UNIVERSITI TUNKU ABDUL RAHMAN

FHCT1024 PROGRAMMING CONCEPTS AND DESIGN

FOUNDATION IN SCIENCE (Session 202110)

GROUP ASSIGNMENT GUIDELINES

ASSESSMENT: The assignment constitutes 2**0%** of the final grade

ASSIGNMENT THEME:
"Ingredients Requirement Planning Calculator for Baking Recipes"

This assignment will assess the following Course Learning Outcomes (CLO) and Domain:

CLO	CLO Description	Domain
CLO 3	Demonstrate ways in solving computing problems using various control structures and text files management.	Cognitive

1.0 Objectives

This assignment is designed to enable you to put into practice the knowledge and concepts you have acquired over the trimester on problem solving and Python Program development to solve a computing problem effectively.

A problem will be provided to you where you will need to employ the program development processes (Six (6) Steps of Program Development) to assist you to build a viable system. These six (6) steps include *defining the problem*, *developing an algorithm*, *testing the algorithm*, *coding the algorithm*, *running the program*, and *documenting & maintaining the program*. [For more details, refer to Lecture Topic 1.]

You are expected to fully cooperate with your team members to take a detailed study of the problem (domain knowledge), decompose a complex problem into smaller, manageable parts so that the entire team can work hand in hand to build efficient and effective modules and functions that can be integrated easily to form an integral solution.

You are expected to inculcate the needed level of problem-solving, coding skills as well as to acquire all related soft skills needed in solving a computing problem in a team at the end of this assignment.

2.0 Assignment Rules

1. **Submission Date**: 20th December 2021, Monday (Week 11) before 12pm

2. **Group size: 4-6** persons in a group.

3. **Submission: a zipped folder that is submitted to Google Form link (will be posted to WBLE)**. It must consist of:
   - ONE (1) report (.docx), and
   - ONE (1) or more source code (.py)

4. **Penalty:** No marks (0%) will be awarded to everyone in the group if **plagiarism** is found, **and** 10% of the entire assignment marks will be deducted each day for **late submission**.

5. **Important Notes to Every Team Member**:
   - Each of you **MUST** contribute to the completion of the assignment.
   - You are **responsible** for forming your own assignment group and choosing your own group leader.
   - You can create a group on MS Teams to discuss or do your assignment. Be **responsive** to other group members.
   - Exchange contact details (emails and phone numbers) and stay contactable.
   - Your lecturer has the right to deduct your assignment marks if you are found to be not playing an active role in completing the assignment.

3.0 Requirements / Output

1.0 Requirements for report:

- You are expected to prepare a **NEAT** and **LEGIBLE** report.
- You are to include **ONLY** those sections listed (table of contents, flowchart, program, screen shots of the system) in the template. **NO additional section(s) is/are allowed.**

2.0 Requirements for algorithm:

- You can use any flowchart-drawing applications to draw your flowcharts. **Hand drawn flowcharts are not allowed.**
- You are advised to construct well-documented algorithm(s) (flowcharts) showing all the detailed steps, in a correct manner, on how the entire system works.
- You are to use multiple pages with clearly labelled page connectors to document your solution and **NOT** to cramp too many symbols into one page. You must provide the right symbols for each of the algorithmic steps you use to solve the problem. You will also need to use the correct font and size (standardize to either Arial or Times New Roman) for your labelling.

3.0 Requirements for Python Program:

- Program written must:
  - be error-free and can produce the right output.
  - be well-thought-out, with the use of the right combination of control structures and subprograms (self-defined functions).
  - be well-presented screen interfaces with the ease to navigate from one screen to another.
  - include data validation, along with, meaningful error messages to alert the user when he or she keys in invalid data.
  - be well documented (internal documentation) but not excessively
- Your program code must be tidy and readable. You are advised to adhere to a good coding standard and style.
  - Adopt a naming convention for constants or variables (eg., prefix of g for global variables).
  - Use tab instead of space for indentation (4 spaces for indentation).
  - Program block (number of steps in each self-defined function/control structures) should not be too large/small.
  - Use equal-width font (eg. courier new) for the codes in the report.

4.0 Choose to enhance the system to include special features and functions that can simulate actual operations of such system.

4.0 Problem / Question

You are required to develop a system that can be used to perform ingredients requirement planning for baking recipes. The system should, at the very minimum be able to operate as such:

1. Create, Read, Update and Delete (CRUD) any ingredient and/or quantity of stock (inventory) currently available for baking in your current setup.
2. Create and/or edit the order(s) to bake.
3. Calculate and check the inventory for any shortfall of ingredients required to fulfil the orders. Provide the user on the shortfall according to the unit of measure (UOM) used.
4. All records on inventory/ingredients available, orders and the computation of ingredients required are to be stored in text file(s).

You are required to have the following feature in your system:

1. A minimum of **THREE (3)** recipes.
2. List of ingredients and quantity available to be used for baking.

Please conduct your own research to gain additional knowledge on ways to enhance the system so that it will be reliable, scalable and maintainable. You may also want to include the following features and function:

1. Recipe maintenance
2. Ingredients inventory maintenance
3. Option to display the recipes and/or planning on the screen and/or save in a file to be printed at a later time.

5.0 Grading (100%)

1. **Algorithm (Flowcharts) (15%)**
   Flowcharts show each step of the solutions in a detailed, clear, and precise manner. Flowcharts are drawn using the correct symbols, line arrows and labels.

2. **Program Specifications / Correctness (30%)**
   All parts of the program work correctly without errors. All basic specifications are met. Your program must produce all expected output accurately. Your program is written according to the steps depicted / represented in the flowcharts.

3. **Program Control and Subprogram Structures (Code Efficiency) (15%)**
   The order in which program instructions are performed should be carefully controlled, and programming problems are reduced to combinations of controlled sequences, selections, and repetitions (loop) in the most efficient manner. You **MUST NOT** to use a recursive function call in your program.

   Programs are organised into subprograms / self-defined functions and are self-contained (independent of other parts of the program) with well-defined data (locally declared variables) and operations.

4. **Interface Design and Flexibility in Navigating between & within Functions and Error Rectification (10%)**
   Clear instructions are provided on the screen for users to navigate through the functions / subprograms / menus / submenus of the system efficiently. Users are also guided with the understanding of what data and data structures to be entered to allow the system to process it into the expected output.

5. **Validation and Error Messages (10%)**
   Proper validation and control are placed to check for correct data type, data range, data length / size, and completeness depending on the use of data structures in the system. Error messages on what goes wrong must be clearly displayed to alert users. Users must be directed clearly on how to go about rectifying the problem.

6. **Documentation and Coding Style (Code Appearance) (10%)**
   Your Python Program is documented in a neat and tidy fashion. Proper header and meaningful comments (internal documentation) are provided accordingly in the program. Line spacing and spaces used in the statements are standardised to allow readability.

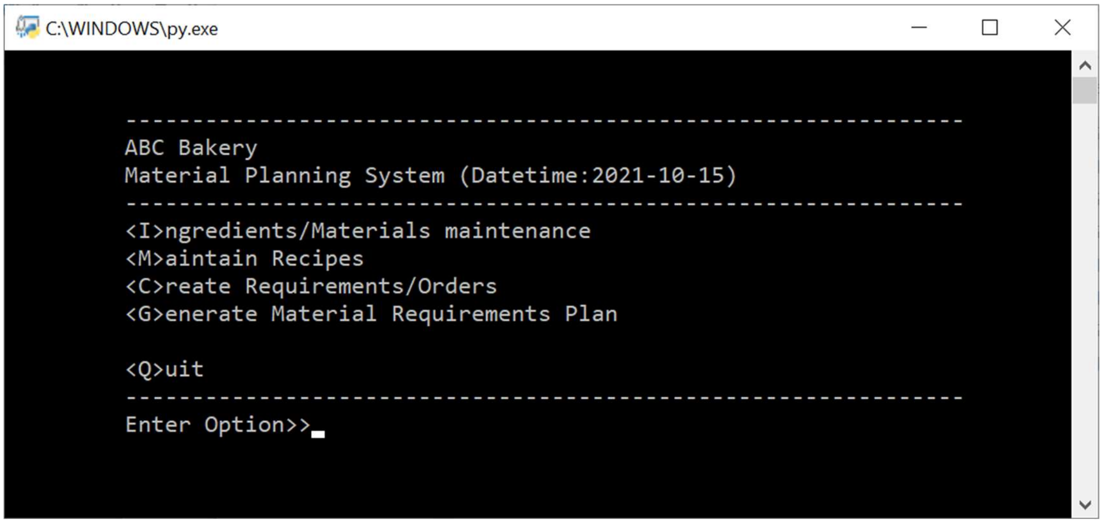
7. **Enhancement of System (10%)**
   Your program is enhanced to cater for features and functions that resemble those of a live (actual working) environment. It is also structured in such a way that it is configurable or adaptable to changes in requirements (minimum changes are required to be made to the program and structure when there is a requirement change) – having little "hardcoding" of any sort.

6.0 Screenshots

Note:

- The following screen shots show some sample interfaces of different functions or modules.
- It is not compulsory to follow the design of the following interfaces. You are encouraged to design your own interfaces.

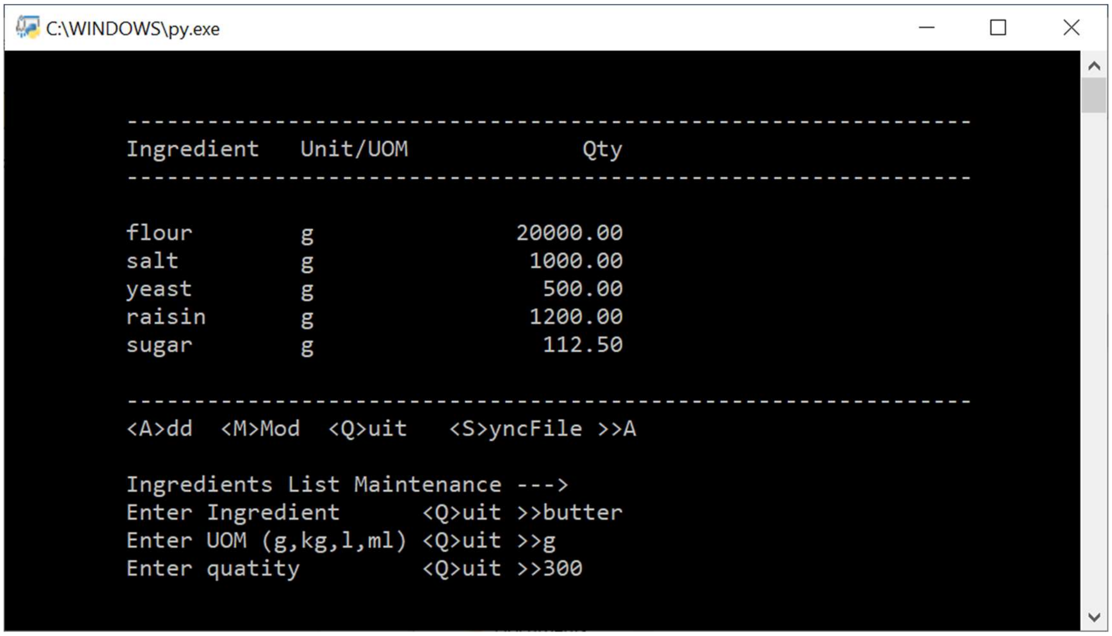
6.1 Main Menu



```
---------------------------------------------------------------
ABC Bakery
Material Planning System (Datetime:2021-10-15)
---------------------------------------------------------------
<I>ngredients/Materials maintenance
<M>aintain Recipes
<C>reate Requirements/Orders
<G>enerate Material Requirements Plan

<Q>uit
---------------------------------------------------------------
Enter Option>>_
```

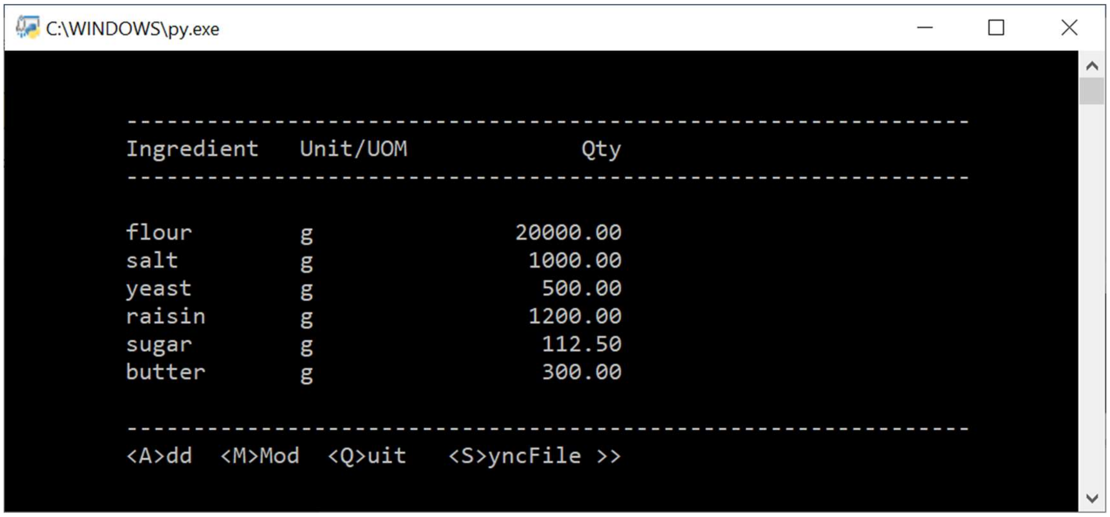
6.2 Ingredient Maintenance

6.2.1 Adding New Ingredient



```
---------------------------------------------------------------
Ingredient   Unit/UOM             Qty
---------------------------------------------------------------

flour        g               20000.00
salt         g                1000.00
yeast        g                 500.00
raisin       g                1200.00
sugar        g                 112.50

---------------------------------------------------------------
<A>dd  <M>Mod  <Q>uit   <S>yncFile >>A

Ingredients List Maintenance --->
Enter Ingredient      <Q>uit >>butter
Enter UOM (g,kg,l,ml) <Q>uit >>g
Enter quatity         <Q>uit >>300
```

6.2.2 New Ingredient Added

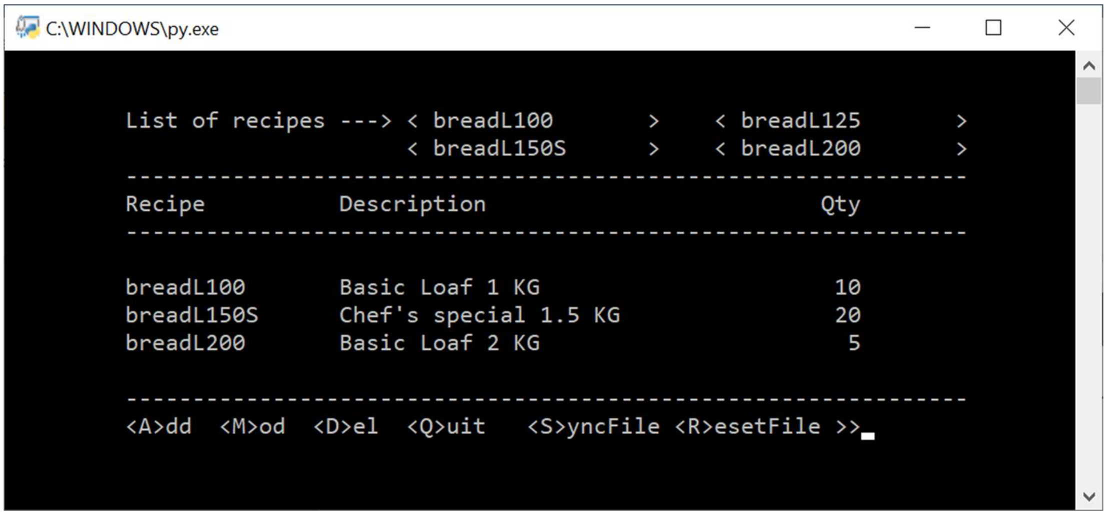


```
-----------------------------------------------------------------
Ingredient   Unit/UOM            Qty
-----------------------------------------------------------------

flour        g                20000.00
salt         g                 1000.00
yeast        g                  500.00
raisin       g                 1200.00
sugar        g                  112.50
butter       g                  300.00

-----------------------------------------------------------------
<A>dd  <M>Mod  <Q>uit   <S>yncFile >>
```

*SyncFile is an option to update the original file. You may choose to program an auto update once there is any changes

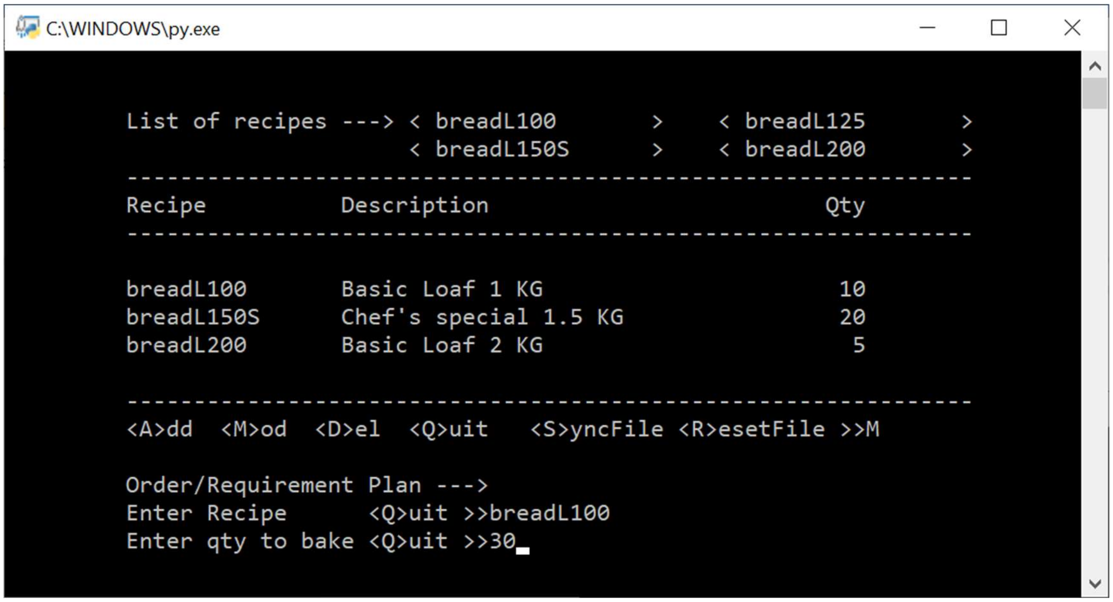
6.3 Order Creation/Update

6.3.1 Existing Order



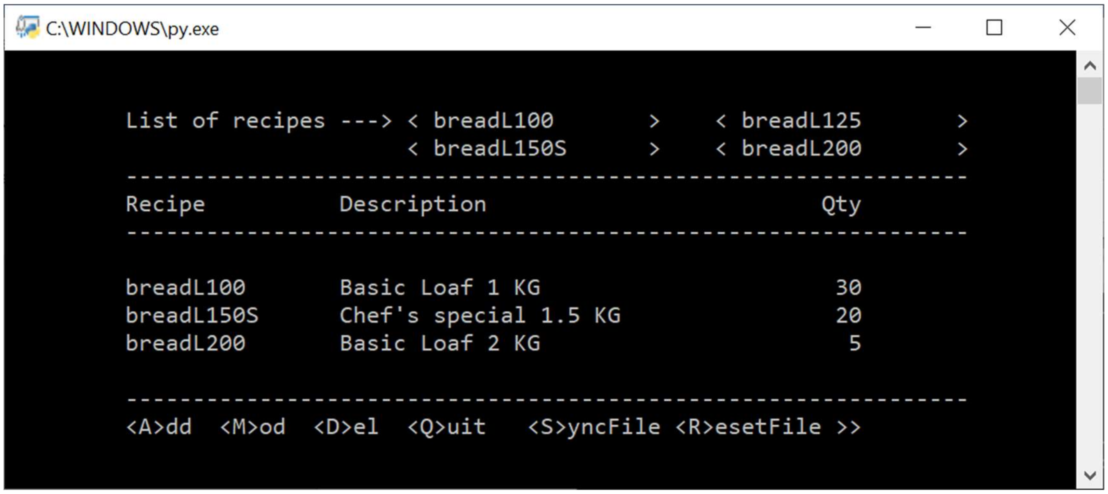
```
List of recipes ---> < breadL100      >    < breadL125      >
                     < breadL150S     >    < breadL200      >
-----------------------------------------------------------------
Recipe          Description                           Qty
-----------------------------------------------------------------

breadL100       Basic Loaf 1 KG                        10
breadL150S      Chef's special 1.5 KG                  20
breadL200       Basic Loaf 2 KG                         5

-----------------------------------------------------------------
<A>dd  <M>od  <D>el  <Q>uit   <S>yncFile <R>esetFile >>
```

6.3.2 Updating Existing Order


C:\WINDOWS\py.exe

```
List of recipes ---> < breadL100       >    < breadL125       >
                     < breadL150S      >    < breadL200       >
---------------------------------------------------------------
Recipe          Description                           Qty
---------------------------------------------------------------

breadL100       Basic Loaf 1 KG                        10
breadL150S      Chef's special 1.5 KG                  20
breadL200       Basic Loaf 2 KG                         5

---------------------------------------------------------------
<A>dd  <M>od  <D>el  <Q>uit   <S>yncFile <R>esetFile >>M

Order/Requirement Plan --->
Enter Recipe      <Q>uit >>breadL100
Enter qty to bake <Q>uit >>30
```

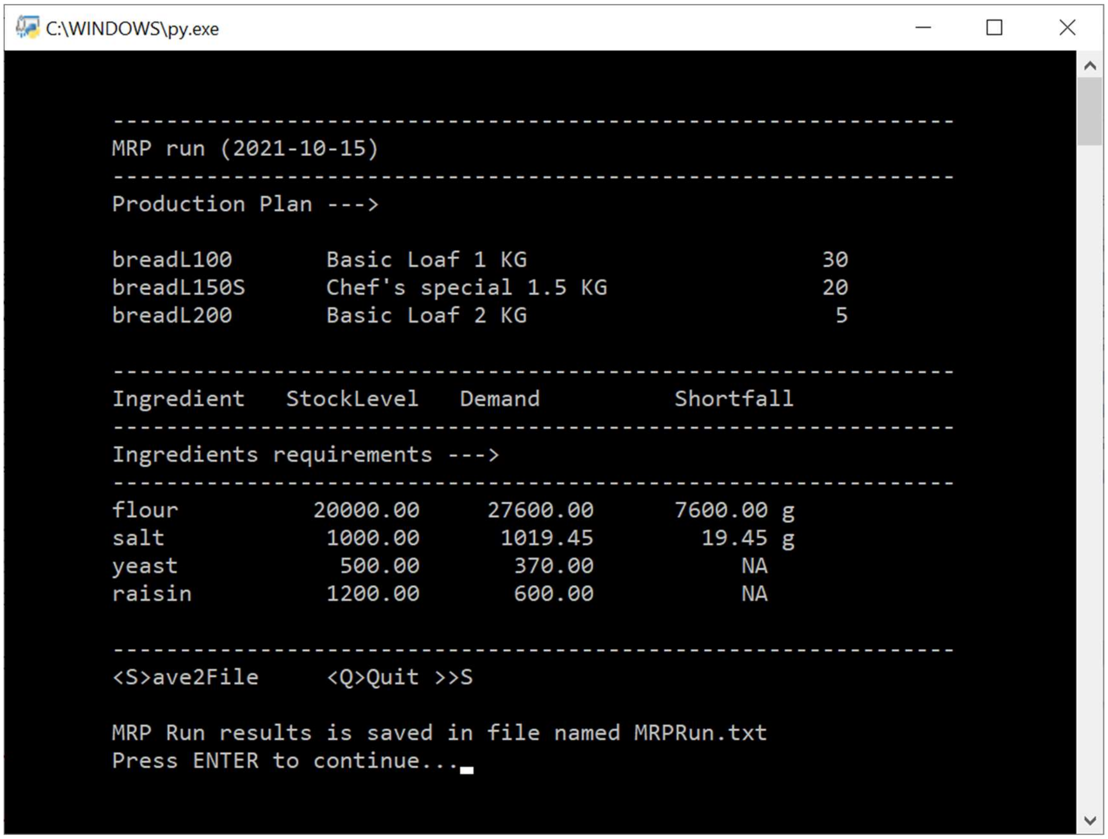
6.3.3 Existing Order Updated


C:\WINDOWS\py.exe

```
List of recipes ---> < breadL100       >    < breadL125       >
                     < breadL150S      >    < breadL200       >
---------------------------------------------------------------
Recipe          Description                           Qty
---------------------------------------------------------------

breadL100       Basic Loaf 1 KG                        30
breadL150S      Chef's special 1.5 KG                  20
breadL200       Basic Loaf 2 KG                         5

---------------------------------------------------------------
<A>dd  <M>od  <D>el  <Q>uit   <S>yncFile <R>esetFile >>
```

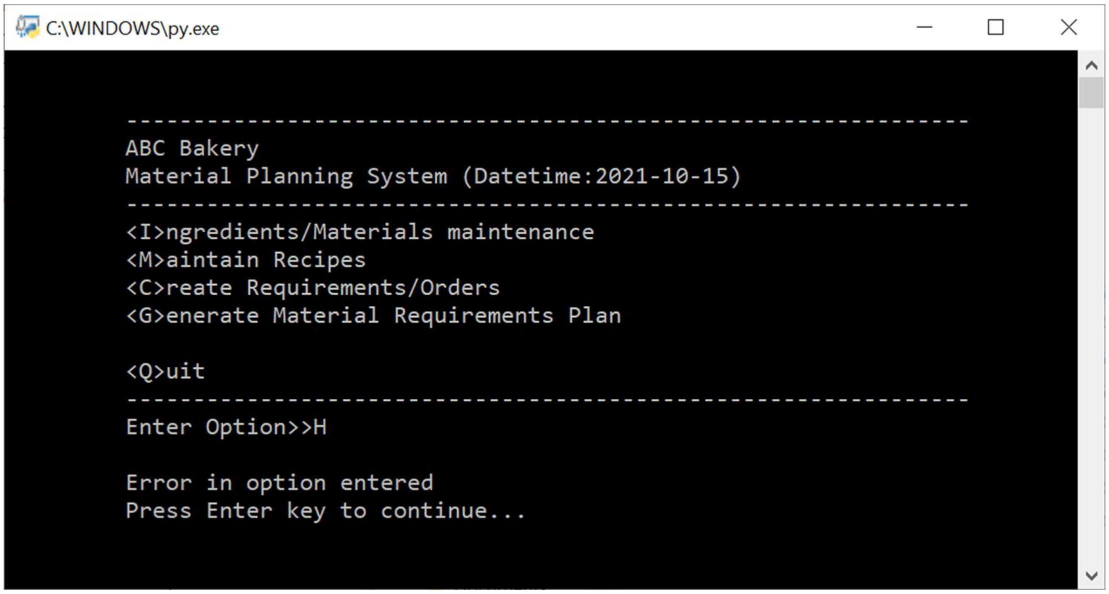
*ResetFile is an option to erase all existing order(s)

6.4 Material Requirement Plan (MRP) and Inventory Generation



```
--------------------------------------------------------------
MRP run (2021-10-15)
--------------------------------------------------------------
Production Plan --->

breadL100       Basic Loaf 1 KG                         30
breadL150S      Chef's special 1.5 KG                   20
breadL200       Basic Loaf 2 KG                          5

--------------------------------------------------------------
Ingredient   StockLevel   Demand          Shortfall
--------------------------------------------------------------
Ingredients requirements --->
--------------------------------------------------------------
flour          20000.00     27600.00      7600.00 g
salt            1000.00      1019.45        19.45 g
yeast            500.00       370.00           NA
raisin          1200.00       600.00           NA

--------------------------------------------------------------
<S>ave2File     <Q>Quit >>S

MRP Run results is saved in file named MRPRun.txt
Press ENTER to continue...
```

6.5 Data Validation



```
--------------------------------------------------------------
ABC Bakery
Material Planning System (Datetime:2021-10-15)
--------------------------------------------------------------
<I>ngredients/Materials maintenance
<M>aintain Recipes
<C>reate Requirements/Orders
<G>enerate Material Requirements Plan

<Q>uit
--------------------------------------------------------------
Enter Option>>H

Error in option entered
Press Enter key to continue...
```

~ END OF GUIDELINES ~