

State the name of an identifier for 02.1 to 02.5:

1

02.1 A one dimensional Array

1

02.2 A user-defined subroutine that has five parameters

1

02.3 A variable used to store a Whole Number

1

02.4 A variable used to store a single character

1

02.5 A user-defined subroutine that returns one or more values

02.6 Loop at the repetition structured in the SetUpBoard subroutine

3

Explain the need for a nested FOR loop and the role of the Row and Column variables.

2

02.7 Why has a named constant been used for the TrainingGame?

3

02.8 Describe the benefit of using the Structure Ship instead of just relying on two separate variables

2

02.9 Describe how and why a two-dimensional array is used for the Board

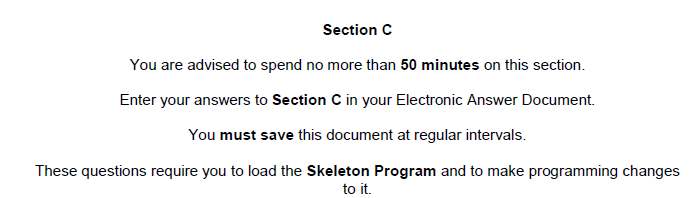
2

02.10 State two reasons why subroutines should, ideally, not use global variables.

02.11 Explain why a For Loop is used over a different loop structure in many Subroutines in the program

2

**Total: 19**



03 This question refers to the subroutine GetMainMenuChoice

The **Skeleton Program** currently has no validation for the menu choice. It should **not** be possible to choose an option that is outside the range 1 to 9 inclusively. The user also should **not** be able to enter anything other than an Integer value for the menu choice.

If the user enters an invalid menu option the program must display the message:

“Incorrect menu option – please try again”

03.1 Include your amended PROGRAM SOURCE CODE for the subroutine GetMainMenuChoice (6)

03.2 SCREEN CAPTURE(S) for a test showing the following values: (4)

0

A

10

1

04 This question refers to the subroutine GetRowColumn

The **Skeleton Program** currently has no validation for the co-ordinates. It should **not** be possible to choose an option that is outside the range 0 to 9 inclusively for both row and column.

If the user enters an invalid co-ordinate the program will ask for the co-ordinate to be entered again.

“Invalid Co-ordinate – please re-enter”

04.1 Include your amended PROGRAM SOURCE CODE for the subroutine GetRowColumn (4)

04.2 SCREEN CAPTURE(S) for a test showing the following values: (3)

Column: -1

Column: 10

Column: 0

Row: -1

Row: 10

Row: 9

05 When starting a new game – the ships are placed in the same positions every time.

What is missing from the code that would ensure that a new game starts with the ships all being placed in jumbled positions? (1)

06 This question refers to the subroutine DisplayMenu

Currently there are just three options:

1. Start new game

2. Load training game

9. Quit

Add a further four options:

3. Save a game

4. Set-Up Ships Manually

5. Display High-Score Table

6. Display Instructions

Write a new subroutine called DisplayInstructions. This subroutine should display the following information to the user:

The aim of the game is to destroy all the ships that have been hidden in a 10 x 10 grid called the board.

The player fires a shot by specifying the location of a square on the board. To specify the location of a square the player enters a column number (between 0 and 9) and then a row number (between 0 and 9).

<press enter to continue>

Alter the subroutine called Main to ensure that DisplayInstructions is called with the correct menu option chosen.

06.1 Include your amended PROGRAM SOURCE CODE for the subroutine DisplayMenu (2)

06.2 Include your amended PROGRAM SOURCE CODE for the subroutine DisplayInstructions (5)

06.3 Include your amended PROGRAM SOURCE CODE for the subroutine Main (3)

06.4 SCREEN CAPTURE(S) for a test showing the menu and option 6 chosen (3)

07 This question refers to the subroutine PlayGame and MakePlayerMove

At the moment, the player can continue to play the game until the board is full. The program should keep track of the number of hits that the player has made and display these, therefore the player can try to beat their personal best.

Study the subroutine PlayGame

Add a variable with the identifier NumberOfShots – this will have to be sent to the MakePlayerMove

Study the subroutine MakePlayerMove

The variable NumberOfShots should be brought in as a parameter and incremented when a correct shot has been fired (e.g. one that causes a hit or a miss – not one that overwrites a square already shot at).

The routine will also display a message:

You have made X shots so far.

Alter the subroutine PlayGame so that if 30 moves are made the game is over – a message should be displayed “No more shots left – Game Over”

07.1 Include your amended PROGRAM SOURCE CODE for the subroutine PlayGame (5)

07.2 Include your amended PROGRAM SOURCE CODE for the subroutine MakePlayerMove (5)

07.3 SCREEN CAPTURE(S) for a test showing four moves being made: (4)

7,7

6,7

7,7

1,1

07.4 SCREEN CAPTURE(S) for a test showing 30 moves being made and the message “No more shots” (1)

Total: Section B/C (out of 65)