



**UNIVERSITI TEKNOLOGI MALAYSIA
FACULTY OF COMPUTING**

TEST 2

SEMESTER I 2022/ 2023

SUBJECT CODE : SECJ1013
SUBJECT NAME : PROGRAMMING TECHNIQUE I
YEAR/COURSE : 1 (SECJH/SECVH/SECBH/SECRH/SECPH)
TIME : 8.00 – 10.30 PM
DATE : 10th JANUARY 2023 (TUESDAY)
VENUE : MPK1 – MPK10, CGMTL, IDAL – N28

INSTRUCTIONS TO THE STUDENTS:

- This test has **TWO** questions. You must answer all of the questions.
- All references to any resources are **strictly prohibited**.
- You have **TWO HOURS AND THIRTY MINUTES** to complete the test, including downloading the test materials and submitting your programs.
- Your programs must adhere to the input and output requirements specified in the text and demonstrated in the examples. You must test the programs with (but not limited to) all of the input provided in the examples.
- All the **COMMENT STATEMENTS** in the submitted program **WILL NOT BE EVALUATED**.

EXAM MATERIALS:

- You are given a compressed file named **Test2_Resource.zip** that contains a source code file with errors (**Test2Q1.cpp**) for Question 1 and a template file (**Test2Q2.cpp**) for Question 2.
- Download the file **Test2_Resource.zip** from UTM's e-learning and decompress it to your computer's local hard drive.
- The provided program files should be used as the base for answering the questions in this test.

SUBMISSION PROCEDURE:

- Only source code files (with the extension.cpp) are required for submission.
- Students are required to name the files as **SectionNo_StudentNameT2Q1.cpp** and **SectionNo_StudentNameT2Q2.cpp** (eg: **04_HalimT2Q1.cpp** and **04_HalimT2Q2.cpp**)
- Submit the source code through the UTM e-learning system.

*This question booklet consists of **12 pages** INCLUDING this page.*

QUESTION 1 - ERROR DEBUGGING**(35 Marks)**

You are given a C++ program (**Test2Q1.cpp**) with 15 errors (syntax errors and/ or logical errors, if any). The program is a Noughts & Crosses or Xs and Os game designed for two (2) players. It has four (4) user-defined functions as listed below:

Function Name	Description
OneUntilNine	To validate either the input is valid or invalid.
Display	To display the game board and shows the number of each box.
WhoWin	To declare the winner either O player wins or X player wins. Determination through the placement of each O or X in the board. If three (3) boxes (O or X) form a line, then the player (O or X) is declared the winner. The game is Draw if all boxes are filled but no line is formed from three (3) boxes.
Occupied	To check either a box is filled or not with X or O. If the box is filled, then player needed to choose other boxes.

Study how all of the above functions were used/ called in the **main** function of the program. You are required to debug the errors, compile, and run the program. You are **NOT ALLOWED** to **remove** any statements in the program. You are only allowed to **update** the statements provided in the program and add a new statement(s) if absolutely necessary. **Table 1** is the three (3) test cases that you can use to test the program to know if you have completely and correctly solved all the bugs.

```
1 //Test2Q1.cpp
2 #include <iostream>
3 using namespace std;
4
5 int OneuntilNine();
6 Display(char, char, char, char, char, char, char, char, char, char);
7 bool WhoWin(char, char, char, char, char, char, char, char, char, char);
8 int Occupied(char, char, char, char, char, char, char, char, char, char);
9
10 int main() {
11     int move;
12     char decision;
13
14     Do {
15         char box1 = ' ', box2 = ' ', box3 = ' ',
16             box4 = ' ', box5 = ' ', box6 = ' ',
17             box7 = ' ', box8 = ' ', box9 = ' ';
18         bool endGame = false;
19         bool Xplayer = false;
20         bool Oplayer = false;
21         int boxTransfer;
22
23         while (endGame != false) {
24             // Player X move
25             Display(box1, box2, box3, box4, box5, box6, box8, box8, box9);
26             cout << "Player X choose box (1-9): ";
27             move = OneUntilNine();
28
29             boxTransfer =
30             Occupied(move, box1, box2, box3, box4, box5, box6, box7, box8, box9);
31             if (boxTransfer == 1) box1 = 'X';
```

```

32     else if (boxTransfer == 2) box2 = 'X';
33     else if (boxTransfer == 3) box3 = 'X';
34     else if (boxTransfer == 4) box4 = 'X';
35     else if (boxTransfer == 5) box5 = 'X';
36     else if (boxTransfer == 6) box6 = 'X';
37     else if (boxTransfer == 7) box7 = 'X';
38     else if (boxTransfer == 8) box8 = 'X';
39     else if (boxTransfer == 9) box9 = 'X';
40
41     Xplayer = WhoWin(box1,box2,box3,box4,box5,box6,box7,box8,box9);
42     if (Xplayer == true) {
43         cout << "\t **PLAYER X WINS!!**" << endl << endl;
44         Display(box1,box2,box3,box4,box5,box6,box7,box8,box9);
45         endGame = true;
46         break;
47     }
48
49     if (box1!=' ' && box2!=' ' && box3!=' ' && box4!=' ' && box5!=' '
50     && box6!=' ' && box7!=' ' && box8!=' ' && box9!=' ') {
51         cout << "\t **DRAW!!**" << endl << endl;
52         Display(box1,box2,box3,box4,box5,box6,box7,box8,box9);
53         endGame = true;
54     }
55
56     // Player O move
57     Display(box1,box2,box3,box4,box5,box6,box7,box8,box9);
58     cout << "Player O choose box (1-9): ";
59     move = OneUntilNine();
60
61     boxTransfer =
62     Occupied(move,box1,box2,box3,box4,box5,box6,box7,box8,box9);
63     if (boxTransfer == 1) box1 = 'O';
64     else if (boxTransfer == 2) box2 = 'O';
65     else if (boxTransfer == 3) box3 = 'O';
66     else if (boxTransfer == 4) box4 = 'O';
67     else if (boxTransfer == 5) box5 = 'O';
68     else if (boxTransfer == 6) box6 = 'O';
69     else if (boxTransfer == 7) box7 = 'O';
70     else if (boxTransfer == 8) box8 = 'O';
71     else if (boxTransfer == 9) box9 = 'O';
72
73     Oplayer = WhoWin(box1,box2,box3,box4,box5,box6,box7,box8,box9);
74     if (Oplayer == true) {
75         cout << "\t **PLAYER O WINS!!**" << endl << endl;
76         Display(box1,box2,box3,box4,box5,box6,box7,box8,box9);
77         endGame = true;
78         break;
79     }
80
81     if (box1!=' ' && box2!=' ' && box3!=' ' && box4!=' ' && box5!=' '
82     && box6!=' ' && box7!=' ' && box8!=' ' && box9!=' ') {
83         cout << "\t **DRAW!!**" << endl << endl;
84         Display(box1,box2,box3,box4,box5,box6,box7,box8,box9);
85         endGame = true;
86         break;
87     }
88 }
89
90 cout << "Do you want to play again?? (Yes for any key and N/n for NO):
91 ";
92 cin >> decision;
93 } while (decision != 'n' || decision != 'N');
94
95 cout << "\t **BYE**" << endl << endl;
96 return 0;
97 }
98
99 int OneUntilNine() {

```

```

100     int input;
101
102     while (!(cin >> input) || input < 1 || input > 9) {
103         cin.clear();
104         cin.ignore(99, '\n');
105         cout << " *Wrong input. Try again: ";
106     }
107     return;
108 }
109
110 void Display(char box1, char box2, char box3, char box4, char box5, char
111 box6, char box7, char box8, char box9) {
112     cout << endl << "Noughts & Crosses";
113     cout << "\t Numbers that represent each box" << endl;
114     cout << " +---+---+---+" << "\t\t\t +---+---+---+" << endl;
115     cout << " | "<<box1<<" | " << box2 << " | "<< box3 << " | " << "\t\t\t |
116 1 | 2 | 3 |" << endl;
117     cout << " +---+---+---+" << "\t\t\t +---+---+---+" << endl;
118     cout << " | "<<box4<<" | " << box5 << " | "<< box6 << " | " << "\t\t\t |
119 4 | 5 | 6 |" << endl;
120     cout << " +---+---+---+" << "\t\t\t +---+---+---+" << endl;
121     cout << " | "<<box7<<" | " << box8 << " | "<< box9 << " | " << "\t\t\t |
122 7 | 8 | 9 |" << endl;
123     cout << " +---+---+---+" << "\t\t\t +---+---+---+" << endl << endl;
124 }
125
126 int Occupied(char move, char box1, char box2, char box3, char box4, char
127 box5, char box6, char box7, char box8, char box9) {
128     bool infinity == true;
129
130     while (infinity == true) {
131         if (move == 1 && box1 == ' ')
132             return 1;
133         else if (move == 1 && box1 != ' ') {
134             cout << "Already used. Choose again: ";
135             move = OneUntilNine();
136         }
137         if (move == 2 && box2 == ' ')
138             return 2;
139         else if (move == 2 && box2 != ' ') {
140             cout << "Already used. Choose again: ";
141             move = OneUntilNine();
142         }
143         if (move == 3 && box3 == ' ')
144             return 3;
145         else if (move == 3 && box3 != ' ') {
146             cout << "Already used. Choose again: ";
147             move = OneUntilNine();
148         }
149         if (move == 4 && box4 == ' ')
150             return 4;
151         else if (move == 4 && box4 != ' ') {
152             cout << "Already used. Choose again: ";
153             move = OneUntilNine();
154         }
155         if (move == 5 && box5 == ' ')
156             return 5;
157         else if (move == 5 && box5 != ' ') {
158             cout << "Already used. Choose again: ";
159             move = OneUntilNine();
160         }
161         if (move == 6 && box6 == ' ')
162             return 6;
163         else if (move == 6 && box6 != ' ') {
164             cout << "Already used. Choose again: ";
165             move = OneUntilNine();
166         }
167         if (move == 7 && box7 == ' ')

```

```

168         return 7;
169     else if(move == 7 && box7 != ' ') {
170         cout >> "Already used. Choose again: ";
171         move = OneUntilNine();
172     }
173     if (move == 8 && box8 == ' ')
174         return 8;
175     else if (move == 8 && box8 != ' ') {
176         cout << "Already used. Choose again: ";
177         move = OneUntilNine();
178     }
179     if (move == 9 && box9 == ' ')
180         return 9;
181     else if (move == 9 && box9 != ' ') {
182         cout << "Already used. Choose again: ";
183         move = OneUntilNine();
184     }
185 }
186 return 'E'; //error!
187 }
188
189 bool WhoWin(char box1,char box2,char box3,char box4,char box5,char box6,char
190 box7,char box8,char box9) {
191     if (box1 != ' ' && box1 == box2 && box1 == box3)        return true;
192     else if (box4 != ' ' && box4 == box5 && box4 == box6) return true;
193     else if (box7 != ' ' && box7 == box8 && box7 == box9) return true;
194     else if (box1 != ' ' && box1 == box4 && box1 == box7) return true;
195     else if (box2 != ' ' && box2 == box5 && box2 == box8) return true;
196     else if (box3 != ' ' && box3 == box6 && box3 == box9) return true;
197     else if (box1 != ' ' && box1 == box5 && box1 == box9) return true;
198     else if (box3 != ' ' && box3 == box5 && box3 == box7) return true;
199     return true;
200 }

```

Table 1: Test cases to run and test the program (user inputs are shown in **bold text)**

TEST CASE 1 (X player wins)	
Noughts & Crosses	Numbers that represent each box
<pre> +-----+ +-----+ +-----+ +-----+ +-----+ </pre>	<pre> +-----+ 1 2 3 +-----+ 4 5 6 +-----+ 7 8 9 +-----+ </pre>
Player X choose box (1-9): 1	
Noughts & Crosses	Numbers that represent each box
<pre> +-----+ X +-----+ +-----+ +-----+ +-----+ </pre>	<pre> +-----+ 1 2 3 +-----+ 4 5 6 +-----+ 7 8 9 +-----+ </pre>
Player O choose box (1-9): 2	
Noughts & Crosses	Numbers that represent each box
<pre> +-----+ X O +-----+ +-----+ +-----+ +-----+ </pre>	<pre> +-----+ 1 2 3 +-----+ 4 5 6 +-----+ 7 8 9 +-----+ </pre>
Player X choose box (1-9): 3	
Noughts & Crosses	Numbers that represent each box

<pre>+---+---+---+ X O X +---+---+---+ +---+---+---+ +---+---+---+</pre>		<pre>+---+---+---+ 1 2 3 +---+---+---+ 4 5 6 +---+---+---+ 7 8 9 +---+---+---+</pre>	
Player O choose box (1-9): 3			
Already used. Choose again: 5			
Noughts & Crosses		Numbers that represent each box	
<pre>+---+---+---+ X O X +---+---+---+ O +---+---+---+ +---+---+---+</pre>		<pre>+---+---+---+ 1 2 3 +---+---+---+ 4 5 6 +---+---+---+ 7 8 9 +---+---+---+</pre>	
Player X choose box (1-9): 4			
Noughts & Crosses		Numbers that represent each box	
<pre>+---+---+---+ X O X +---+---+---+ X O +---+---+---+ +---+---+---+</pre>		<pre>+---+---+---+ 1 2 3 +---+---+---+ 4 5 6 +---+---+---+ 7 8 9 +---+---+---+</pre>	
Player O choose box (1-9): 6			
Noughts & Crosses		Numbers that represent each box	
<pre>+---+---+---+ X O X +---+---+---+ X O O +---+---+---+ +---+---+---+</pre>		<pre>+---+---+---+ 1 2 3 +---+---+---+ 4 5 6 +---+---+---+ 7 8 9 +---+---+---+</pre>	
Player X choose box (1-9): 7			
PLAYER X WINS!!			
Noughts & Crosses		Numbers that represent each box	
<pre>+---+---+---+ X O X +---+---+---+ X O O +---+---+---+ X +---+---+---+</pre>		<pre>+---+---+---+ 1 2 3 +---+---+---+ 4 5 6 +---+---+---+ 7 8 9 +---+---+---+</pre>	
Do you want to play again?? (Yes for any key and N/n for NO): N			
BYE			
TEST CASE 2 (O player wins)			
Noughts & Crosses		Numbers that represent each box	
<pre>+---+---+---+ +---+---+---+ +---+---+---+ +---+---+---+</pre>		<pre>+---+---+---+ 1 2 3 +---+---+---+ 4 5 6 +---+---+---+ 7 8 9 +---+---+---+</pre>	
Player X choose box (1-9): 1			
Noughts & Crosses		Numbers that represent each box	
<pre>+---+---+---+</pre>		<pre>+---+---+---+</pre>	

<pre> X +---+---+---+ +---+---+---+ +---+---+---+ </pre>	<pre> 1 2 3 +---+---+---+ 4 5 6 +---+---+---+ 7 8 9 +---+---+---+ </pre>
<p>Player O choose box (1-9): 2</p>	
<p>Noughts & Crosses</p> <pre> +---+---+---+ X O +---+---+---+ +---+---+---+ +---+---+---+ </pre>	<p>Numbers that represent each box</p> <pre> +---+---+---+ 1 2 3 +---+---+---+ 4 5 6 +---+---+---+ 7 8 9 +---+---+---+ </pre>
<p>Player X choose box (1-9): 3</p>	
<p>Noughts & Crosses</p> <pre> +---+---+---+ X O X +---+---+---+ +---+---+---+ +---+---+---+ </pre>	<p>Numbers that represent each box</p> <pre> +---+---+---+ 1 2 3 +---+---+---+ 4 5 6 +---+---+---+ 7 8 9 +---+---+---+ </pre>
<p>Player O choose box (1-9): 5</p>	
<p>Noughts & Crosses</p> <pre> +---+---+---+ X O X +---+---+---+ O +---+---+---+ +---+---+---+ </pre>	<p>Numbers that represent each box</p> <pre> +---+---+---+ 1 2 3 +---+---+---+ 4 5 6 +---+---+---+ 7 8 9 +---+---+---+ </pre>
<p>Player X choose box (1-9): 7</p>	
<p>Noughts & Crosses</p> <pre> +---+---+---+ X O X +---+---+---+ O +---+---+---+ X +---+---+---+ </pre>	<p>Numbers that represent each box</p> <pre> +---+---+---+ 1 2 3 +---+---+---+ 4 5 6 +---+---+---+ 7 8 9 +---+---+---+ </pre>
<p>Player O choose box (1-9): 8</p> <p style="text-align: center;">**PLAYER O WINS!!**</p>	
<p>Noughts & Crosses</p> <pre> +---+---+---+ X O X +---+---+---+ O +---+---+---+ X O +---+---+---+ </pre>	<p>Numbers that represent each box</p> <pre> +---+---+---+ 1 2 3 +---+---+---+ 4 5 6 +---+---+---+ 7 8 9 +---+---+---+ </pre>
<p>Do you want to play again?? (Yes for any key and N/n for NO): Y</p>	
<p>Noughts & Crosses</p> <pre> +---+---+---+ +---+---+---+ +---+---+---+ </pre>	<p>Numbers that represent each box</p> <pre> +---+---+---+ 1 2 3 +---+---+---+ 4 5 6 +---+---+---+ </pre>

				7 8 9			
+---+---+---+				+---+---+---+			
Player X choose box (1-9):							
TEST CASE 3 (Draw)							
Noughts & Crosses				Numbers that represent each box			
+---+---+---+				+---+---+---+			
				1 2 3			
+---+---+---+				+---+---+---+			
				4 5 6			
+---+---+---+				+---+---+---+			
				7 8 9			
+---+---+---+				+---+---+---+			
Player X choose box (1-9): 1							
Noughts & Crosses				Numbers that represent each box			
+---+---+---+				+---+---+---+			
X				1 2 3			
+---+---+---+				+---+---+---+			
				4 5 6			
+---+---+---+				+---+---+---+			
				7 8 9			
+---+---+---+				+---+---+---+			
Player O choose box (1-9): 2							
Noughts & Crosses				Numbers that represent each box			
+---+---+---+				+---+---+---+			
X O				1 2 3			
+---+---+---+				+---+---+---+			
				4 5 6			
+---+---+---+				+---+---+---+			
				7 8 9			
+---+---+---+				+---+---+---+			
Player X choose box (1-9): 3							
Noughts & Crosses				Numbers that represent each box			
+---+---+---+				+---+---+---+			
X O X				1 2 3			
+---+---+---+				+---+---+---+			
				4 5 6			
+---+---+---+				+---+---+---+			
				7 8 9			
+---+---+---+				+---+---+---+			
Player O choose box (1-9): 4							
Noughts & Crosses				Numbers that represent each box			
+---+---+---+				+---+---+---+			
X O X				1 2 3			
+---+---+---+				+---+---+---+			
O				4 5 6			
+---+---+---+				+---+---+---+			
				7 8 9			
+---+---+---+				+---+---+---+			
Player X choose box (1-9): 7							
Noughts & Crosses				Numbers that represent each box			
+---+---+---+				+---+---+---+			
X O X				1 2 3			
+---+---+---+				+---+---+---+			
O				4 5 6			
+---+---+---+				+---+---+---+			
X				7 8 9			
+---+---+---+				+---+---+---+			
Player O choose box (1-9): 5							
Noughts & Crosses				Numbers that represent each box			


```

+---+---+---+
| X | O | X |
+---+---+---+
| O | O |   |
+---+---+---+
| X |   |   |
+---+---+---+

```

```

+---+---+---+
| 1 | 2 | 3 |
+---+---+---+
| 4 | 5 | 6 |
+---+---+---+
| 7 | 8 | 9 |
+---+---+---+

```

Player X choose box (1-9): **6**

Noughts & Crosses

Numbers that represent each box

```

+---+---+---+
| X | O | X |
+---+---+---+
| O | O | X |
+---+---+---+
| X |   |   |
+---+---+---+

```

```

+---+---+---+
| 1 | 2 | 3 |
+---+---+---+
| 4 | 5 | 6 |
+---+---+---+
| 7 | 8 | 9 |
+---+---+---+

```

Player O choose box (1-9): **9**

Noughts & Crosses

Numbers that represent each box

```

+---+---+---+
| X | O | X |
+---+---+---+
| O | O | X |
+---+---+---+
| X |   | O |
+---+---+---+

```

```

+---+---+---+
| 1 | 2 | 3 |
+---+---+---+
| 4 | 5 | 6 |
+---+---+---+
| 7 | 8 | 9 |
+---+---+---+

```

Player X choose box (1-9): **8**

****DRAW!!****

Noughts & Crosses

Numbers that represent each box

```

+---+---+---+
| X | O | X |
+---+---+---+
| O | O | X |
+---+---+---+
| X | X | O |
+---+---+---+

```

```

+---+---+---+
| 1 | 2 | 3 |
+---+---+---+
| 4 | 5 | 6 |
+---+---+---+
| 7 | 8 | 9 |
+---+---+---+

```

Do you want to play again?? (Yes for any key and N/n for NO): **n**

****BYE****

QUESTION 2 – PROBLEM SOLVING

(65 Marks)

Write a complete C++ program that act as a simple Academic Management System. This program should be used by PT1 lecturers. It will prompt the lecturer to enter all student IDs, coursework, and final exam marks. Please write the program using the given template file (**Test2Q2.cpp**). Your program should be able to do the following tasks:

Task 1: Define a function named **getInput**. (6 marks)

- a) The purpose of this function is to get data from the keyboard for the n^{th} student.
- b) It accepts an array of student IDs and an array of marks as input parameters.
- c) The user needs to enter the student ID, coursework, and final exam marks into the array received in (b) using the appropriate index.
- d) The function must then update the number of students, every time it is called.
Hint: Static variable can be used for this purpose.
- e) The function should return the updated number of students calculated in (d).

Task 2: Define a function named **calcTotal**. (4 marks)

- a) The purpose of this function is to compute the total mark for the n^{th} student.
- b) It takes as input parameters an array of marks and an array index to represent the n^{th} student.
- c) Calculate the total mark for the n^{th} student by adding the coursework and final exam marks.
- d) This function should return the calculation result in (c).

Task 3: Define a function named **deterGrade**. (8 marks)

- a) The purpose of this function is to determine the grade for the n^{th} student.
- b) It takes as input parameters an array of marks and an array index to represent the n^{th} student.
- c) Determine the grade for the n^{th} student using the total mark and the criteria listed in **Table 2**.
- d) This function should return the grade determined in (c).

Table 2: Grading criteria

Mark Range	Grade
80 and above	A
$60 \leq \text{Mark} < 80$	B
$45 \leq \text{Mark} < 60$	C
$30 \leq \text{Mark} < 45$	D
Below 30	E

Task 4: Define a function named **dispLine**. (2.5 marks)

- a) The purpose of this function is to display a line using the 42 characters of '-'.
- b) This is a non-returning function.

- c) The function should display the line using a loop.

Task 5: Define a function named **dispOutput**. (19 marks)

- a) The purpose of this function is to display output as shown in **Figure 1**.
- b) This is a non-returning function.
- c) It accepts as input parameters an array of student IDs, an array of marks, an array of grades, and the number of students.
- d) Determine the highest and lowest marks.
- e) Then, calculate the average of total marks.
- f) This function should display the student ID, coursework, final exam, total marks and grades of all students, the highest and lowest marks determined in (d), and the average total mark calculated in (e). You should invoke the **dispLine** function. **Figure 1** shows an example of program output.

Task 6: Define a **main** function to perform the following tasks: (10.5 marks)

- a) Define three arrays:
 - A one-dimensional or two-dimensional array used to store a list of student IDs.
 - A two-dimensional array with three columns for storing a list of student marks. The first column contains the list of coursework marks, the second column contains the list of final exam marks, and the third column contains the total marks.
 - A one-dimensional array for storing a list of student grades.
- b) Use an appropriate **LOOP** to execute the process in this function. The loop will be repeated when the user press 'Y' or 'y'.
- c) The function must call the functions defined in Tasks 1 until Task 3 and Task 5 to produce the output shown in **Figure 1**. The values in **bold** in **Figure 1** are the user inputs.

Task 7: You must ensure that your program meet the following criteria: (15 marks)

- a) The program is able to run and generate a correct output. (5 marks)
- b) Using an appropriate structure for the program:
 - Applying proper styles, e.g. indentation and comments (2 marks)
 - All required header files are included. (1 mark)
 - All required variables are declared and properly initialized. (5 marks)
 - The function **main** is properly written. (2 marks)

```

Matrics No.: A22CS4007
Coursework (max. 65): 25.1
Final Exam (max. 35): 3.6

Press 'Y' or 'y' to continue: Y

Matrics No.: A22CS0157
Coursework (max. 65): 63.6
Final Exam (max. 35): 25.9

Press 'Y' or 'y' to continue: y

Matrics No.: A22CS0104
Coursework (max. 65): 47.5
Final Exam (max. 35): 15.3

Press 'Y' or 'y' to continue: Y

Matrics No.: A22CS4005
Coursework (max. 65): 39.7
Final Exam (max. 35): 2.4

Press 'Y' or 'y' to continue: y

Matrics No.: A22CS0079
Coursework (max. 65): 43.3
Final Exam (max. 35): 11.6

Press 'Y' or 'y' to continue: Y

Matrics No.: A22CS0084
Coursework (max. 65): 59.1
Final Exam (max. 35): 22.2

Press 'Y' or 'y' to continue: y

Matrics No.: A22CS0137
Coursework (max. 65): 46.3
Final Exam (max. 35): 12.2

Press 'Y' or 'y' to continue: n

-----
Results of SECJ1013 for 7 students
-----
1)  A22CS4007  25.1    3.6    28.7    E
2)  A22CS0157  63.6    25.9    89.5    A
3)  A22CS0104  47.5    15.3    62.8    B
4)  A22CS4005  39.7     2.4    42.1    D
5)  A22CS0079  43.3    11.6    54.9    C
6)  A22CS0084  59.1    22.2    81.3    A
7)  A22CS0137  46.3    12.2    58.5    C
-----
Highest mark : 89.5 (A22CS0157)
Lowest mark  : 28.7 (A22CS4007)
Average mark : 59.6857
-----

```

Figure 1: Example output of the program