AS
COMPUTER SCIENCE

Paper 1

Tuesday 13 May 2025 Afternoon Time allowed: 1 hour 45 minutes

Materials

For this paper you must have:

- a computer
- a printer
- appropriate software
- the Electronic Answer Document
- an electronic version and a hard copy of the Skeleton Program
- an electronic version and a hard copy of the Preliminary Material
- an electronic version of the Data Files **MapData.txt** and **HiddenData.txt**

You must **not** use a calculator.

Instructions

- Type the information required on the front of your Electronic Answer Document.
- Before the start of the examination make sure your **Centre Number**, **Candidate Name** and **Candidate Number** are shown clearly **in the footer** of every page (not the front cover) of your Electronic Answer Document.
- Enter your answers into the Electronic Answer Document.
- Answer **all** questions.
- Save your work at regular intervals.

Information

- The marks for questions are shown in brackets.
- The maximum mark for this paper is 75.
- No extra time is allowed for printing and collating.
- The Question Paper is divided into **three** sections.

Advice

You are advised to allocate time to each section as follows:
Section A – 20 minutes; **Section B** – 25 minutes; **Section C** – 60 minutes.

At the end of the examination

Tie together all your printed Electronic Answer Document pages and hand them to the Invigilator.

Warning

It may not be possible to issue a result for this paper if your details are not on every page of your Electronic Answer Document.

Section A

You are advised to spend no more than **20 minutes** on this section.

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

Question **03** in this section asks you to write program code **starting from a new program/project/file**.

You are advised to save your program at regular intervals.

0 1 **Figure 1** shows the data structures `Letter`, `L` and `R` used by the algorithm shown in **Figure 2**.

Figure 1

	Letter
[0]	
[1]	A
[2]	B
[3]	C
[4]	D
[5]	E
[6]	F
[7]	G
[8]	H
[9]	I
[10]	J
[11]	K
[12]	L
[13]	M
[14]	N
[15]	O
[16]	P
[17]	Q
[18]	R
[19]	S
[20]	T
[21]	U
[22]	V
[23]	W
[24]	X
[25]	Y
[26]	Z

	L
[0]	5
[1]	18
[2]	-1
[3]	-1
[4]	2
[5]	9
[6]	-1
[7]	26
[8]	-1
[9]	19
[10]	-1
[11]	3
[12]	-1
[13]	7
[14]	4
[15]	-1
[16]	-1
[17]	-1
[18]	12
[19]	8
[20]	14
[21]	6
[22]	-1
[23]	16
[24]	-1
[25]	-1
[26]	-1

	R
[0]	20
[1]	23
[2]	-1
[3]	-1
[4]	24
[5]	1
[6]	-1
[7]	17
[8]	-1
[9]	21
[10]	-1
[11]	25
[12]	-1
[13]	15
[14]	11
[15]	-1
[16]	-1
[17]	-1
[18]	-1
[19]	22
[20]	13
[21]	-1
[22]	-1
[23]	10
[24]	-1
[25]	-1
[26]	-1

Figure 2

```
M ← "1001"
Current ← 0
FOR i ← 0 TO 3
  Symbol ← M[i]
  IF Symbol = "0" THEN
    Current ← L[Current]
  ELSE
    Current ← R[Current]
  ENDIF
ENDFOR
OUTPUT Letter[Current]
```

Complete **Table 1** by hand-tracing the algorithm in **Figure 2**.

The strings are zero index based. For example, the character with index 0 in the string "ABCD" is "A".

You may not need to use all the rows in **Table 1**.

The first row of **Table 1** has already been completed for you.

Table 1

M	i	Symbol	Current	Output
"1001"			0	

Copy the contents of all the unshaded cells in **Table 1** into your Electronic Answer Document.

[3 marks]

Turn over for the next question

0 2 **Figure 3** and **Figure 4** show two blocks of pseudo-code.

Figure 3

```
INPUT X
WHILE X > 0
  X ← X - 1
ENDWHILE
```

Figure 4

```
INPUT X
REPEAT
  X ← X - 1
UNTIL X ≤ 0
```

Explain why the pseudo-code in **Figure 3** operates differently to the pseudo-code in **Figure 4** when the inputs are negative numbers.

[2 marks]

0 3 **Figure 5** shows an algorithm represented using pseudo-code.

Figure 5

```
S ← ""
WHILE S ≠ "x"
  OUTPUT "Enter a word or phrase: "
  INPUT S
  Max ← LENGTH(S) - 1
  Matched ← True
  FOR i ← 0 TO Max
    Letter1 ← S[i]
    Letter2 ← S[Max - i]
    IF Letter1 ≠ Letter2 THEN
      Matched ← False
    ENDIF
  ENDFOR
  IF Matched = True THEN
    OUTPUT "Palindrome"
  ELSE
    OUTPUT "Not a palindrome"
  ENDIF
ENDWHILE
```

The strings are zero index based. For example, the character with index 0 in the string "ABCD" is "A".

What you need to do:

Task 1
Write a program to implement the algorithm in **Figure 5**.

Task 2
Test that your program works:
- run your program
- enter `madam`
- enter `maam`
- enter `adam`
- enter `aam`
- enter `x`

Evidence that you need to provide
Include the following evidence in your Electronic Answer Document.

0 3 . 1 Your PROGRAM SOURCE CODE for **Task 1**.

[8 marks]

0 3 . 2 SCREEN CAPTURE(S) showing the test described in **Task 2**.

[1 mark]

0 3 . 3 State **one** reason why the algorithm in **Figure 5** is not efficient.

[1 mark]

0 4 . 1 Define the term subroutine.

[1 mark]

0 4 . 2 State **three** advantages of using subroutines.

[3 marks]

0 4 . 3 Describe how parameters improve the effectiveness of subroutines.

[2 marks]

0 5 Explain what is meant by data abstraction.

[1 mark]

Turn over for the next section

Section B

You are advised to spend no more than **25 minutes** on this section.

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

These questions refer to the **Preliminary Material** and the **Skeleton Program**, but do **not** require any additional programming.

Refer **either** to the **Preliminary Material** issued with this Question Paper **or** your electronic copy.

0 6 State the identifier of a variable in the **Skeleton Program** that is used to hold one of only two possible values.

[1 mark]

0 7 State the identifier of a subroutine in the **Skeleton Program** that uses nested indefinite iteration.

[1 mark]

0 8 . 1 State the identifier of a subroutine in the **Skeleton Program** that uses exception handling.

[1 mark]

0 8 . 2 Explain why exception handling is used in this subroutine.

[2 marks]

0 9 The **Skeleton Program** uses a number of data structures.

0 9 . 1 State the identifier of a data structure that stores values of more than one data type.

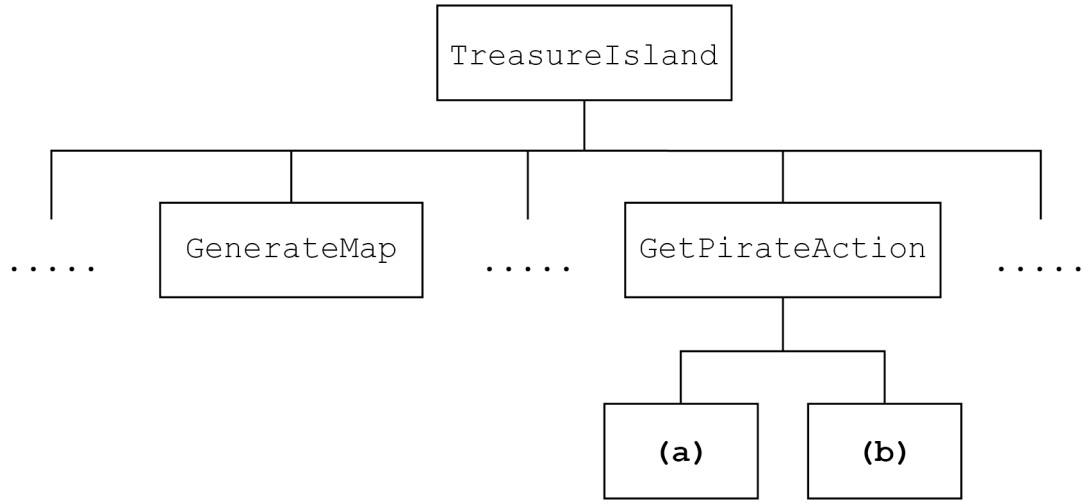
[1 mark]

0 9 . 2 State the identifier of a data structure that stores values of only one **built-in** data type.

[1 mark]

1 0 **Figure 6** shows an incomplete hierarchy chart for part of the **Skeleton Program**.

Figure 6



1 0 . 1 What does a box in a hierarchy chart represent?

[1 mark]

1 0 . 2 What should be written in box **(a)** in **Figure 6**?

[1 mark]

1 0 . 3 What should be written in box **(b)** in **Figure 6**?

[1 mark]

1 0 . 4 State the stage of software development when a hierarchy chart is normally developed.

[1 mark]

Turn over for the next question

1 1 This question refers to the subroutine `Move`.

Explain why it is necessary to check that the value stored at the pirate's position is equal to the value represented by `PIRATES` **before** changing the location to represent `SAND`.

[2 marks]

1 2 This question refers to the subroutine `CheckPath`.

Describe **one similarity** and **two differences** between the FOR loop used when the direction is `"E"` and the FOR loop used when the direction is `"W"`.

In your response you should refer only to the FOR loops, **not** the code inside the FOR loop structures.

[3 marks]

1 3 This question refers to the subroutine `PirateWalks`.

Explain why it is sometimes necessary to set `ValidDirection` to false after it has been set to true by the call to the `CheckDirection` subroutine.

[3 marks]

1 4 There are **three** conditions that cause the game to end.

One condition is that the score goes below 0.

1 4 . 1 Describe the other **two** conditions that cause the game to end.

[2 marks]

The score can increase and decrease during the game.

1 4 . 2 Give **one** pirate action that **increases** the score.

[1 mark]

1 4 . 3 Give **one** pirate action that **decreases** the score.

[1 mark]

Turn over for the next section

Section C

You are advised to spend no more than **60 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.

1 5 This question extends the functionality of the **Skeleton Program**.

The **Skeleton Program** is to be changed so that the time the pirate spends walking is recorded and displayed at the end of the game.

The subroutine `PirateWalks` needs to be modified so that a total of the time the pirate walks is kept:

- walking one length N, E, S or W counts as 1.0 hours walking
- walking one length in a diagonal direction (NE, SE, SW, NW) counts as 1.4 hours walking.

At the end of the game, the total hours walked is to be output as part of the results.

What you need to do:

Task 1
In `PirateRecord` create an extra field `WalkTime` to store the number of hours walked as a number with a fractional part value.

Initialise the value of `WalkTime` to `0.0` in the `ResetPirateRecord` subroutine.

Task 2
Amend the subroutine `PirateWalks` to calculate the time walked and update the `WalkTime` field in the `Pirate` variable.

Task 3
Amend the subroutine `DisplayResults` to output the total time spent walking with an appropriate message.

Task 4
Test that the changes you have made work by conducting the following test:
- run your amended **Skeleton Program**
- enter `W`
- enter `8NE`
- enter `W`
- enter `9E`
- press `ENTER`

Evidence that you need to provide
Include the following evidence in your Electronic Answer Document.

1 5 . 1 Your PROGRAM SOURCE CODE:

- for the entire subroutine `ResetPirateRecord`
- for the entire subroutine `PirateWalks`
- for the entire subroutine `DisplayResults`.

[4 marks]

1 5 . 2 SCREEN CAPTURE(S) showing the requested test described in **Task 4**.

The SCREEN CAPTURE(S) must show the map with the pirate's final position as well as the results.

[1 mark]

Turn over for the next question

1 6 The **Skeleton Program** is to be extended so that the pirate can come ashore at any location that is a beach. The user will be asked to select the landing place.

What you need to do:

Task 1
Amend the subroutine `FindLandingPlace` so it asks the user whether they want to use a different landing place to the one marked by `X` on the map.

If the user chooses to change the landing place, they should be asked for a row and a column for the new landing place.

The subroutine should then check whether the given square is part of a beach.
A beach square is sand and has a square of water on the N, E, S or W side.

- If the given square is a beach square:
  - the pirate's position should be set to this square
  - the pirate (`P`) should be placed on the map
  - the map should be displayed.

- If the given square is **not** a beach square:
  - the pirate will use the landing place given on the map (marked by `X`)
  - a suitable message should be output.

You should assume that the user enters a row and column number that are not at the very edge of the map, ie the row number is not 0 or 19 and the column number is not 0 or 41 for the map in the data file.

Task 2
Test that the changes you have made work by conducting the following test:
- run your amended **Skeleton Program**
- choose a different landing place
- enter `6` for the row
- enter `1` for the column

Task 3
Test that the changes you have made work by conducting the following test:
- run your amended **Skeleton Program**
- choose a different landing place
- enter `17` for the row
- enter `21` for the column
- enter `W`
- enter `1N`

Evidence that you need to provide
Include the following evidence in your Electronic Answer Document.

16.1 Your PROGRAM SOURCE CODE for the entire subroutine `FindLandingPlace`.

[10 marks]

16.2 SCREEN CAPTURE(S) showing the requested test described in **Task 2**.

The SCREEN CAPTURE(S) must show the user input and the map showing where the pirate lands.

[1 mark]

16.3 SCREEN CAPTURE(S) showing the requested test described in **Task 3**.

The SCREEN CAPTURE(S) must show the user input and the map showing the pirate's position after walking.

[1 mark]

Turn over for the next question

1 7 This question extends the functionality of the **Skeleton Program**. If the pirate stands one square to the west of the hut (`H`), a clue about where the treasure is buried will be displayed.

What you need to do:

Task 1
Write a new subroutine `WestOfHut` that checks the pirate's position.

If the pirate is one square to the west of the hut the subroutine should return true; otherwise it should return false.

Your solution should work for any map, not just the map provided in the data files.

Task 2
Write a new subroutine `GetClue` that finds the position of the rock (`R`) on the map and the position of the treasure (`T`) on the hidden map. The distances (number of squares) between the rock and the treasure are to be calculated in the north-south direction and the east-west direction. The subroutine should display these distances with suitable messages.

For example, in **Figure 7**, the distance between the rock and the treasure is 1 square in the north-south direction and 3 squares in the east-west direction.

Figure 7

```
..............
..............
......T.......
...R..........
..............
```

Your solution should work for any map, not just the map provided in the data files. A map always has one rock and one treasure.

Task 3
Amend the subroutine `PirateWalks` so that it calls the `WestOfHut` subroutine after the pirate has finished walking to check whether the pirate is one square to the west of the hut and, if so, calls the `GetClue` subroutine.

Task 4
Test that the changes you have made work by conducting the following test:
- run your amended **Skeleton Program**
- if you have answered Question **16**, select for the pirate to land where the `X` is located on the map.
- enter `W`
- enter `9NE`
- enter `W`
- enter `6E`

Evidence that you need to provide
Include the following evidence in your Electronic Answer Document.

1 7 . 1 Your PROGRAM SOURCE CODE:

- for the entire subroutine `WestOfHut`
- for the entire subroutine `GetClue`
- for the entire subroutine `PirateWalks`.

[12 marks]

1 7 . 2 SCREEN CAPTURE(S) showing the requested test described in **Task 4**.

The SCREEN CAPTURE(S) must show the map with the pirate's position as well as the displayed clue.

[1 mark]

END OF QUESTIONS

There are no questions printed on this page