Emerging Trends in IT Diploma in IT/FI Year 3 (2019/20) Semester 2	Week **15 ~ 17**
Assignment	

Part 1 – General Information

1. This is a **GROUP** activity.
2. This assignment is to demonstrate the team's knowledge in **applying DevOps** concepts and best practices to a software development of an application.
3. The software development of the project **must** have Jenkins as the central integrator.
4. A list of required plugins is to be submitted **one week** upon receipt of the assignment. This is to allow for the Jenkins administration to begin installation of the required plugin prior to start of DevOps development environment that the team will be working on.

Part 2 – Assignment - Project Details

1. Refer to appendix A.

Part 3 – Assignment Presentation

1. Each team shall be given 15 ~ 20 minutes of presentation time. This would include the setting up time of the presentation if any.
2. All team members are required to take part in the presentation. It is advisable that all members be given equal amounts of presentation material and related to the assigned role in the development.
3. A short demo of the application and the DevOps setup is required.
4. Be prepared for a short "Question and Answer" session at the end of the presentation.

Part 4 – Assignment Report

1. An assignment report is required to be submitted as part of the deliverables.
2. The assignment report should contain the following:
    a. Summary.
    b. Roles allocation.
    c. Team DevOps setup.
    d. Considerations made for selected DevOps setup.
    e. Approaches adopted at each phase of the Assignment.
    f. Considerations made for the approaches.
    g. QA Test cases used.
    h. Results or outcomes.
    i. Short analysis of the results or outcomes.
    j. References/ Appendices
    k. Contribution list based on member and section.

Part 5 - Assessment Criteria
Group Assessment (60%)

Item	Basic -1	Average -2	Accomplished -3	Advanced -4
Assignment Objective	DevOps intent is not evident	DevOps intent is vague	DevOps intent is explicit	Clearly defined. DevOps intent is explicit
Presentation Organization and Flow	Disorganized, illogical flow	Somewhat organized but lacks flow from one section to another	Well organized but lack flow in some sections	Well organized, logical, clear flow of ideas from section to section
Processes proposed	Standard setup as covered in class with Jenkins as the central integrator.	Standard setup, Jenkins as the central integrator with 1~3 new plugins/ systems used.	Setup differs slightly from the setup taught in class with Jenkins as the central integrator with 1~3 new plugins/ system used.	Setup differs totally from the setup taught in class with Jenkins as the central integrator with all new plugins/ system used.
Results	Some features completed with TDD and tested	Some features completed with TDD and tested with justifications	Most features completed with TDD and tested with justifications	All features completed with TDD and tested
Considerations made at each stage of the assignment.	No considerations made, just follow class setup.	Considerations made for some stages	Considerations made for most stages.	Considerations made for all stages
Understanding of the assignment at each phase.	No discussion and analysis were made, just execute	Proper discussion and analysis made for at least 2 phases	Proper discussion and analysis made for at least 4 phases	Discussion and analysis made for all phases.
Discussion content	Discussion content to quality assurance phase.	Discussion content up to operations phase	Discussion content up to deployment phase	Discussion content up to monitoring phase
Responses to questions during Q and A	Not able to answer questions	Somewhat able to answer questions	Able to answer all questions	Able to answer all questions and offer suggestions and opinions.
Assignment Setup Demonstration	Not workable demonstration	Somewhat workable demonstration	Working demonstration	Working and provides useful statistics

Individual Assessment (40%)

Item	**Basic – Up to 15%**	**Average – 15% to 25%**	**Accomplished – 25% to 35%**	**Advanced – 35% to 40%**
Project Manager	Produces project timeline Demonstrates project tasks allocations and sometimes monitors progress	Produces project timeline Demonstrates project tasks allocations and always monitors progress Does some feature development or testing.	Produces project timeline Demonstrates project tasks allocations and always monitors progress Does feature development or testing.	Produces project timeline Demonstrates project tasks allocations and always monitors progress Does feature development or testing Make analysis of project results.
Technical Lead	Produces development timeline Demonstrates development tasks allocations and sometimes monitors progress	Produces development timeline Demonstrates development task allocations and always monitors progress. Assist in feature development with some passing and failing test cases using TDD	Produces development timeline Demonstrates development task allocations and always monitors progress. Assist in feature development with passing and failing test cases using TDD	Produces development timeline Demonstrates development task allocations and always monitors progress. Assist in feature development with passing and failing test cases using TDD Make analysis of the development test results
Quality Assurance Lead	Produces passing test cases Conducts acceptance tests.	Produces passing and failing test cases Conducts acceptance tests.	Produces passing and failing test cases Conducts acceptance tests Monitors testing results for all tests done.	Produces passing and failing test cases Conducts acceptance tests Monitors testing results for all tests done Makes analysis of the results.

Individual Assessment (40%)

Item	**Basic** **– Up to 15%**	**Average** **– 15% to 25%**	**Accomplished** **– 25% to 35%**	**Advanced** **– 35% to 40%**
Developer	Develop features Conduct some form of Unit Test using TDD	Develop features Conduct Unit Test with passing test cases using TDD	Develop features Conduct Unit Test with passing and failing test cases using TDD	Develop features, Conduct Unit Test with passing and failing test cases using TDD Monitors testing results for all tests done.

Part 6 - Timeline

1. The assignment presentation shall be due on Week 17.
2. The assignment report shall be submitted in softcopy to MEL by **9nd February 2020, 2359hrs** and hardcopy by **10rd February, 0900hrs** to the General Office, Assignment submission drop-off box.
3. Presentation dates and time shall be separately sent out via MEL.

Part 7 – Deliverables

1. Presentation Slides as stated in Part 3
2. Group report as stated in Part 4 with the following items as appendix or references where applicable.
   a. Project Manager
      i. Project timeline
      ii. Project task allocation records
      iii. Development records or Test records
      iv. Project analysis records
   b. Technical Lead
      i. Development timeline
      ii. Development task allocation records
      iii. Development records
      iv. Development analysis records
   c. Quality Assurance Lead
      i. Passing test case records
      ii. Failing test case records
      iii. Acceptance test records
      iv. Test analysis records
   d. Developer
      i. Source codes snippets and source codes link
      ii. Passing unit test case records
      iii. Failing unit test case records
      iv. Unit test records

Part 8 – Peer Review

1. Each member of the team is required to submit separately via email a peer review ranking each member of the team in descending order starting with the one who did the most contribution for the assignment. The peer review is due the **same day** as the assignment submission date. A template of the peer review will be made available in MEL.

Appendix A

1. OBJECTIVE

This assignment assesses the student's ability to apply relevant DevOps concepts and standard best practices to develop a simple application.

2. BACKGROUND

A simple maze game is to be developed and your team is selected for it.

3. SCOPE

A simple maze is to be developed for users to play. You are assigned to develop a simple program.

A data file containing an initial maze diagram is shown in Figure 1 below, where 'X' represents the walls of the maze, 'O' represents the open paths in the maze, 'A' represents the start position of the game and 'B' represents the exit/end position in the maze. You are to read the data from the data file, process it and display the information accordingly.

```
XXXXXXXX
XOOOXOAX
XOXOXOXX
XOXOXOOX
XOXOXXOX
XOXOXOOX
XOXOOOXX
XBXXXXXX
```

Figure 1 – content of data file maze.csv

For this development, you are expected to:

- **Setup and demonstrate the use of DevOps practices up to the Operations phase.**
- **Present findings and considerations made for each phase up to the Operations phase.**
- **Plan, discuss and present the potential use of your DevOps setup during the deployment phase using continuous deployment.**
- **Implement, test and verify each feature of the application using Test-Driven Development (TDD)**
- **Produce relevant deliverables in each phase up to the Operations phase.**

4. Application Requirements and Sample Screens

- **Display main menu** (and allow for repetition)

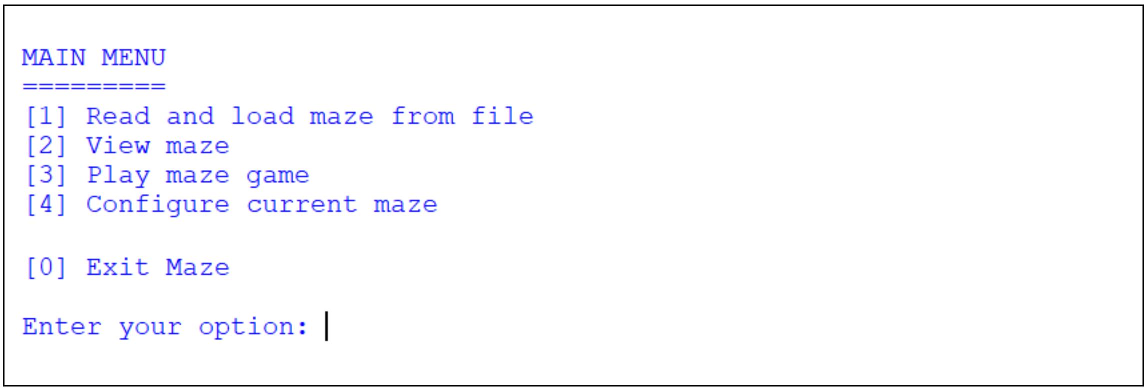
  When the program is run, it should display the main menu as shown in Figure 2. When a user enters an option from 0 to 4, the program will process the option accordingly. After the option has been processed, the program will display the main menu again and the process is repeated until the user chooses to exit.

Main menu

1. **Read and load maze from file**
2. **View maze**
3. **Play maze game**
4. **Configure current maze**

0. **Exit Maze**

Enter your option:

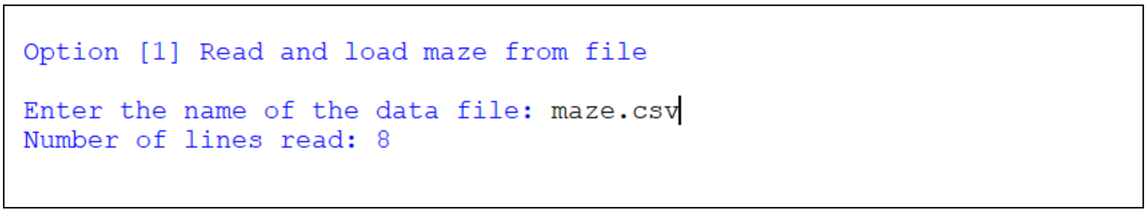


```
MAIN MENU
=========
[1] Read and load maze from file
[2] View maze
[3] Play maze game
[4] Configure current maze

[0] Exit Maze

Enter your option: |
```

Figure 2 - Main Menu

- **Read and load maze from file**

  This feature allows the application to read the maze diagram from the data file and store it in memory. It also stores the start (A) location and the end (B) location. The application should prompt the user for the name of the data file. The number of records/lines read is then displayed as shown in Figure 3:

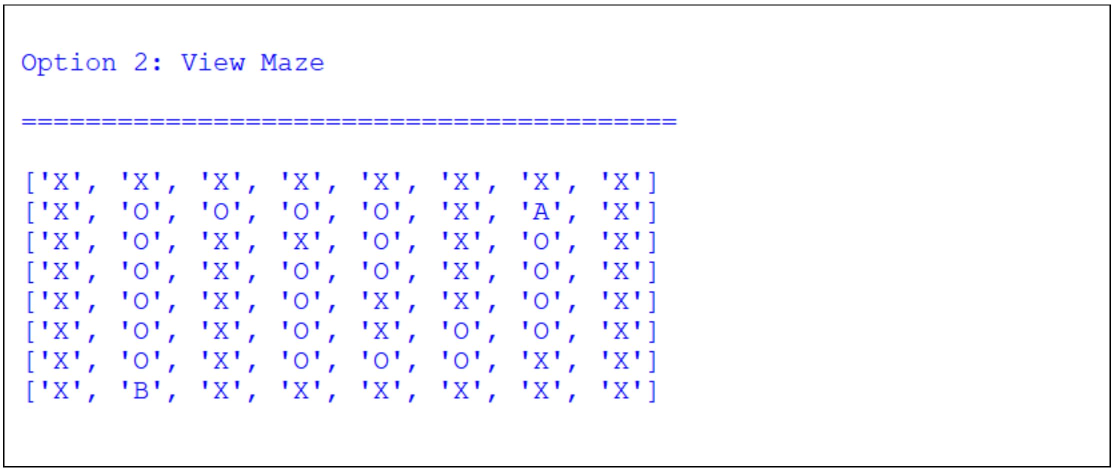


```
Option [1] Read and load maze from file

Enter the name of the data file: maze.csv
Number of lines read: 8
```

Figure 3 – Read and load maze from file

- **View maze**

  This feature allows the user to view the stored maze from the list created, the start and end location done in option 1 as shown in Figure 4:



```
Option 2: View Maze

=========================================

['X', 'X', 'X', 'X', 'X', 'X', 'X', 'X']
['X', 'O', 'O', 'O', 'O', 'X', 'A', 'X']
['X', 'O', 'X', 'X', 'O', 'X', 'O', 'X']
['X', 'O', 'X', 'O', 'O', 'X', 'O', 'X']
['X', 'O', 'X', 'O', 'X', 'X', 'O', 'X']
['X', 'O', 'X', 'O', 'X', 'O', 'O', 'X']
['X', 'O', 'X', 'O', 'O', 'O', 'X', 'X']
['X', 'B', 'X', 'X', 'X', 'X', 'X', 'X']
```

Figure 4 – View Maze

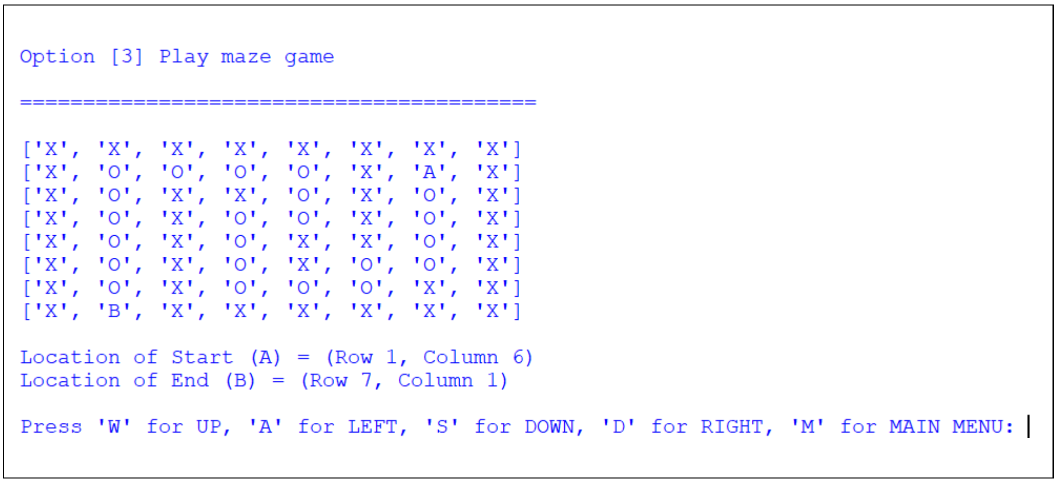
NOTE: The code for this option should not read the maze from the data file again but instead from the stored memory.

- **Play maze game**

  This option allows the user to play the maze game. It displays the maze from the list created, the start and end location as was done in Option 2 and the user can enter one of the following characters: W,A,S,D or M for moving UP, LEFT, DOWN, RIGHT or MAIN MENU respectively, the program will process the option accordingly if it is a valid option.

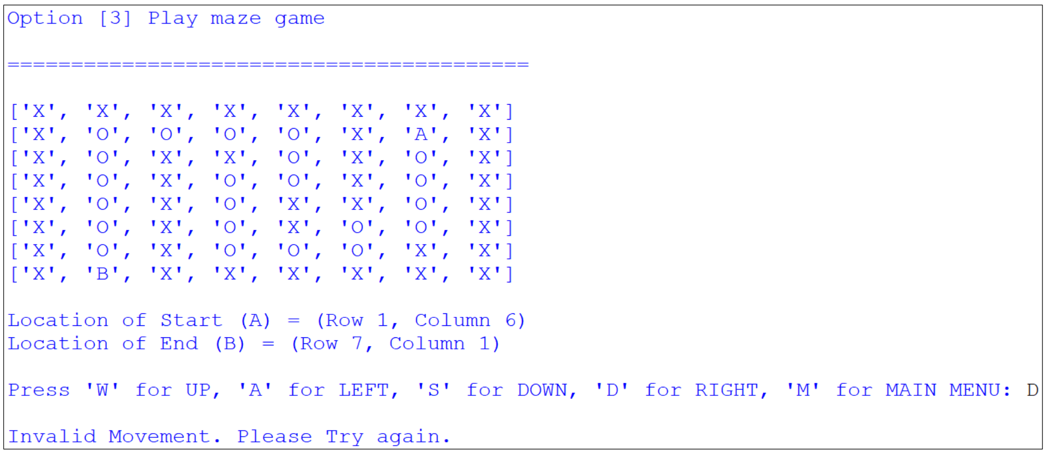
  I.e. if the user enters W and there is a path for going up, then the program will move the START (A) up one space, otherwise it stays in the current location and a message that the move is invalid will be displayed.

After the option has been processed, the program will display the maze again showing the new position of the start point and end point for the maze and the process is repeated until the user chooses to exit to the main menu.



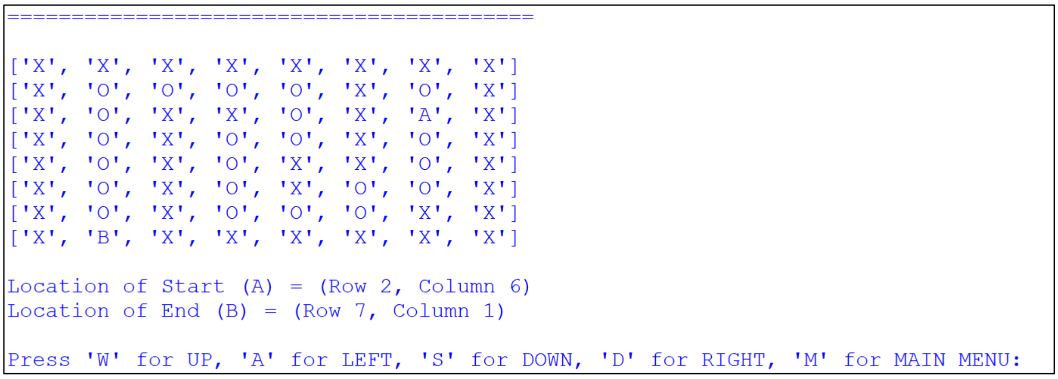
```
Option [3] Play maze game

=========================================

['X', 'X', 'X', 'X', 'X', 'X', 'X', 'X']
['X', 'O', 'O', 'O', 'O', 'X', 'A', 'X']
['X', 'O', 'X', 'X', 'O', 'X', 'O', 'X']
['X', 'O', 'X', 'O', 'O', 'X', 'O', 'X']
['X', 'O', 'X', 'O', 'X', 'X', 'O', 'X']
['X', 'O', 'X', 'O', 'X', 'O', 'O', 'X']
['X', 'O', 'X', 'O', 'O', 'O', 'X', 'X']
['X', 'B', 'X', 'X', 'X', 'X', 'X', 'X']

Location of Start (A) = (Row 1, Column 6)
Location of End (B) = (Row 7, Column 1)

Press 'W' for UP, 'A' for LEFT, 'S' for DOWN, 'D' for RIGHT, 'M' for MAIN MENU: |
```

Figure 5a – Play maze game



```
Option [3] Play maze game

=========================================

['X', 'X', 'X', 'X', 'X', 'X', 'X', 'X']
['X', 'O', 'O', 'O', 'O', 'X', 'A', 'X']
['X', 'O', 'X', 'X', 'O', 'X', 'O', 'X']
['X', 'O', 'X', 'O', 'O', 'X', 'O', 'X']
['X', 'O', 'X', 'O', 'X', 'X', 'O', 'X']
['X', 'O', 'X', 'O', 'X', 'O', 'O', 'X']
['X', 'O', 'X', 'O', 'O', 'O', 'X', 'X']
['X', 'B', 'X', 'X', 'X', 'X', 'X', 'X']

Location of Start (A) = (Row 1, Column 6)
Location of End (B) = (Row 7, Column 1)

Press 'W' for UP, 'A' for LEFT, 'S' for DOWN, 'D' for RIGHT, 'M' for MAIN MENU: D

Invalid Movement. Please Try again.
```

Figure 5b – Play maze game: invalid move



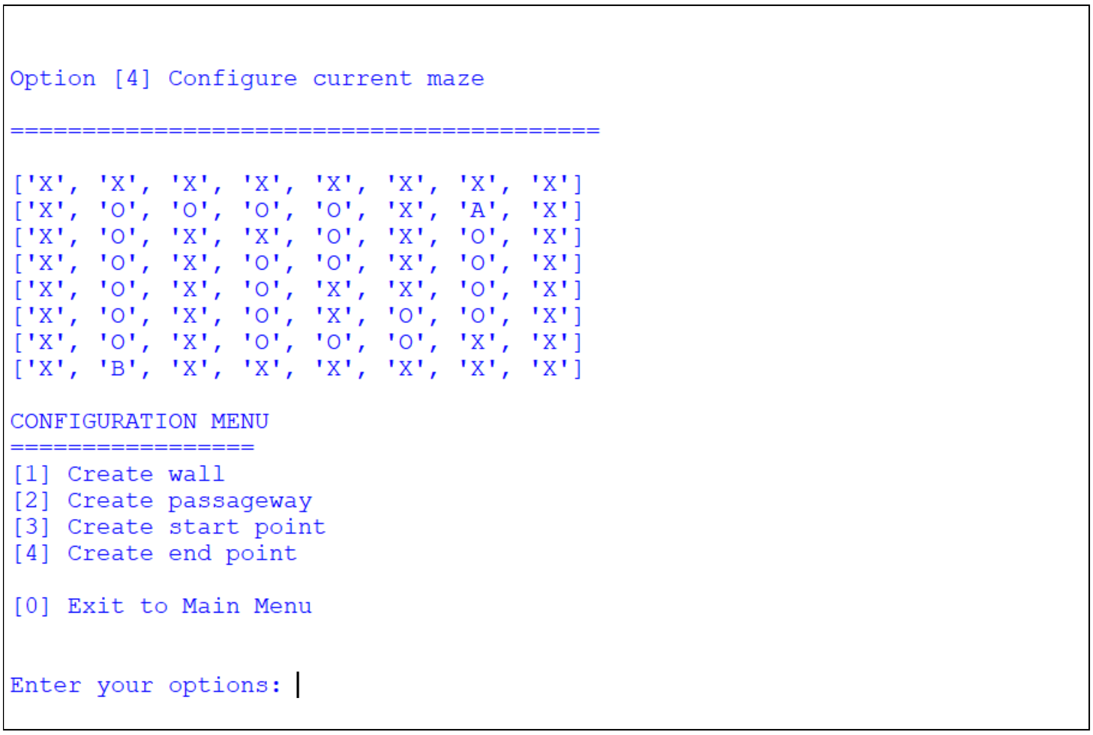
```
=========================================

['X', 'X', 'X', 'X', 'X', 'X', 'X', 'X']
['X', 'O', 'O', 'O', 'O', 'X', 'O', 'X']
['X', 'O', 'X', 'X', 'O', 'X', 'A', 'X']
['X', 'O', 'X', 'O', 'O', 'X', 'O', 'X']
['X', 'O', 'X', 'O', 'X', 'X', 'O', 'X']
['X', 'O', 'X', 'O', 'X', 'O', 'O', 'X']
['X', 'O', 'X', 'O', 'O', 'O', 'X', 'X']
['X', 'B', 'X', 'X', 'X', 'X', 'X', 'X']

Location of Start (A) = (Row 2, Column 6)
Location of End (B) = (Row 7, Column 1)

Press 'W' for UP, 'A' for LEFT, 'S' for DOWN, 'D' for RIGHT, 'M' for MAIN MENU:
```

Figure 5c – Play maze game: update in A for valid move

- **Configure current maze**

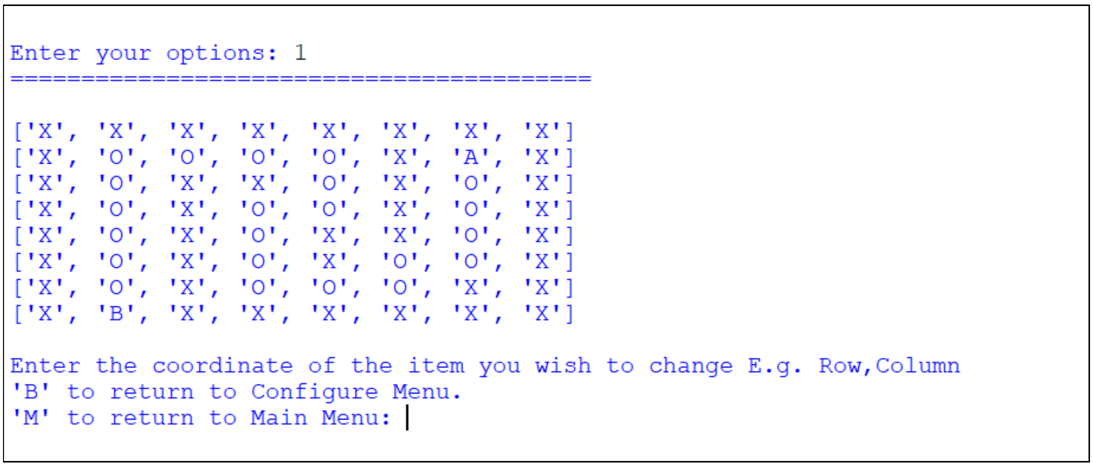
  This feature allows the user to change the configuration of the current maze by changing some of the walls and paths, the start and end locations. The sample outputs show one way this can be done. You are free to design your method for configuring the maze.

  Sample outputs are shown in Figure 6a and 6b.



```
Option [4] Configure current maze

=========================================

['X', 'X', 'X', 'X', 'X', 'X', 'X', 'X']
['X', 'O', 'O', 'O', 'O', 'X', 'A', 'X']
['X', 'O', 'X', 'X', 'O', 'X', 'O', 'X']
['X', 'O', 'X', 'O', 'O', 'X', 'O', 'X']
['X', 'O', 'X', 'O', 'X', 'X', 'O', 'X']
['X', 'O', 'X', 'O', 'X', 'O', 'O', 'X']
['X', 'O', 'X', 'O', 'O', 'O', 'X', 'X']
['X', 'B', 'X', 'X', 'X', 'X', 'X', 'X']

CONFIGURATION MENU
=================
[1] Create wall
[2] Create passageway
[3] Create start point
[4] Create end point

[0] Exit to Main Menu


Enter your options: 
```

Figure 6a – Configure current maze: Configuration Menu



```
Enter your options: 1
=========================================

['X', 'X', 'X', 'X', 'X', 'X', 'X', 'X']
['X', 'O', 'O', 'O', 'O', 'X', 'A', 'X']
['X', 'O', 'X', 'X', 'O', 'X', 'O', 'X']
['X', 'O', 'X', 'O', 'O', 'X', 'O', 'X']
['X', 'O', 'X', 'O', 'X', 'X', 'O', 'X']
['X', 'O', 'X', 'O', 'X', 'O', 'O', 'X']
['X', 'O', 'X', 'O', 'O', 'O', 'X', 'X']
['X', 'B', 'X', 'X', 'X', 'X', 'X', 'X']

Enter the coordinate of the item you wish to change E.g. Row,Column
'B' to return to Configure Menu.
'M' to return to Main Menu: 
```

Figure 6b – Configure current maze: Option 1