MONASH University

FIT9136 Assignment 1

Last Update: 06/04/2020

Version: 1.6

Table of Contents

1. Introduction

This assignment is due on **Week 6, 1st May, 2020 at 5pm**. It is worth 20% of the total unit marks. A **penalty of 20%** per day will apply for late submission including weekends and public holidays. Refer to the FIT9136 Unit Guide for the policy on extensions or special considerations.

Note that this is an individual assignment and must be your own work. Please pay attention to Section 4.2 of this document on the university policies for the Academic Integrity, Plagiarism and Collusion.

All the program file and any supporting documents should be compressed into one single file for submission.

(The submission details are given in Section 4.)

Note: for the sake of this assignment you do not need to import any libraries. However, if you want to, you may import math.

2. Simulating an Inventory System

2.1 Overview

Your assignment is to simulate the stocking level of one product: Cantilever Umbrella, inside an inventory management system in an Australian firm. The firm provides this product to its distributors together with a recommended retail price (RRP).

*Tips: the most basic information inside inventory management system contains at least the recommended retail price (RRP) and quantity of the product. (There are a lot of other fields too, but we have ignored those fields due to complexity)

Here is some information regarding this product and its relevant parties:

1. The Australian firm was established in January 1st, 2000. For the past 20 years, there have been no changes to the inventory management system. There was also no change in product model in the past 20 years as it is quite robust design, the product model will not change in the future either.
   On January 1st, 2000, when the firm was first open for business, there were 1000 cantilever umbrellas in stock, the distribution number of the cantilever umbrella on that day to the distributors was 36, and each cantilever umbrella's RRP was $705 AUD (This date has taken the peak season, which causes increase in sales quantity and increase in price, into account, details of peak season is mentioned in part 2).
   Please note when the inventory stock drops to 400, the firm will restock 600 cantilever umbrellas back to the warehouse (We do not consider any cost related to restocking fee).

2. Cantilever Umbrella has a peak selling season. It is from 1st November to end of February each year (Number of days in February is decided by whether that year is a leap year or not).
   During the peak season, the company is expected to have a 35% increase in quantity for distribution (Which means the number of cantilever umbrellas that goes out of the inventory system is increased by 35%, rounded up to an integer).
   It is also expected to have 20% increase in RRP (Recommended Retail Price, contains 2 decimal places, it will be the same for the rest of the document) during peak season as it is hard to supply enough umbrellas to meet the demand.

3. The stocking system is updated daily at 11:59 pm. This number has been consistent every day until the beginning of a new financial year.
   At the beginning of the new financial year each year (1st July), the company will impose a 10% increase in the supply of cantilever umbrellas to its distributors (rounded up) and 5% increase of the RRP due to inflation.

4. Based on statistics, global financial crisis happens every 9 years, and lasts for another 2 years, the number of cantilever umbrella distributed to distributors will drop by 20% in the first year when global financial crisis hit the market, the number will continue to drop by 10% and 5% for the next 2 years when the economy is recovering.
   In order to make up the losses, during the year that a global financial crisis starts, the company will add an additional 10% increase in RRP to the product, the increase of the product RRP will become 5% in the next year, and 3% the year after to make up the loss.

During the crisis, the price inflation and increase in quantity for distribution are still valid and applicable.

Note: the crisis will start on 1st Jan on the 9th year, and end on 31st Dec the 11th year. In this example, it will start on 1st January, 2009 and ends on 31st Dec, 2011. And there will be another crisis start on 1st January, 2020 etc.

Example of price increase to make up the loss during crisis is: the price increase will start on 1st January, 2009, and it will have another increase on 1st January, 2010, etc.

5. It is expected that 5% of items will be defective and returned to warehouse every month~~. (Which means the inventory number will increase by 5%).~~

   Defective items will be refurbished and redistributed at 80% of original price (original price is the RRP at the time the product is returned) in the following months.

This firm assesses the quantity of product distributed and total revenue earned from distributors every year (The most basic formula for revenue is: Revenue = RRP x total quantity. But please bear in mind you also need to consider the defective items as well as global financial crisis and inflation and increase over the supply of cantilever umbrella).

The firm also runs predictions 20 years in advance. (We are currently in year 2020, the company will have a prediction of this information to year 2040)

For your program, the starting information will need to be read from a file called “AU_INV_START.txt”.

After operation of your program, the information will be saved to a file called “AU_INV_END.txt”.

Be sure to read the specification very carefully. Your code will be partially marked based on its output which must match those given in this specification.

2.2 The assignment

Warning: You cannot hard code data, penalty applies when hard coding is used inside any part of the program.

Example of hard coding is: if the requirement ask you perform simulation for one year after year 2020 which should output 2021, you directly write out end year as 2021 instead of using the structure such as variable = 2020, then variable + 1 as output (Pseudo code). In such cases, if the starting year is 2050, the hard-coded program is not going to work.

2.2.1 Task 1: Reading and writing data and calculating total revenue and total stock

Detailed instructions are as follows:
Create a python script called {YourStudentID}_task1.py.
Note: you should not include {} in your file name

Create a function that reads in data, the function name for reading data is called `read_data()`
Note: This data you are reading should be stored in one single variable which is a dictionary data type, variable name is up to you.
Make sure the keys are exactly as follows:

```
{
  "start_year": XXX,
  "start_stock": XXX,
  "start_revenue": XXX
}
```

The file that needs to be read is called “AU_INV_START.txt” and structure will be the following format:

1. Starting year
2. Total stock available.
3. Total revenue it made for that year

Create the second function in your python file that calculate total stock remaining and total revenue of a single year's cycle.

Note: The function name is `cal_stock_revenue(first_variable)`. (Instead of first_variable, you should use a descriptive name for the parameter). Example for single year cycle is, 1st Jan, 2000 to 31st Dec 200~~1~~0, the year can be either normal year or leap year, starting day for the year is always 1st Jan and ending year is always 31st Dec.

This function should take in one variable which you created based on the reading data part and output one single dictionary variable that contains the end year in 4 digits, total stock available ~~(2 decimal places if have)~~ at the end of the year and total revenue (2 decimal places if have) it made at the end of the year.

The structure of dictionary is as follows:

```
{
  "end_year": XXX,
  "end_stock": XXX,
  "end_revenue": XXX
}
```

Lastly, your python file should have a function that can create a new file called “AU_INV_END.txt”, writing the data you calculated.

Note: The function name is called `write_data(second_variable),` the variable is the dictionary variable that has been output from the `cal_stock_revenue` function.

The file that you are writing will have the following format (The txt file will have 3 lines inside):

1. Ending year ~~and month and day~~
2. Total stock available
3. Total revenue it made for end year

Examples of format of the two files are showing below:

~~Total stock and~~ total revenue can have at least 2 decimal places if there are any decimals

	AU_INV_START.txt	AU_INV_END.txt
Example 1	2000 Y *(Y is a number for total stock)* X *(X is a number for total revenue)*	2001 YY *(YY is a number for total stock)* XX *(XX is a number for total revenue)*
Example 2	2010 Y *(Y is a number for total stock)* X *(X is a number for total revenue)*	2011 YY *(YY is a number for total stock)* XX *(XX is a number for total revenue)*

2.2.2 Task 2: Performing the Simulation

Using the techniques demonstrated in the previous tasks, create the full simulation of the town in a script called {YourStudentID}_task2.py.
Note: you should not include {} inside your file name. The file structure and dictionary structure are exactly the same with your Task 1.

You do not need to import the files and functions from Task 1 to Task 2. So, it is sufficient to copy and paste the function from Task 1 to this python file.

Your simulation should adhere to the following algorithm which was created using the overview in section 2.2.1:

1. Starting year, date, stock and revenue could be customized inside the “AU_INV_START.txt”
2. Number of years of simulation could be set inside the python file with a constant variable `NO_YEAR_SIM`, default value is 3.
3. Percentage of defective items could be adjusted with a constant variable `PER_DEF`. Default value is 5.
4. The global financial crisis reoccurring frequency can be customized by a constant variable `CRIS_RECUR_FREQUENCY`, default value is 9.

Example data are as follows:

Note: ~~Total stock and~~ total revenue can have at least 2 decimal places if there are any decimals

	AU_INV_START.txt	AU_INV_END.txt
Example 1	20000101 Y *(Y is a number for total stock)* X *(X is a number for total revenue)*	20050101 YY *(Y is a number for total stock)* XX *(XX is a number for total revenue)*
Example 2	20100202 Y *(Y is a number for total stock)* X *(X is a number for total revenue)*	20150202 YY *(Y is a number for total stock)* XX *(XX is a number for total revenue)*

3 Important Notes

3.1 Documentation

Commenting your code is essential as part of the assessment criteria (refer to Section 3.2).

You should also include comments at the beginning of your program file, which specify your name, your student ID, the start date, and the last modified date of the program, as well as with a high-level description of the program.

In-line comments within the program are also part of the required documentation. References to any source material used to develop your assignment should be included in the in-line comments (URL and the date of retrieval is sufficient).

(Please write this properly as we can see these details from your file properties)

3.2 Marking Criteria

The assessment of this assignment will be based on the following marking criteria:

- 60% for working program - functionality;
- 10% for code architecture - algorithms, data types, control structures, and use of libraries;
- 10% for coding style - clear logic, clarity in variable names, and readability;
- 20% for documentation - program comments.
- **The program logic is going to be marked by auto maker.**

You cannot ask mentors about whether your output is correct or not in any circumstance.

You can discuss your logic with your mentor, they will provide general answers for this.

4 Submission

There will be NO hard copy submission required for this assignment. You are required to submit your assignment as a **.zip** file name with your Student ID.

For example, if your Student ID is 31303030, you would submit a zipped file named “A1_31303030.**zip**”.

Note that marks will be deducted if this requirement is not strictly complied with (For example, uploaded an .rar file).

Your submission must be done via the assignment submission link on the FIT9136 S1 2020

Moodle site by the deadline specified in Section 1. Submissions **will not be accepted** if left in Draft mode. Be sure to submit your files before the deadline to not incur late penalties.

4.1 Deliverables

Your submission should contain the following documents:

Two Python scripts named as follows:

- {YourStudentID}_task1.py
- {YourStudentID}_task2.py
  (Please note you do not need include {} inside your file name)

NOTE: Your programs must at least run on the computers in the University's computer labs. Any submission that does not run accordingly will receive no marks.

Electronic copies of ALL your files that are needed to run your programs.

Your programs must be written as python scripts. Jupyter Notebooks or any other IDE **will not be accepted** as part of submission for this assignment.

Marks will be deducted for any of these requirements that are not strictly complied with.

4.2 Academic Integrity: Plagiarism and Collusion

Plagiarism - Plagiarism means to take and use another person's ideas and or manner of expressing them and to pass them off as your own by failing to give appropriate acknowledgement.

This includes materials sourced from the Internet, staff, other students, and from published and unpublished works.

Collusion - Collusion means unauthorised collaboration on assessable work (written, oral, or practical) with other people. This occurs when you present group work as your own or as the work of another person. Collusion may be with another Monash student or with people or students external to the University. This applies to work assessed by Monash or another university.

It is your responsibility to make yourself familiar with the University's policies and procedures in the event of suspected breaches of academic integrity. (Note: Students will be asked to attend an interview should such a situation is detected.)

The University's policies are available at: http://www.monash.edu/students/academic/policies/academic-integrity