Laboratory Assignment 1
Programming Exercises in Python

CSC372-M72: Optimisation

2023-24

1 Objectives.

- To solve simple programming problems using Python.
- To use Jupyter Notebook in writing Python programs.

2 Programming Exercises

Write functions (with appropriate input and output parameters) in a Jupyter Notebook using Python to perform the tasks in the following section.

2.1 Tasks.

1. Write a function that takes an integer and returns True if it is even. A way to figure out if an integer is even or odd is to divide the number by 2 and compute the remainder. If there is a remainder then it is an odd number, otherwise it is even. Use the following numbers to test your code: 5 and 122.

2. In a module, there is a coursework and a quiz. The contribution from each element is configurable as a percentage, e.g. $x\%$ contribution from coursework and $y\%$ contribution from the quiz, where $x + y = 100$. With x and y, we create an array $\mathbf{p} = (x, y)$. Consider that someone achieves c marks for the coursework, and q marks for the quiz; all these marks are out of 100. So, the marks array is: $\mathbf{m} = (c, q)$. Write a function that takes the arrays $\mathbf{p}$ and $\mathbf{m}$, and returns the overall percentage of marks achieved for a student.

3. Write a function that can return the maximum number in a list of integers using only *for* loop and comparison operators. Use the following numpy array to test your function: $(1, 19, 2, 3, 4, 100)$; your program should return 100.

4. Write a function that can produce an $m \times n$ dimensional random integer uniformly. Here m and n should be user defined. [Hint: you may find the *random.uniform* function useful]

5. The Fibbonacci numbers, commonly denoted as F_n for a sequence, called the Fibbonacci sequence, such that each number is the sum of the two preceding ones, starting from 0 and 1.

Formally, it can be defined as:

$$F_n = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F_{n-1} + F_{n-2} & \text{if } n > 1 \end{cases}$$

Write a function that computes the nth Fibonacci number F_n for a user-defined n.

3 Simple plotting with *matplotlib*

Python has a powerful plotting library called *matplotlib*. You can see some of the beautiful plots that can be created with the library in the following link.

`matplotlib.org/gallery.html`

To install the module, please issue the following command in your terminal.

```
> pip install matplotlib
```

For a simple plot, try the following code in your Jupyter notebook:

```
1 import matplotlib.pyplot as plt # import library
2 plt.ion() # this enables interactive plotting
3 x = [1, 2, 3, 5, 6] # horizontal axis value
4 y = [2, 4, 5, 6, 6] # vertical axis values
5 plt.plot(x, y) # plot a line graph
```

This should produce the following plot in Figure 1.

3.1 Task.

Using the solutions for exercises in 2.1 (5), generate a plot of the function response when n varies from 0 to 100.

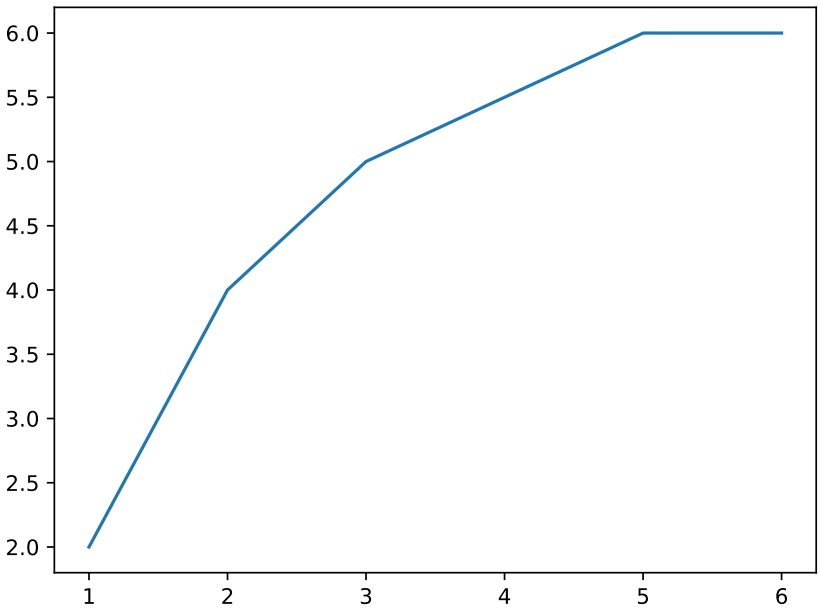

Figure 1: A simple plot