**RA Evaluation exam answer sheet**

1. **Unix Commands**

**A1.**

**/tmp/data/images/** - there could be more folder before tmp

**tmp/data/images/ -** there are no folder before tmp

**./tmp/data/images/** - tmp is the root directory

**A2.**

**~** means home directory in Linux. “echo ~” will print the current home directory in the terminal.

**A3.**

“mkdir {1..10}” this command will create 10 directories named 1 to 10 in the directory the terminal is in.

**A4.**

“mkdir data” only creates a data folder in the current directory of the terminal is in.

“mkdir -p data” makes sure that **mkdir**adds any missing parent directories in the process.

Between these two the **second** one should be used for best practice.

**A5.**

Write the following command in terminal.

* cd /image\_directory

go the image directory where the image file resides.

* eog 100000.png

This line will open the required image file

Or you can directly write the following command.

* eog /image\_directory/100000.png

**A6.**

find -type f -name "filename" -exec grep -l 'message\_text' {} +

1. **Python Basic**

**B1.**

The solution of Part is given at “**B1\_part\_1\_and\_part\_2.py**” file.

**B2.**

The solution of this problem is given at “**B2.py**”

**B3.**

The solution of both Part is given at “**B3\_part\_1.py”** file.

1. **Python OOP**

**C1.**

**\_variable\_name** single pre underscore is mainly used for internal use only. **Python** doesn't import the names which starts with a **single pre underscore.** Single pre underscore doesn't stop from accessing the single pre underscore variable. But, single pre underscore effects the names that are imported from the module.

**\_\_variable\_name** double pre underscore tells the Python interpreter to rewrite the attribute name of subclasses to avoid naming conflicts.

**variable\_name\_** post single underscore is used for naming variable. It doesn’t affect any other module.

**C2.**

The solution of this problem is given at “**C2.py**” file.

**C3.**

The given code will print **“f from class A”.**

**C4.**

The solution of this problem is given at **“C4.py”**

**C5.**

Adding this following line in the **FlowerImage** class’s \_\_init\_\_() function will call the \_\_init\_\_() function of the **Image** class:

**super().\_\_init\_\_(width, height)**

solution given at the C5\_superclass.py file.

**D. Data Structures and algorithms**

**D1.**

For the given voter id card problem, we could use a nested dictionary structure. The dictionary will look like the following:

voter\_list = {voter\_id\_card\_number\_1 : {'name': 'Abir', 'Address': 'Dhaka', 'Date\_of\_birth': '2/10/96'},

voter\_id\_card\_number\_2 : {'name': 'Emon', 'Address': 'Dinajpur', 'Date\_of\_birth': '2/10/97'}

}

The complexity of this dictionary will be O(1). Each voter number is a unique number thus we are using it as the key then we can use iterate through the dictionary through this. Complexity this structure is very fast and have the small look up time. Thus this is the best data structure to use in such scenerios.

**D2.**

The solution of this problem is written at “**D2\_recursion.py**” file.

**D3.**

Solution of this problem as written at “**D3\_fibonacci.py**” file.

**D4.**

Each time we call the “translator.predict” function we need to use the saved model for result. Rather we could store the results in a **dictionary** where based on the input output will be stored. Rather doing inference first we will check dictionary if the result is already there or not. If result is there, we don’t have to use predict function.

1. **Numpy**

**E1.**

The solution of this problem is given at **“E1.py”** file.

**F. Deep Learning**

**F1.**

The solution of this problem is given at **“F1.py”** file.

**F2.**

The solution of this problem is given at **“F2.py”** file.

**F3.**

The solution of this problem is given at **“F3.py”** file.