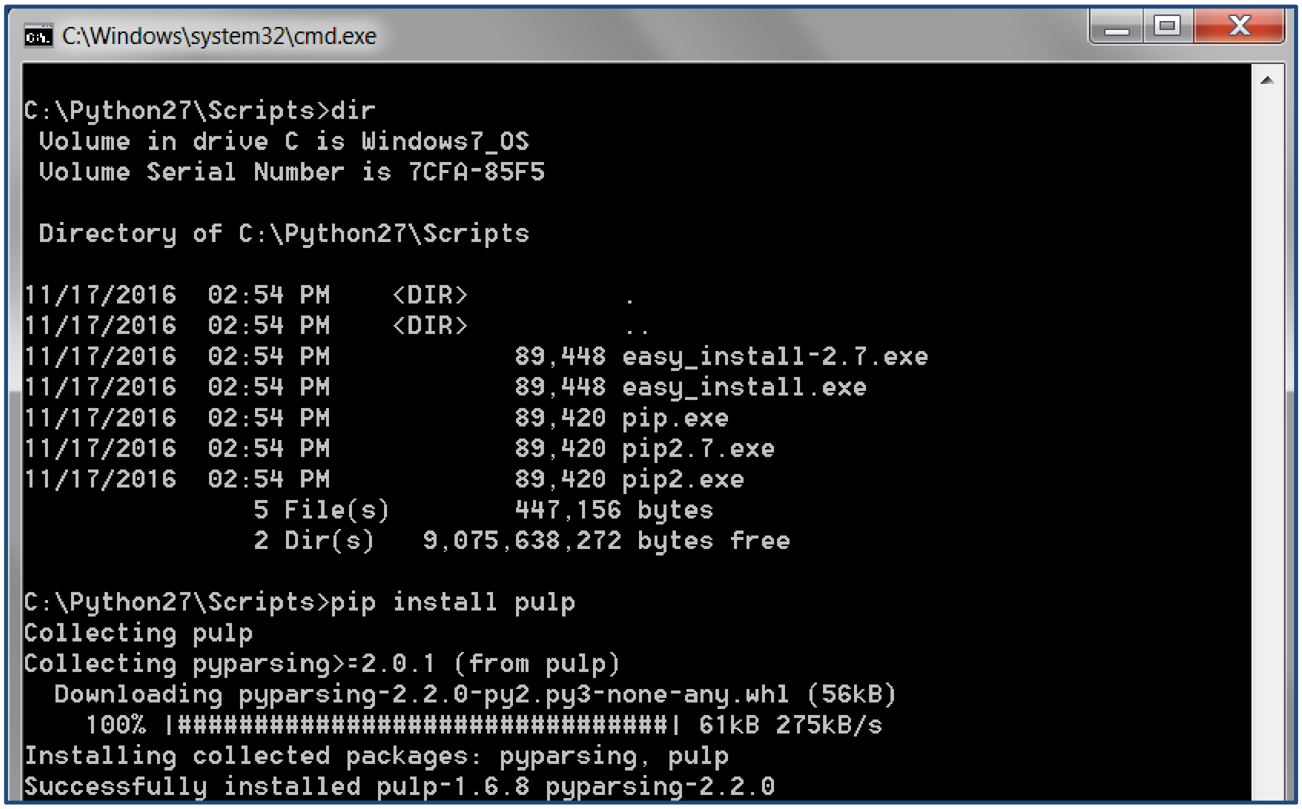
Step-2: Go to "C:\Python27\Scripts" directory on the command prompt.

Type "`cd C:Python27\Scripts`" on the command prompt and press enter to go to C:\Python27\Scripts directory.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Hakan.X.Sonmez>cd C:\Python27\Scripts

C:\Python27\Scripts>
```

Step-3: Run “`pip install pulp`” command on the command prompt to install PuLP package as displayed in the screenshot below.

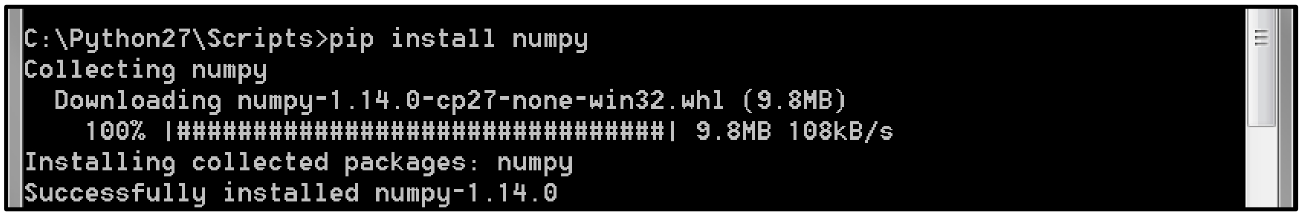


C:\Windows\system32\cmd.exe

```
C:\Python27\Scripts>dir
 Volume in drive C is Windows7_OS
 Volume Serial Number is 7CFA-85F5

 Directory of C:\Python27\Scripts

11/17/2016  02:54 PM    <DIR>          .
11/17/2016  02:54 PM    <DIR>          ..
11/17/2016  02:54 PM            89,448 easy_install-2.7.exe
11/17/2016  02:54 PM            89,448 easy_install.exe
11/17/2016  02:54 PM            89,420 pip.exe
11/17/2016  02:54 PM            89,420 pip2.7.exe
11/17/2016  02:54 PM            89,420 pip2.exe
               5 File(s)        447,156 bytes
               2 Dir(s)   9,075,638,272 bytes free

C:\Python27\Scripts>pip install pulp
Collecting pulp
Collecting pyparsing>=2.0.1 (from pulp)
  Downloading pyparsing-2.2.0-py2.py3-none-any.whl (56kB)
    100% |################################| 61kB 275kB/s
Installing collected packages: pyparsing, pulp
Successfully installed pulp-1.6.8 pyparsing-2.2.0
```

After successful installation, you should see the “successfully installed pulp-1.6.8” line on the command prompt.

Step-4: Run “pip install numpy” command on the command prompt to install PuLP package as displayed in the screenshot below.



```
C:\Python27\Scripts>pip install numpy
Collecting numpy
  Downloading numpy-1.14.0-cp27-none-win32.whl (9.8MB)
    100% |################################| 9.8MB 108kB/s
Installing collected packages: numpy
Successfully installed numpy-1.14.0
```

After successful installation, you should see the “successfully installed numpy-1.14.0” line on the command prompt.

After this one time procedure, you should be ready to run the Load Distributor Tool script on the Python GUI.