Lab - Basic Problems in Python

# Objectives

1. Part 1: Prepare a Computer with Python 3 Installed

Part 2: Solve the Problems

# Background / Scenario

The goal of this lab is to solve some Python problems to become familiar with the programming language, its structure, and how to run code developed in a local environment.

# Required Resources

* Host computer with at least 4 GB of RAM and 15 GB of free disk space
* Internet access to download Visual Studio Code.

# Instructions

## Prepare a Computer with Python 3 Installed

In this Part, you will download and install desktop VS Code.

### Download and install Visual Studio Code (VS Code).

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages and runtimes (such as C++, C#, Java, Python, PHP, Go, .NET).

* + - 1. Navigate to <https://code.visualstudio.com/>. Click the download link on this page.
      2. Choose and download the appropriate VS Code installation file based on your operating system.
      3. Run the installer for **VS Code** and accept the default installation settings.
      4. VS Code is open and ready for the next step.

### Install Python 3

* + - 1. Navigate to the Extension tab in the VS Code User Interface (UI).
      2. Search for “Python” in the Search Extension box.
      3. Click on “Install” and wait until the installation finished.
      4. Restart VS Code.

## Solve the Problems

In this part, the intention is to write a single .py file with all the functions that are being implemented. Please note the name of the feature for each item as the feature review process is automated and requires the name to be as specified.

Please use the next structure as name of your file before uploading: <NAME>-lab-basic-python.py (e.g. juan-caviedes-lab-basic-python.py)

|  |  |  |  |
| --- | --- | --- | --- |
| **Python Basics** | | | |
| **Id.** | **Function Name** | **Description** | **Value** |
| 1 | fibonacci\_series | Write a function that returns N numbers from the Fibonacci series. The number N is an input determined by the user using the script. | 2 |
| 2 | factorial\_number | Write a function that allows you to calculate the factorial of a number. | 2 |
| 3 | time\_from\_12 | Write a function that gets the time as three integer arguments (for hours, minutes, and seconds) and returns the number of seconds since the last time the clock “reached 12.” Use this function to calculate the amount of time in seconds between two hours, when both are within a 12-hour cycle. | 4 |
| 4 | smallest\_number | Write a function that returns the smallest of three numbers. | 1 |
| 5 | is\_prime\_number | Write a function that determines if a number is prime. | 3 |
| 6 | prime\_numbers\_to\_1000 | Write a function that displays all prime numbers between 1 and 1000. | 3 |
| 7 | reversed\_integer | Write a function that takes an integer value and returns the number with its digits reversed. For example, given the number 7631, the function should return 1367. | 3 |
| 8 | gcf\_lcm\_numbers | Write a function that returns the greatest common factor or least common multiple of two numbers. The user can choose whether he wants the greatest common divisor, the least common multiple, or both. | 4 |
| 9 | separate\_words | Write a program that receives a sentence and separates the words by spaces. For example, if I receive the text "I love programming in Python", it should return "I" "love" "programming" "in" "Python" in a list. | 4 |
| 10 | is\_palindrome\_word | Write a program that determines if a word is a palindrome. | 5 |
| 11 | number\_of\_each\_letter | Write a function that reads a text file and determines the number of each letter of the alphabet in that text. For example, “I love Python programming” has one “y”, two “m”, two “p”, etc. The function should return the count of only the letters that exist in the function. | 5 |
| 12 | number\_of\_letter\_word | Write a function that indicates the number of one-letter words, two-letter words, three-letter words, etc. For example, “I love Python programming” has zero one-letter words, two two-letter words, one three-letter word, zero four- and five-letter words, one six-letter word, and one word with more than six letters. | 5 |
| 13 | sum\_nautral\_numbers | Write a function that returns the sum of the first N natural numbers. | 1 |

End of document