

Section C

You are advised to spend no more than **50 minutes** on this section.

Enter your answers to **Section C** in your Electronic Answer Document. You **must save** this document at regular intervals.

These questions require you to load the **Skeleton Program** and to make programming changes to it.

0 9

This question refers to the subroutine `GetRowColumn`.

What you need to do:

Task 1

Adapt the program source code for the subroutine `GetRowColumn` so that it checks that the **row** entered by the player is in the allowed range.

If an invalid value is entered for the row the program should output:

`Invalid value entered`

The subroutine should not return any values until a valid row has been entered.

Task 2

Test that the changes you have made work by conducting the following test:

- run the **Skeleton Program**
- select option 2 from the menu
- fire a shot at column 6, row 10
- fire a shot at column 6, row 9.

Evidence that you need to provide

Include the following evidence in your Electronic Answer Document.

0 9

. 1

Your amended PROGRAM SOURCE CODE for the subroutine `GetRowColumn`.

[5 marks]

0 9

. 2

SCREEN CAPTURE(S) showing the requested test.

[1 mark]

1	0
---	---

This question will extend the functionality of the game.

The game needs to be changed so that a message is displayed when all the squares occupied by a ship have been hit.

Figure 6 shows an example of this functionality of the game using the board from the training game.

Figure 6

The player has made three shots and achieved 2 hits and a miss.

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9					m	h	h			

The player's next shot into column 7, row 9 is a third hit and sinks the ship.
The game displays the message 'Destroyer is sunk!' to the player.

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9					m	h	h	h		

Destroyer is sunk!

What you need to do:**Task 1**

Create a new subroutine `CheckSunk` that, when given the current shot location, board and list of ships, will:

- check the board to find out which kind of ship has been hit
- reduce, by one, the number of squares occupied by that ship in the list of ships
- display the message 'XX is sunk!' where XX is the type of the ship, if the ship now occupies zero squares.

Task 2

Modify the `MakePlayerMove` subroutine so that the subroutine `CheckSunk` is called when appropriate.

Task 3

Test your program works by conducting the following test:

- run the **Skeleton Program**
- select option 2 from the menu
- fire a shot at column 1, row 3
- fire a shot at column 1, row 4
- fire a shot at column 1, row 5.

Evidence that you need to provide

Include the following evidence in your Electronic Answer Document.

1	0	.	1	Your PROGRAM SOURCE CODE for the subroutine <code>CheckSunk</code> .	[8 marks]
1	0	.	2	Your amended PROGRAM SOURCE CODE for the subroutine <code>MakePlayerMove</code> .	[1 mark]
1	0	.	3	SCREEN CAPTURE(S) showing the final board layout and the message 'Patrol Boat is sunk!'.	[1 mark]

1 1

This question will further extend the functionality of the game.

The game is to be altered so that a player has the option of firing either a standard shot or a torpedo.

The player can only fire a torpedo once during the game.

If a player decides to fire a torpedo it behaves in the following manner:

- if the torpedo lands directly on a ship (even if this part of the ship has already been hit) it behaves like a standard shot and explodes resulting in the square being marked as a hit and an appropriate message is displayed
- if the torpedo lands in an empty square or a square already marked as a miss then the torpedo keeps moving up one square until it either hits a ship or moves off the board
- each square that the torpedo has moved from will be marked as a miss
- if the torpedo moves into a square that is occupied by a ship this will cause a hit like a standard shot (even if this part of the ship has already been hit), the torpedo will be removed from the board and an appropriate message is displayed
- if the torpedo moves off the board it will disappear and an appropriate message is displayed.

You can choose to display the board each time the torpedo moves or to display the board only after the torpedo has finished moving.

Figures 7 to 9 show an example of a torpedo being used.

Figure 7

A torpedo is fired at column 8, row 6 and does not hit a ship.

At this stage no change is made to the contents of the board as the torpedo did not achieve a direct hit.

	0	1	2	3	4	5	6	7	8	9
0									m	
1									h	
2									h	
3										
4										
5										
6										
7										
8										
9										

Figure 8

The torpedo moves up one square but does not hit anything.
Column 8, row 6 (the square the torpedo came from) is marked as a miss.

	0	1	2	3	4	5	6	7	8	9
0									m	
1									h	
2									h	
3										
4										
5										
6									m	
7										
8										
9										

Figure 9

The torpedo moves up one square and hits a ship.
The torpedo is removed from the board.
Column 8, row 5 (the square the torpedo came from) is marked as a miss.
Column 8, row 4 is marked as a hit.

	0	1	2	3	4	5	6	7	8	9
0									m	
1									h	
2									h	
3										
4									h	
5									m	
6									m	
7										
8										
9										

Question 11 continues on the next page

What you need to do:**Task 1**

Create a new subroutine `MakePlayerTorpedoMove` that when given a current board:

- asks the player to enter the row and column
- fires a torpedo starting at the square indicated by the player.

The torpedo should behave as described in **points a–e** on **page 16**.

Task 2

Adapt the `PlayGame` subroutine so that, if the player has not already fired a torpedo during the game, the player is asked:

Fire a torpedo? (Y/N)

If the player responds 'N' a standard shot should be fired.

If the player responds 'Y' the subroutine `MakePlayerTorpedoMove` should be called.

Task 3

Test your program works by conducting the following test:

- run the **Skeleton Program**
- select option 2 from the menu
- fire a torpedo at column 1, row 7
- fire a standard shot at column 2, row 1.

Evidence that you need to provide

Include the following evidence in your Electronic Answer Document.

1 1 . 1 Your amended PROGRAM SOURCE CODE for the subroutines `MakePlayerTorpedoMove` and `PlayGame` and any other code that you have changed or added. **[12 marks]**

1 1 . 2 SCREEN CAPTURE(S) showing the requested test. **[2 marks]**

END OF QUESTIONS