BRIEF OF WHAT TO DO   
  
  
Taks 2 is to write a business automation program to simulate business performance monitoring. It is necessary to monitor business performance to identify the best performing periods so management can zoom in to analyze the success factors. Management can discuss and decide if the success factors can be repeated and improved for greater success. At the same time, worst performing periods can also be analyzed to check if there are any failure factors that can be avoided in the future. This is a very tedious process that requires a lot of time and continuous effort. Automation will improve the efficiency of performance monitoring.

Python can be used to automate many office tasks. In this task, your team is required to develop a business automation program to automate the monitoring process:

Extract and summarise data from the finance dashboard in the final round of business stimulation game in MAB module. Refer to MAB module and MonsoonSim game on how to download the files. Your team is required to extract data from **day 11 to day 90**. You need to download the data a few times and combine the data together.

The automation will perform the tasks from the following csv files:

* Profit & Loss csv : The program will firstly compute the difference in the net profit column. If the net profit is always increasing, find out the day and amount the highest increment occurs. If the net profit is always decreasing, find out the day and amount the highest decrement occurs. If net profit fluctuates, list down all the days and amount when deficit occurs, and find out the **top 3** highest deficit amount and the days it happened.
* Cash-On-Hand csv: The program will firstly compute the difference in Cash-on-Hand. If the cash-on-hand is always increasing, find out the day and amount the highest increment occurs. If the cash-on-hand is always decreasing, find out the day and amount the highest decrement occurs. If cash-on-hand fluctuates, list down all the days and amount when deficit occurs, and find out the **top 3** highest deficit amount and the days it happened.
* Overheads csv: The program will find the highest overhead category.
* Write the computed amount from a to d will to a text file and name it as summary\_report.txt.

Figure 1.0 included three scenarios to illustrate the automation objectives and the expected output in summary\_report.txt.

Your team will be required to analyze the best / worst performing day for success / failure factors as per the requirements in other modules, such as SAPB. For profit and loss, if it is always increasing / decreasing, your team is required to analyze the factors of highest surplus / deficit; when profit and loss fluctuates, your team is required to analyze the top 3 deficit amount and present the most meaningful analysis in the presentation. For details, please look through the group project requirements from other modules.

The same process applies to the cash-on-hand.

# PROJECT TASK DESCRIPTIONS AND REQUIREMENTS

**Files and Project Directory**

You should organise your program and csv files into the following folder structure. **(Your team is required to follow the structure and file names exactly. Otherwise, your team will be penalized.)**

A screenshot of a computer

Description automatically generated

Dedicate each python file to achieve specific tasks. For example, the cash\_on\_hand.py should only contain codes that compute the difference in Cash-on-Hand, while overheads.py should only contain codes that find the highest overhead category.

Organizing code this way makes the overall program more manageable, easier to maintain and debug errors.

**Coding Skills**

To complete the assignment successfully, you need to use only the programming topics learn from PFB, unless given the permission to do so.

The use of external modules not taught will **severely affect the grade**. External module refers to additional module installed with pip install command.

However, you may use any built-in functions or/and modules.

**Standard Criteria**

The project will be evaluated based on:

1. Program Correctness

2. Code Readability

3. Code Elegance/ Efficiency

4. Code Documentation

5. Assignment Specification

. Modularized the python files  
**What is a modular program?**

* Modularization is the technique of splitting a large programming task into smaller, separate, and manageable subtasks.
* To achieve modularization, you can further organized the code in each python file as a function.
* A main python file (main.py) will import these functions, to coordinate and execute the functions.
* In this way the overall program becomes even more manageable, easier to maintain and debug errors.
* Refer to Figure 2.0 for an example of modularizing a complex program.

**Figure 1.0 Automation Objectives**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

You will have to make the program modular!!