SINGAPORE UNIVERSITY OF SOCIAL SCIENCES

ICT133
Structured Programming

Tutor-Marked Assignment

January 2021 Presentation

TUTOR-MARKED ASSIGNMENT (TMA)

This assignment is worth 18 % of the final mark for ICT133, Structured Programming

The cut-off date for this assignment is **Wednesday, 03 Mar 2021, 2355hrs.**

Assignment Requirements:

- Do NOT define classes for this TMA.
- **Unless specified in the question**, **you CANNOT use packages not covered in this module, e.g., numpy, pandas etc.**
- Provide sufficient comments to your code, and ensure that your program adheres to good programming practices such as not using global variables.
- **Do not use the exit() function.**
- Failing to do so can incur a penalty of as much as 50% of the mark allotted.

Submission Details:

- Use the template word document provided - **SUSS_PI_No-FullName_TMA.docx.** Rename the file with your SUSS PI and full name join with "_TMA" e.g. "**SUSS_PI-TomTanKinMeng_TMA.docx**" (without the quotes).
- Include the following particulars on the first page of the word document, on separate lines: **Course Code, SUSS PI No., Your Name, Tutorial Group and Submission Date**.
- Copy and paste the source code of each program you write in **text** format. Submit screenshots for **only** output of your program, where applicable.
- If you submit the source code as a screenshot, your code will not be marked. That is, **screenshot code will be awarded zero mark.**
- Submit your solution in the form of a **single MS Word document. Do NOT submit as a pdf document**. You will be penalised if you fail to submit a word document.

Answer all questions. (Total 100 marks)

Question 1

(a) Suppose Jane gets her groceries from several supermarkets, each with a different number of aisles. Suppose Jane gets items from two aisles per supermarket visit, and to reduce the amount of carrying she needs to do, she places a basket at the beginning of one aisle. Assume that Jane will not visit the two aisles more than once.

Jane wishes to know which aisle she should place her basket so that she can reduce the distance she needs to walk from where she places the basket to the start of the aisle she picks items from, and from the start of the aisle where she picks items from to the basket to place her picked items. Assume the aisles are 1 unit distance from one another.

Write and express a program to perform the following tasks:

- Read the number of aisles in the supermarket that Janet is currently at.
- Read the two aisles Jane needs to get items from. You can assume that two numbers entered are between 1 and the number of aisles in the supermarket.
- Compute and display the minimum distance she needs to walk to the basket. You may apply any algorithm to compute the minimum distance.

Refer to Figure Q1 for an example algorithm for how the minimum distance can be computed. Suppose there are 5 aisles in a particular supermarket, and Jane needs to pick items from aisle 1 and aisle 4, represented by X. If Jane places her basket in front of aisle 3, represented by **B**, then the distance she needs to walk is (3-1) x 2 + (4-3) x 2 = 6 units.

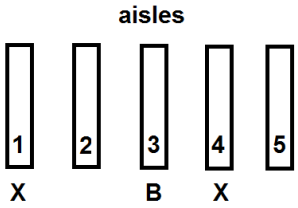

Figure Q1

Example run:

```
Enter number of aisles in supermarket: 5
Enter first aisle: 1
Enter second aisle: 4
The minimum distance is 6 units
```

(4 marks)

(b) Make **TWO (2)** copies of your answer to Question 1(a) and modify a copy for each of the two parts (i) and (ii).

(i) Your program will first prompt for whether Jane needs to visit one aisle. If the first non-whitespace character of the input is y (case insensitive) the program displays the minimum distance as 0. Otherwise, the program prompts for the number of aisles in the supermarket and the two aisles that Jane needs to visit. As in Question 1(a), the program performs the same computation and displays the minimum distance.

Example runs

Run 1

```
Do you need to go to just one aisle? (yes/no):    ya
The minimum distance is 0 units
```

Run 2

```
Do you need to go to just one aisle? (yes/no): nah
Enter number of aisles in supermarket: 10
Enter first aisle: 4
Enter second aisle: 6
The minimum distance is 4 units
```

(ii) Your program will first prompt for the number of aisles Jane needs to visit.

- If the number of aisles is 1, the minimum distance is 0.
- If the number of aisles is 2, the program prompts for the number of aisles in the supermarket and the two aisles that Jane needs to visit. As in Question 1(a), the program performs the same computation and displays the minimum distance.
- If the number of aisles is any other number other than 1 or 2, the program displays the error message: `Invalid number of aisles for Jane!`

Example runs

Run 1

```
Enter the number of aisles to visit: 1
The minimum distance is 0 units
```

Run 2

```
Enter the number of aisles to visit: 2
Enter number of aisles in supermarket: 10
Enter first aisle: 4
Enter second aisle: 6
The minimum distance is 4 units
```

Run 3

```
Enter the number of aisles to visit: 5
Invalid number of aisles for Jane!
```

(6 marks)

(c) Make **TWO (2)** copies of your answer to Question 1(a) and modify a copy for each of the two parts (i) and (ii).

(i) Your program will first prompt for the number of supermarkets Jane has to visit. For each visit, prompt for the number of aisles the supermarket has and the two aisles Jane has to visit. As in Question 1(a), the program performs the same computation and displays the minimum distance for each supermarket visit.

Example run

```
Enter number of supermarkets to visit: 3
Enter number of aisles in supermarket 1: 5
Enter first aisle: 1
Enter second aisle: 4
The minimum distance is 6 units
Enter number of aisles in supermarket 2: 10
Enter first aisle: 4
Enter second aisle: 6
The minimum distance is 4 units
Enter number of aisles in supermarket 3: 100
Enter first aisle: 1
Enter second aisle: 2
The minimum distance is 2 units
```

(ii) Your program will first prompt for the number of aisles the supermarket has. The program then repeatedly prompts for the aisle numbers that Jane needs to visit. When 0 is entered, the input for aisle numbers is terminated.

The program computes the minimum distance based on the best location Jane place her basket and the number of aisles she needs to pick items from. The program then displays the minimum distance.

Example runs

Run 1

```
Enter number of aisles in supermarket: 100
Enter the aisle number to visit: 1
Enter the aisle number to visit: 8
Enter the aisle number to visit: 5
Enter the aisle number to visit: 4
Enter the aisle number to visit: 10
Enter the aisle number to visit: 0
The minimum distance is 26 units
```

Run 2

```
Enter number of aisles in supermarket: 100
Enter the aisle number to visit: 1
Enter the aisle number to visit: 0
The minimum distance is 0 units
```

Run 3

```
Enter number of aisles in supermarket: 100
Enter the aisle number to visit: 0
Not visiting any aisle
```

(10 marks)

Question 2

A factory makes products using parts that are identified by letters A to L. Each product is made of at least 3 different parts, and is identified by a product code made up of letters identifying the parts. The letters in a product code is sorted alphabetically. E.g., a product with product code ABBBG is made up of 1 A, 3 B and 1 G parts.

(a) Write and develop a function `getPartsInCode` which has one string parameter: `productCode`. The function returns a string in this format: $n_1p_1\ n_2p_2 \ldots\ n_kp_k$. Note n_ip_is are separated by space. For example, if product code is ABBBG, then the function returns 1A 3B 1G.

Note that the letters in the string parameter are in uppercase and are already sorted. You can use collections of only str type for function. This means collection types such as set, list, dict are not allowed.

(5 marks)

Apply and employ structured programming and write an application to manage the factory. Other than the data specified in Q2(b), you may use collections of any types for other data in your application. However, you should **assume that the user input for product code is not sorted alphabetically**.

(b) Initialise the following program data in your main method (that is, these variables are not global variables):

- The stock level for each part is 100 at the start of the application.
  `startLevel = 100`
- The reorder point for each part is 20. Once the stock level is 20 or below, the part should be reordered.
  `reorderPoint = 20`
- A string records the letters A – L. One letter is used to identify a part in the factory.
  `partIds = 'ABCDEFGHIJKL'`
- An un-nested list records the stock level of each part. The stock level for each part starts with the value recorded in `startLevel`. For example, stock level for part A should be recorded at the 0-index, stock level for part B should be recorded at the 1-index, etc.
- Use another un-nested list to store existing product codes. Start the list empty.

(2 marks)

(c) Offer this menu repeatedly until option 0 is selected to end the application:

Menu
1. Add product code
2. List inventory
3. Update inventory
4. Make product
5. Get summarised data
0. Exit

Respond to the menu choices according to the description below.

(3 marks)

- Add product code

  The program reads a product code. Note that as the letters may be unsorted when it is entered and the letters may be in uppercase or lowercase. The program first sorts the product code and then checks whether the code is valid. A valid code fulfills these criteria:
  - The code is at least 3 characters in length.
  - Each character is a letter from A to L.

  If the code is valid, the program adds it to a list of product codes if the code is not already in the list of product codes. Display the following messages where appropriate:
  - The product code is invalid
  - The product code already exists
  - The product code has been added successfully

Example runs

Run 1

```
Enter new product code: abcz
The product code is invalid
```

Run 2

```
Enter new product code: abdc
The product code has been added successfully
```

Note that the code added is ABCD (sorted and in uppercase)

Run 3

```
Enter new product code: acbd
The product code already exists
```

(5 marks)

- List inventory

  The stock level for each part is listed. Note that parts that should be reordered are displayed with asterisk.

Example run

```
Part  Stock Level
A     100
B     100
C     100
D     100
E      12  ***
F     100
G     100
H      20  ***
i     100
J     100
K     100
L     100
Legend: *** stock level at or lower than reorder point 20
```

(3 marks)

- Update inventory
  This option allows the stock level of one or more parts to be restocked. The program repeatedly reads part identifier and the amount to add to the stock level for the part, until an empty string is entered for the part identifier. Each update of a part should be recorded in a file, `transactions.txt` on separate line in this format:
  `restock letter quantityAdded`

Example run

```
Enter part identifier or <ENTER> to end: T
The part identifier is invalid
Enter part identifier or <ENTER> to end: E
Current stock level for E = 12
Enter quantity to add: 0
The quantity is invalid
Enter part identifier or <ENTER> to end: E
Current stock level for E = 12
Enter quantity to add: -2
The quantity is invalid
Enter part identifier or <ENTER> to end: E
Current stock level for E = 12
Enter quantity to add: 60
Updated stock level for E = 72
Enter part identifier or <ENTER> to end: F
Current stock level for F = 100
Enter quantity to add: 10
Updated stock level for F = 110
Enter part identifier or <ENTER> to end:
```

Content of transaction.txt

```
restock E 60
restock F 10
```

(7 marks)

- Make product
  This option allows a product code and a quantity to make to be entered. Again, assume that the letters in product code is unsorted and may be in uppercase or lowercase. You must use the function `getPartsInCode` in Q2(a).

  Display the following messages where appropriate:

  - `Invalid product code y`
    where y is an non-existing product code.
  - `Invalid quantity x`
    where x is the a quantity to make, if x is zero or negative.
  - `x product y successfully made`
    where x is the quantity made and y is the product code.
    The inventory should be updated.
  - `x product y made at the current inventory level. z outstanding.`
    where x is the quantity made, z is the quantity that cannot be fulfilled, and y is the product code.
    The inventory should be updated according to the quantity made.

If a part is made and/or if there is outstanding product quantity, the details should be recorded in a file, `transactions.txt` on separate line in this format:

- if the product is made, record in this format:
  `make code quantityMade`
- if there is outstanding product quantity, record in this format:
  `outstanding code quantityOutstanding`

Example runs

Run 1

```
Enter product code: ABCD
Invalid product code ABCD
```

Run 2

```
Enter product code: ABBB
Enter quantity to make: -2
Invalid quantity -2
```

Run 3

```
Enter product code: ABBB
Enter quantity to make: 2
2 product ABBB successfully made
```

Run 4

```
Enter product code: ABBB
Enter quantity to make: 2
Insufficient inventory.
1 product ABBB made at the current inventory level.
1 outstanding.
```

Content of transaction.txt

```
… # add to previous contents
make ABBB 2
make ABBB 1
outstanding  ABBB 1
```

(10 marks)

- Get summarised data
  Prompt and read the type of data (0-restock, 1-make, 2-outstanding) the summary is for. Then, open a file `transactions.txt` with an appropriate file mode, to produce the summarized report, sorted by quantity followed by either by part or product code.

Example runs

Assume the following 12 lines are in transactions.txt:

```
make ABBB 2
make ABBC 2
make ABBB 1
outstanding  ABBB 1
restock E 60
restock F 10
restock A 60
make ABBC 12
make ABBB 2
make ABBB 2
outstanding  ABBB 2
restock F 10
```

Run 1

```
0-restock, 1-make, 2-outstanding. Enter choice: 0
Summarised Data for RESTOCK ***
A           60
E           60
F           20
```

Run 2

```
0-restock, 1-make, 2-outstanding. Enter choice: 1
Summarised Data for MAKE ***
ABBC        14
ABBB         7
```

Run 3

```
0-restock, 1-make, 2-outstanding. Enter choice: 2
Summarised Data for OUTSTANDING ***
ABBB         3
```

(5 marks)

Question 3

Develop an NxN memory card game and allow 2-4 players to play. On the face of each card is a single letter, A-Z. The cards are shuffled and randomly placed face down in N-column and N-row grid.

Players take turn to pick two cards at each turn. Whenever a player picks 2 cards with matching letters, the cards are placed face up, and the player earns one point. The player gets to pick another pair repeatedly until the letters do not match or there are no more cards to pick.

The game can be placed at 2 levels:

Level 1

N is 4, resulting in a 4x4 grid. Letters are randomly chosen and assigned to either 2 cards or 4 cards, repeatedly until 16 cards have been assigned letters.

Level 2

N is an integer between 2 to 7. If the number of cells in the grid is odd, the last grid position is left empty and is marked NIL so only an even number of cards will be used in each game. For example, when N is 3, 5 or 7, there are 8, 24 or 48 cards respectively. Letters are randomly chosen and assigned to either only 1 card or 2 cards. The letters on cards may come in pairs (50% chance) or there can be exactly 2 letters without a match (50% chance).

Implement the memory card game. Your collections can be ONLY of these types: un-nested lists, dict or str collections.

Note: The implementation of level 2 game play is worth 14 marks. You may implement level 1 first and then build level 2 on top of the implementation for level 1.

- Start the application by getting the names of the players. There should be 2 to 4 players with no repeating names. The same set of players will play one or more memory games until the application ends.

  Before each new game commences, the application randomly orders the players, and this will be the order that the players take their turn.

  (4 marks)

- Allow the player to choose a game level. If the input is invalid, allow data entry to be repeated. The valid game level is either 1 or 2.

  If the game level is 1, do not prompt for grid size. If the game level is 2, the players choose a grid size. If the input is invalid, allow data entry to be repeated. The valid grid size is between 2 and 7.

  The application generates the cards according to the grid size (4x4 for level 1) and the constraints as specified for the game level, and the cards are shuffled.

  (11 marks)

- The application displays the cards face down and matched cards are displayed face up. Display NIL in the last grid position when N is odd (for level 2).

  (9 marks)

- At the start of each game, display a cheat sheet to help you test your program. The players take turn to pick until there are no more cards to pick (for both levels) or when the last two remaining cards do not match (for level 2 only).

  Display the score of a player when it is the player's turn to pick two cards. To pick a card, the player specifies the row and column numbers. Validate the row and column numbers, e.g., not NIL (for level 2 only), not a card that is already placed face up (for both levels), not outside grid boundary (for both levels) and the two cards picked are different cards. Display the updated score if the picked cards match.

  (13 marks)

- When the game ends, display the players with the most pairs of cards matched, in alphabetical order of names of players that tie. Then prompt whether the players wish to play a new game. If so, allow the players to choose a grid size and the game level.

  (3 marks)

Example run

```
Enter player name or <ENTER> to exit (min 2, max 4 players):
Minimum 2 players. Currently, 0 player
Enter player name or <ENTER> to exit (min 2, max 4 players): ben
Enter player name or <ENTER> to exit (min 2, max 4 players):
Minimum 2 players. Currently, 1 player
Enter player name or <ENTER> to exit (min 2, max 4 players): BEN
Repeated name. Choose another name
Enter player name or <ENTER> to exit (min 2, max 4 players): cindy
Enter player name or <ENTER> to exit (min 2, max 4 players): alan
Enter player name or <ENTER> to exit (min 2, max 4 players):
Enter game level(1 to 2): -1
Please enter a valid game level
Enter game level(1 to 2): 3
Please enter a valid game level
Enter game level(1 to 2): 1
*** Cheat Sheet ***
                  Columns
               Rows| 1 | 2 | 3 | 4 |
                   -----------------
                  1| H | S | I | S |
                   -----------------
                  2| V | G | T | T |
                   -----------------
                  3| O | G | V | I |
                   -----------------
                  4| O | I | I | H |
```

```
                    -----------------
*** End Cheat Sheet ***
Starting new game ..........
                     Columns
                Rows| 1 | 2 | 3 | 4 |
                    -----------------
                   1|   |   |   |   |
                    -----------------
                   2|   |   |   |   |
                    -----------------
                   3|   |   |   |   |
                    -----------------
                   4|   |   |   |   |
                    -----------------
Player Cindy, enter your guess
Current number of correct pairs: 0
Enter first card row and column, separated by space: 10 10
Please choose a card row and column within the grid
Enter first card row and column, separated by space: 1 1
Open H
Enter second card row and column, separated by space: 1 1
Please choose a different row and column from your earlier choice
Enter second card row and column, separated by space: 4 4
Open H
Correct !!!!
Updated number of correct pairs: 1
                     Columns
                Rows| 1 | 2 | 3 | 4 |
                    -----------------
                   1| H |   |   |   |
                    -----------------
                   2|   |   |   |   |
                    -----------------
                   3|   |   |   |   |
                    -----------------
                   4|   |   |   | H |
                    -----------------
Player Cindy, enter your guess
Current number of correct pairs: 1
Enter first card row and column, separated by space: 1 1
Please choose a card that is not face up yet
Enter first card row and column, separated by space: 2 3
Open T
Enter second card row and column, separated by space: 2 4
Open T
Correct !!!!
Updated number of correct pairs: 2
                     Columns
                Rows| 1 | 2 | 3 | 4 |
                    -----------------
                   1| H |   |   |   |
                    -----------------
                   2|   |   | T | T |
                    -----------------
                   3|   |   |   |   |
                    -----------------
                   4|   |   |   | H |
                    -----------------
Player Cindy, enter your guess
Current number of correct pairs: 2
Enter first card row and column, separated by space: 1 2
Open S
Enter second card row and column, separated by space: 1 3
Open I
Non-matching. No update
                     Columns
                Rows| 1 | 2 | 3 | 4 |
                    -----------------
                   1| H |   |   |   |
                    -----------------
                   2|   |   | T | T |
                    -----------------
                   3|   |   |   |   |
                    -----------------
```

```
                  4|   |   |   | H |
                   -----------------
Player Alan, enter your guess
Current number of correct pairs: 0
Enter first card row and column, separated by space: 1 2
Open S
Enter second card row and column, separated by space: 1 4
Open S
Correct !!!!
Updated number of correct pairs: 1
                       Columns
                Rows| 1 | 2 | 3 | 4 |
                    -----------------
                   1| H | S |   | S |
                    -----------------
                   2|   |   | T | T |
                    -----------------
                   3|   |   |   |   |
                    -----------------
                   4|   |   |   | H |
                    -----------------
Player Alan, enter your guess
Current number of correct pairs: 1
Enter first card row and column, separated by space: 1 3
Open I
Enter second card row and column, separated by space: 3 4
Open I
Correct !!!!
Updated number of correct pairs: 2
                       Columns
                Rows| 1 | 2 | 3 | 4 |
                    -----------------
                   1| H | S | I | S |
                    -----------------
                   2|   |   | T | T |
                    -----------------
                   3|   |   |   | I |
                    -----------------
                   4|   |   |   | H |
                    -----------------
Player Alan, enter your guess
Current number of correct pairs: 2
Enter first card row and column, separated by space: 2 1
Open V
Enter second card row and column, separated by space: 2 2
Open G
Non-matching. No update
                       Columns
                Rows| 1 | 2 | 3 | 4 |
                    -----------------
                   1| H | S | I | S |
                    -----------------
                   2|   |   | T | T |
                    -----------------
                   3|   |   |   | I |
                    -----------------
                   4|   |   |   | H |
                    -----------------
Player Ben, enter your guess
Current number of correct pairs: 0
Enter first card row and column, separated by space: 2 2
Open G
Enter second card row and column, separated by space: 3 2
Open G
Correct !!!!
Updated number of correct pairs: 1
                       Columns
                Rows| 1 | 2 | 3 | 4 |
                    -----------------
                   1| H | S | I | S |
                    -----------------
                   2|   | G | T | T |
                    -----------------
                   3|   | G |   | I |
```

```
                    -----------------
                  4|   |   |   | H |
                    -----------------
Player Ben, enter your guess
Current number of correct pairs: 1
Enter first card row and column, separated by space: 4 2
Open I
Enter second card row and column, separated by space: 4 3
Open I
Correct !!!!
Updated number of correct pairs: 2
                       Columns
                Rows| 1 | 2 | 3 | 4 |
                    -----------------
                   1| H | S | I | S |
                    -----------------
                   2|   | G | T | T |
                    -----------------
                   3|   | G |   | I |
                    -----------------
                   4|   | I | I | H |
                    -----------------
Player Ben, enter your guess
Current number of correct pairs: 2
Enter first card row and column, separated by space: 3 1
Open O
Enter second card row and column, separated by space: 4 1
Open O
Correct !!!!
Updated number of correct pairs: 3
                       Columns
                Rows| 1 | 2 | 3 | 4 |
                    -----------------
                   1| H | S | I | S |
                    -----------------
                   2|   | G | T | T |
                    -----------------
                   3| O | G |   | I |
                    -----------------
                   4| O | I | I | H |
                    -----------------
Player Ben, enter your guess
Current number of correct pairs: 3
Enter first card row and column, separated by space: 2 1
Open V
Enter second card row and column, separated by space: 3 3
Open V
Correct !!!!
Updated number of correct pairs: 4
Game ends!
                       Columns
                Rows| 1 | 2 | 3 | 4 |
                    -----------------
                   1| H | S | I | S |
                    -----------------
                   2| V | G | T | T |
                    -----------------
                   3| O | G | V | I |
                    -----------------
                   4| O | I | I | H |
                    -----------------
Winners with 4 matched pairs: Ben
Play again? <ENTER> an empty string to play again:
Enter game level(1 to 2): 2
Enter grid size(2 to 7): 8
Please enter a valid grid size
Enter grid size(2 to 7): 0
Please enter a valid grid size
Enter grid size(2 to 7): 3
*** Cheat Sheet ***
                       Columns
                Rows| 1 | 2 | 3 |
                    -------------
                   1| W | E | P |
```

```
                    -------------
                  2| K | W | O |
                    -------------
                  3| K | O |NIL|
                    -------------
*** End Cheat Sheet ***
Starting new game .........
                     Columns
               Rows| 1 | 2 | 3 |
                    -------------
                  1|   |   |   |
                    -------------
                  2|   |   |   |
                    -------------
                  3|   |   |NIL|
                    -------------
Player Ben, enter your guess
Current number of correct pairs: 0
Enter first card row and column, separated by space: 3 3
Please choose a different row and column that is not for NIL
Enter first card row and column, separated by space: -1 -1
Please choose a card row and column within the grid
Enter first card row and column, separated by space: 2 1
Open K
Enter second card row and column, separated by space: 3 1
Open K
Correct !!!!
Updated number of correct pairs: 1
                     Columns
               Rows| 1 | 2 | 3 |
                    -------------
                  1|   |   |   |
                    -------------
                  2| K |   |   |
                    -------------
                  3| K |   |NIL|
                    -------------
Player Ben, enter your guess
Current number of correct pairs: 1
Enter first card row and column, separated by space: 1 2
Open E
Enter second card row and column, separated by space: 1 3
Open P
Non-matching. No update
                     Columns
               Rows| 1 | 2 | 3 |
                    -------------
                  1|   |   |   |
                    -------------
                  2| K |   |   |
                    -------------
                  3| K |   |NIL|
                    -------------
Player Cindy, enter your guess
Current number of correct pairs: 0
Enter first card row and column, separated by space: 2 1
Please choose a card that is not face up yet
Enter first card row and column, separated by space: 2 2
Open W
Enter second card row and column, separated by space: 1 1
Open W
Correct !!!!
Updated number of correct pairs: 1
                     Columns
               Rows| 1 | 2 | 3 |
                    -------------
                  1| W |   |   |
                    -------------
                  2| K | W |   |
                    -------------
                  3| K |   |NIL|
                    -------------
Player Cindy, enter your guess
Current number of correct pairs: 1
```

```
Enter first card row and column, separated by space: 2 3
Open O
Enter second card row and column, separated by space: 1 3
Open P
Non-matching. No update
                    Columns
                Rows| 1 | 2 | 3 |
                    -------------
                   1| W |   |   |
                    -------------
                   2| K | W |   |
                    -------------
                   3| K |   |NIL|
                    -------------
Player Alan, enter your guess
Current number of correct pairs: 0
Enter first card row and column, separated by space: 2 3
Open O
Enter second card row and column, separated by space: 3 2
Open O
Correct !!!!
Updated number of correct pairs: 1
Remaining two cards are non-matching: E and P
Game ends!
                    Columns
                Rows| 1 | 2 | 3 |
                    -------------
                   1| W | E | P |
                    -------------
                   2| K | W | O |
                    -------------
                   3| K | O |NIL|
                    -------------
Winners with 1 matched pairs: Alan, Ben, Cindy
Play again? <ENTER> an empty string to play again: n
Memory Game terminating
```

---- END OF ASSIGNMENT ----