A-level

COMPUTER SCIENCE

Paper 1

7517/1

Monday 10 June 2024 Afternoon

Time allowed: 2 hours 30 minutes

[Turn over]

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 puzzle1.txt, puzzle2.txt, puzzle3.txt and puzzle4.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 (also at the top of 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.**

[Turn over]

INFORMATION

- **The marks for questions are shown in brackets.**
- **The maximum mark for this paper is 100.**
- **No extra time is allowed for printing and collating.**
- **The question paper is divided into FOUR sections.**

ADVICE

You are advised to allocate time to each section as follows:
SECTION A – 40 minutes;
SECTION B – 20 minutes;
SECTION C – 20 minutes;
SECTION D – 70 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.

DO NOT TURN OVER UNTIL TOLD TO DO SO

SECTION A

You are advised to spend no longer than 40 MINUTES on this section.

Enter your answers for SECTION A in your Electronic Answer Document.

You MUST SAVE this document at regular intervals.

0	1

State THREE advantages of using subroutines.

For each advantage, you must explain how the advantage is achieved.
[3 marks]

0 2

Circular queues and linear queues are examples of data structures that can be implemented using a fixed-length array.

0 2 . 1

Explain why, when implemented using a fixed-length array, a circular queue is usually considered to be a better choice of data structure than a linear queue.
[2 marks]

0 2 . 2

Describe the steps that must be completed to remove (dequeue) an item from a circular queue that has been implemented using a fixed-length array.
[5 marks]

0 3

0 3 . 1

Describe the Halting problem. [2 marks]

0 3 . 2

Explain the importance of the Halting problem. [1 mark]

BLANK PAGE

[Turn over]

0 4

FIGURE 1 shows four sets R, S, T and U. Three dots (…) means the remaining members of the set follow the same pattern as the previous members of the set.

FIGURE 1

R = {a, b}
S = {a, abb, abbbb, abbbbbb, …}
T = {bb, bbbb, bbbbbb, …}
U = {c, d, bb, b}

0 4 . 1

What is meant by the cardinality of a set? [1 mark]

0 4 . 2

Explain what is wrong with the statement:

'The only subsets of R are the sets {a}, {b} **and** {a, b}'. **[1 mark]**

0 4 . 3

How many members are there in the set formed by the intersection of R and U? [1 mark]

0 4 . 4

The language defined by a regular expression can be represented as a set.

Explain the functionality of the | (vertical bar) metacharacter when it is used in a regular expression. [1 mark]

The members of the set V are strings that match the regular expression `a?b+`

Set W is formed by the union of sets S and T.

Set X is formed by the set operation V – W.

0 4 . 5

Write a regular expression that would match with all the members of the set W. [2 marks]

0 4 . 6

Write a regular expression that would match with all the members of the set X. [2 marks]

[Turn over]

0 5

`(3 + 4) * 5` **is an example of an infix expression. The same expression has been represented in a different expression format in FIGURE 2.**

FIGURE 2

```
3 4 + 5 *
```

0 5 . 1

What is the name of the expression format used in FIGURE 2? [1 mark]

0 5 . 2

Represent the infix expression `5 + 2 * 3 + 4` **in the same expression format used in FIGURE 2. [2 marks]**

BLANK PAGE

[Turn over]

0	6

FIGURE 3 shows a graph containing five nodes. FIGURE 4 shows how the graph in FIGURE 3 could be represented using three one-dimensional arrays: `Data`**,** `Dir1` **and** `Dir2`.

FIGURE 3

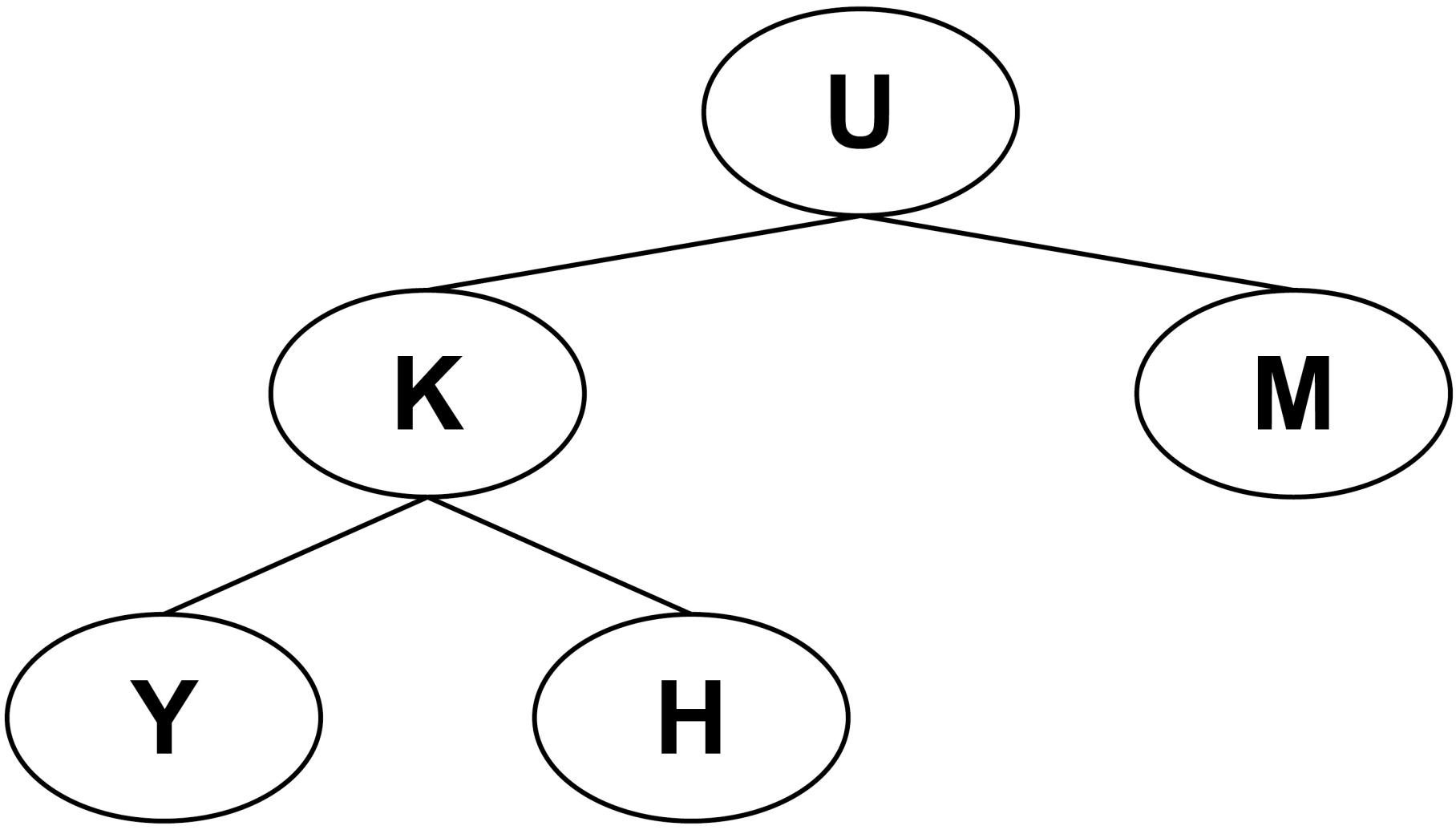


FIGURE 4

Index	Data	Dir1	Dir2
[0]	U	1	4
[1]	K	2	3
[2]	Y	−1	−1
[3]	H	−1	−1
[4]	M	−1	−1

0 6 . 1

The graph in FIGURE 3 is a binary tree. A binary tree is a rooted tree where each node has at most two child nodes.

There are three properties that a graph needs to have for it to be a tree. One of those properties is that it contains no cycles.

State the other TWO properties of this graph that make it a tree. [2 marks]

FIGURE 5, on the opposite page, contains pseudo-code for an algorithm that uses the arrays in FIGURE 4, on page 17.

FIGURE 5

```
Done ← False
Pos ← -1
Current ← 0
WHILE Done = False
WHILE Current ≠ -1
  Pos ← Pos + 1
  Temp[Pos] ← Current
  Current ← Dir1[Current]
ENDWHILE
IF Pos = -1 THEN
  Done ← True
ELSE
  OUTPUT Data[Temp[Pos]]
  Current ← Dir2[Temp[Pos]]
  Pos ← Pos - 1
ENDIF
ENDWHILE
```

[Turn over]

0 6 . 2

Complete the unshaded cells in TABLE 1, on pages 22 to 23, to show the result of tracing the algorithm shown in FIGURE 5 using the arrays in FIGURE 4.

FIGURE 4 is provided on page 17.

FIGURE 5 is provided on page 19.

Copy the contents of the unshaded cells in TABLE 1, on pages 22 to 23, into the table in your Electronic Answer Document. [7 marks]

BLANK PAGE

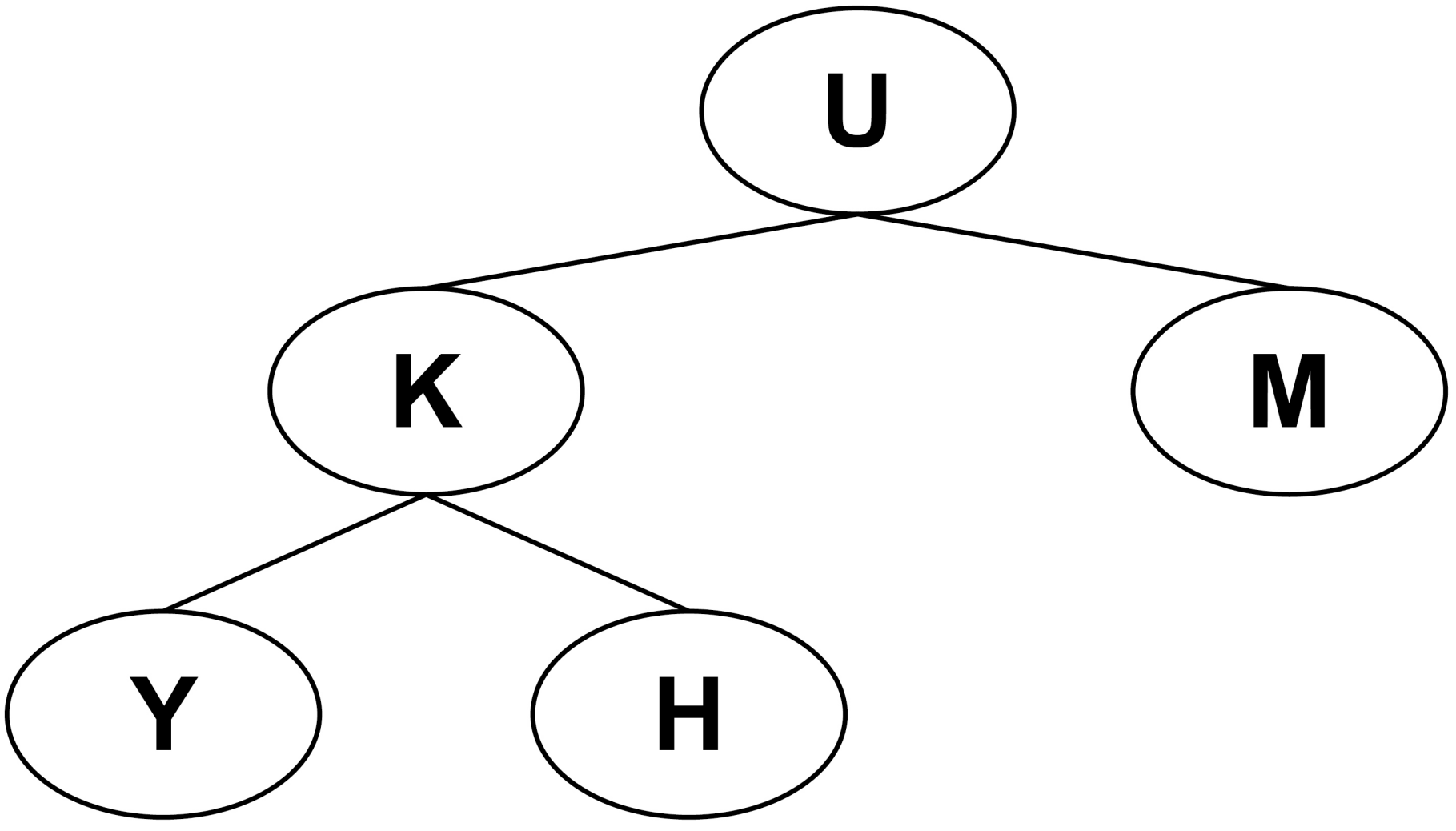
[Turn over]

TABLE 1

		Temp				
Done	Pos	[0]	[1]	[2]	Current	OUTPUT

[Turn over]

REPEAT OF FIGURE 3



0 6 . 3

The array `Temp` **needs to be able to store three values when used with the binary tree shown in FIGURE 3.**

For some binary trees with only five nodes, the array `Temp` **would need to be able to store five values.**

Describe the structure of a five-node binary tree that would require `Temp` **to be able to store five values. [2 marks]**

0 6 . 4

State the type of data structure the algorithm shown in FIGURE 5, on page 19, implements using the array `Temp`**. [1 mark]**

REPEAT OF FIGURE 5

```
Done ← False
Pos ← -1
Current ← 0
WHILE Done = False
  WHILE Current ≠ -1
    Pos ← Pos + 1
    Temp[Pos] ← Current
    Current ← Dir1[Current]
  ENDWHILE
  IF Pos = -1 THEN
    Done ← True
  ELSE
    OUTPUT Data[Temp[Pos]]
    Current ← Dir2[Temp[Pos]]
    Pos ← Pos - 1
  ENDIF
ENDWHILE
```

0 6 . 5

Describe the changes that need to be made to the algorithm shown in FIGURE 5 so that the order that the data values are output in is reversed.

In your answer, you should only describe changes to the existing lines of code in the algorithm; you must NOT suggest the addition of extra lines of code. [1 mark]

[Turn over]

SECTION B

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

Enter your answers to SECTION B in your Electronic Answer Document.

You MUST SAVE this document at regular intervals.

The question 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	7

Write a program that gets the user to enter an integer. It should keep doing this until they enter a value greater than 0.

The program should then tell the user if they have entered a perfectly bouncy number, a bouncy number or a number that is not bouncy.

A BOUNCY NUMBER is a number that is not an increasing number and not a decreasing number.

An INCREASING NUMBER is one where each digit is greater than or equal to the previous digit in the number.

[Turn over]

A DECREASING NUMBER is one where each digit is less than or equal to the previous digit in the number.

A PERFECTLY BOUNCY NUMBER is a bouncy number in which the number of digits that are followed by a larger digit is equal to the number of digits that are followed by a smaller digit.

EXAMPLES

- **13578 is not a bouncy number because it is an increasing number.**
- **973 is not a bouncy number because it is a decreasing number.**
- **98657 is a bouncy number.**
- **1111 is not a bouncy number because it is both an increasing number and a decreasing number.**

- **13421 is a perfectly bouncy number as exactly two digits are followed by a larger digit and there are also exactly two digits followed by a smaller digit.**
- **1829361 is a perfectly bouncy number as exactly three digits are followed by a larger digit and there are also exactly three digits followed by a smaller digit.**
- **13333331 is a perfectly bouncy number as there is exactly one digit followed by a larger digit and also exactly one digit followed by a smaller digit.**

EVIDENCE THAT YOU NEED TO PROVIDE

Include the following evidence in your Electronic Answer Document.

0 7 . 1

Your PROGRAM SOURCE CODE
[12 marks]

0 7 . 2

SCREEN CAPTURE(S) showing the results of testing the program by entering the integers:

- `-3`
- `14982`
- `1234`

You will need to execute your program more than once to test all of the integers. [1 mark]

SECTION C

You are advised to spend no more than 20 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 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 8

This question is about the `Puzzle` **class.**

The `DisplayPuzzle` **method uses concatenation.**

0 8 . 1

Explain what is meant by concatenation. [1 mark]

0 8 . 2

State the number of methods in the `Puzzle` **class that read data from a text file. [1 mark]**

0 8 . 3

State the number of methods in the `Puzzle` class that read data from a binary (non-text) file. [1 mark]

0 9

This question is about the `Cell` class.

0 9 . 1

Explain why `IsEmpty` could have been a private method instead of a public method. [1 mark]

0 9 . 2

The `CheckSymbolAllowed` **method in the** `Cell` **class uses a local variable.**

Explain ONE difference between a local variable and a private class attribute.

[1 mark]

[Turn over]

1 0

This question is about the `CheckForMatchWithPattern` **method in the** `Puzzle` **class.**

`CheckForMatchWithPattern` **uses nested iteration and exception handling.**

1 0 . 1

State the name of another method in the `Puzzle` **class that uses nested iteration.**

[1 mark]

1 0 . 2

Explain what problem would occur if the first two iteration structures in the method `CheckForMatchWithPattern` **had been inside the exception handling structure instead of outside it.**
[1 mark]

1 0 . 3

Describe what exception handling is used for in the `CheckForMatchWithPattern` **method. [3 marks]**

10.4

A value of 10 is returned sometimes when it should not be. FIGURE 6 and FIGURE 7 show example 4×4 grids. The pattern in FIGURE 6 correctly results in a value of 10 being returned. The pattern in FIGURE 7, on the opposite page, also results in a value of 10 being returned, but it should return 0.

FIGURE 6

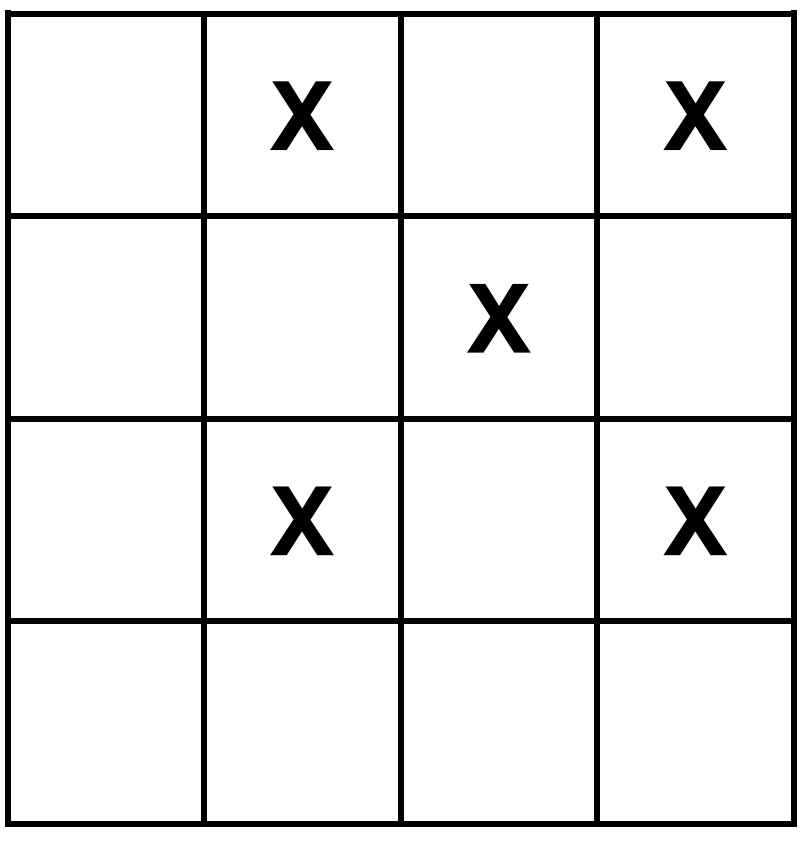
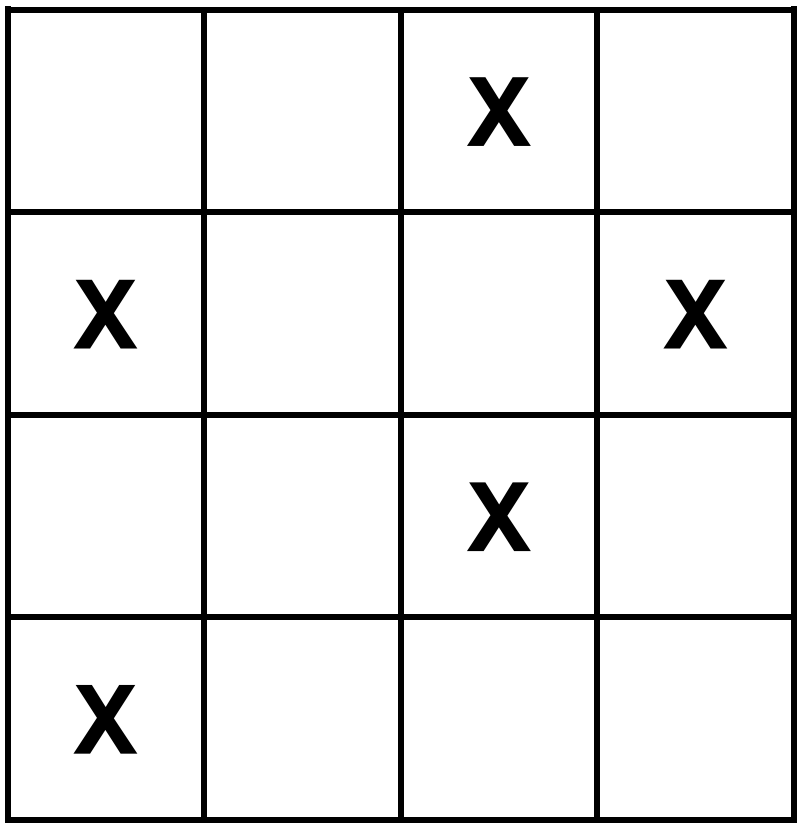


FIGURE 7



Explain why the pattern in FIGURE 7 results in the value 10 being returned by the `CheckForMatchWithPattern` **method. [1 mark]**

[Turn over]

SECTION D

You are advised to spend no more than 70 MINUTES on this section.

Enter your answers to SECTION D 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	1

This question refers to the method `AttemptPuzzle` **in the** `Puzzle` **class.**

The Skeleton Program is to be changed so that it checks the choice made by the user for the COLUMN number. A choice is valid if it is between one and the number of columns in the puzzle inclusive. You do NOT need to add checks for the row number.

The program should keep getting the user to enter a value until a valid choice has been made.

[Turn over]

WHAT YOU NEED TO DO

TASK 1

Modify the method `AttemptPuzzle` **so it checks that the value entered by the user is valid. If an invalid value is entered, the user should be made to enter another value.**

TASK 2

Test that the changes you have made work:

- **run the Skeleton Program**
- **press the Enter key**
- **enter** `1`
- **enter** `10`
- **enter** `4`

EVIDENCE THAT YOU NEED TO PROVIDE

Include the following evidence in your Electronic Answer Document.

1 1 . 1

Your PROGRAM SOURCE CODE for the amended method `AttemptPuzzle`**.**
[4 marks]

1 1 . 2

SCREEN CAPTURE(S) showing the results of the requested test. [1 mark]

1	2

This question extends the Skeleton Program by tracking the user's average score and highest score for the puzzle.

After the user chooses not to do another puzzle, the program should display the user's highest score achieved on the puzzle and the user's average score. Your amended Skeleton Program should calculate the average and highest scores for any number of puzzle attempts.

EXAMPLE

If the user:

- **gets a score of 30 when they complete a puzzle**
- **decides to do another puzzle**
- **gets a score of 20 for the second puzzle**
- **decides not to do another puzzle**

the program should display messages saying that their highest score was 30 and their average score was 25

WHAT YOU NEED TO DO

TASK 1

Modify the `Main` subroutine so that after the user decides not to attempt another puzzle their highest score for the puzzles attempted and their average score for the puzzles attempted are displayed.

[Turn over]

TASK 2

Test that the changes you have made work:

- **run the Skeleton Program**
- **load the file** `puzzle1`
- **enter** `5`
- **enter** `5`
- **enter** `X`
- **enter** `Y`
- **load the file** `puzzle1`
- **enter** `1`
- **enter** `4`
- **enter** `X`
- **enter** `N`

EVIDENCE THAT YOU NEED TO PROVIDE

Include the following evidence in your Electronic Answer Document.

1 2 . 1

Your PROGRAM SOURCE CODE for the amended subroutine `Main`**. [5 marks]**

1 2 . 2

SCREEN CAPTURE(S) showing the requested test. [1 mark]

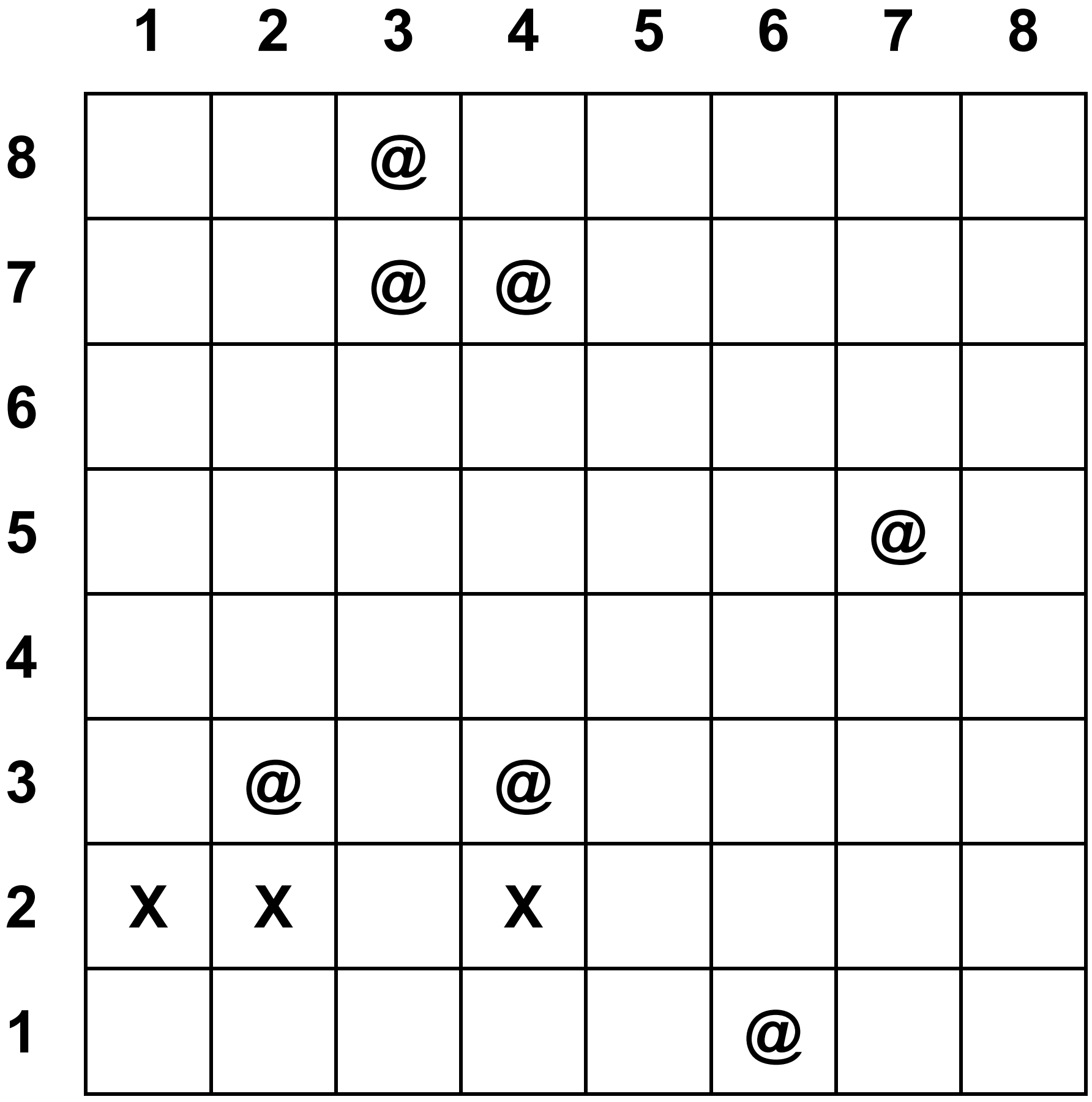
1	3

This question extends the Skeleton Program by giving the user an option to shift all the cells in one row in the puzzle one place to the left.

When shifting left, the first cell in the row will move to the last position in the row.

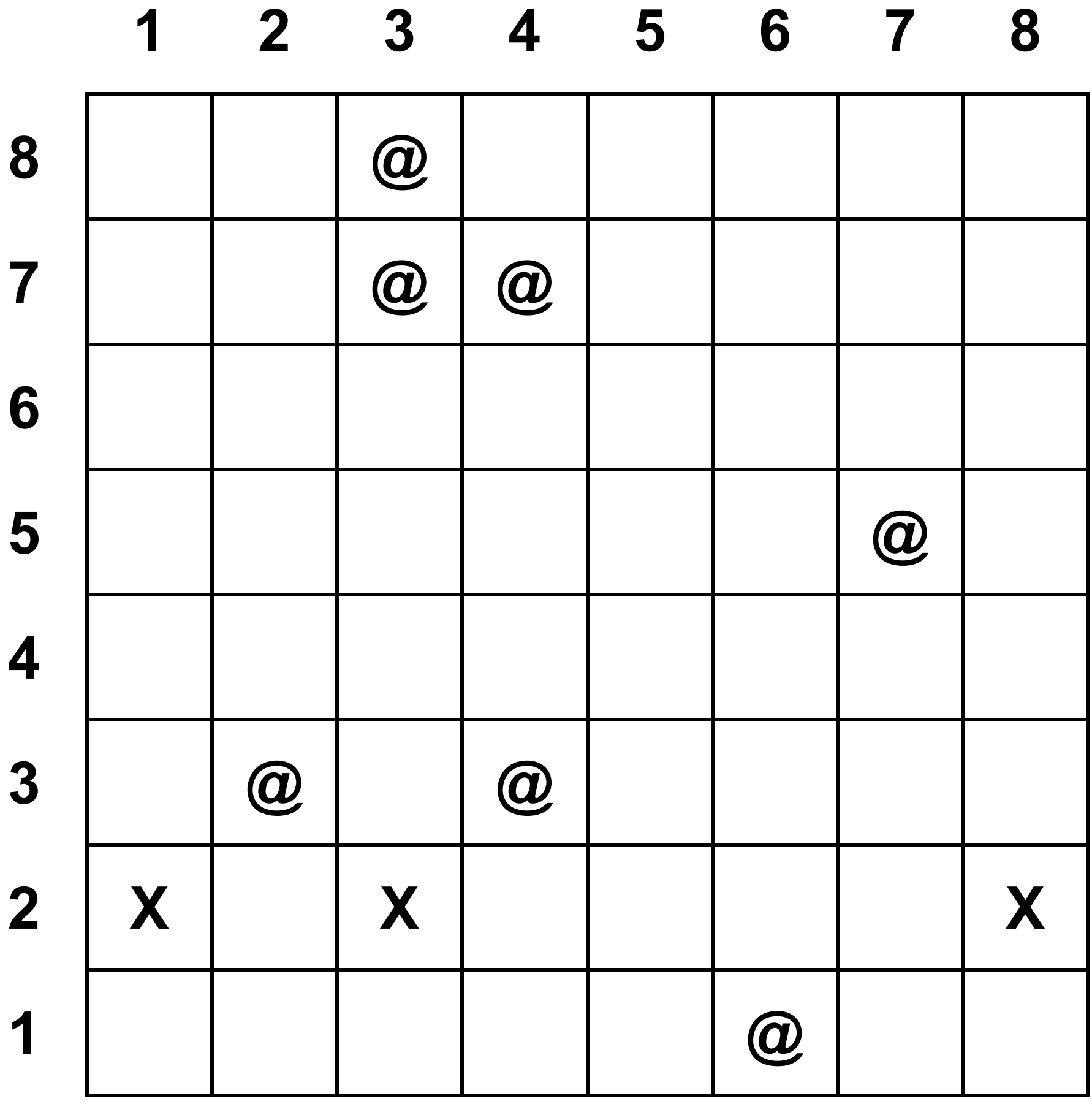
FIGURE 8, on the opposite page, shows an example puzzle and FIGURE 9, on page 52, shows the result obtained from shifting all the cells in row 2 from the puzzle in FIGURE 8 one place to the left.

FIGURE 8



[Turn over]

FIGURE 9



After the cells have been shifted, the user's score should be reduced by 20 and the new state of the grid and the user's new score should be displayed. No other changes to the user's score should be made.

Shifting the cells does not count as a turn. After choosing to shift cells, the user should carry on with the rest of their turn by selecting a cell to place a symbol in.

When answering this question, you should make sure your program code will work for any size of puzzle grid.

[Turn over]

WHAT YOU NEED TO DO

TASK 1

Create a new method called `ShiftCellsInRowLeft` **in the** `Puzzle` **class that takes an integer parameter that specifies the row number to use with the shift.**

Each cell, in the row indicated by the parameter, should be moved in `Grid` **so that it is one place to the left; the leftmost cell should shift to the end of the row.**

TASK 2

Modify the `AttemptPuzzle` **method in the** `Puzzle` **class so that it gives the user the option to shift the cells in a row. Inside the iteration structure used to get a row number from the user your program should:**

- display a modified message telling the user to enter a row number or to enter `0` to shift cells
- if the user enters `0`:
  - ask the user to enter the row number for the row to shift
  - call the method `ShiftCellsInRowLeft` with the number entered as a parameter
  - subtract 20 from the user's score
  - display the new grid
  - display the new score
  - set `Valid` to be false.

[Turn over]

TASK 3

Test that the changes you have made work:

- **run the Skeleton Program**
- **load the file** `puzzle2`
- **enter** `0`
- **enter** `1`
- **enter** `1`

EVIDENCE THAT YOU NEED TO PROVIDE

Include the following evidence in your Electronic Answer Document.

1 3 . 1

Your PROGRAM SOURCE CODE for the new method `ShiftCellsInRowLeft` **and the amended method** `AttemptPuzzle`**. [11 marks]**

1 3 . 2

SCREEN CAPTURE(S) showing the requested test. [1 mark]

1 3 . 3

The movement of the cells within a row can be described using vectors.

State the 2-vector that describes the movement for all but the leftmost cell in the row. [1 mark]

1 3 . 4

State the 2-vector that describes the movement for the leftmost cell in the row.

Your vector should work for any grid size. [1 mark]

1	4

This question extends the Skeleton Program so that there is a new type of cell – a countdown cell.

A countdown cell has a timer associated with it that decreases by one each time the user tries to place a symbol in the puzzle grid.

The location of a countdown cell is shown by a numeric digit representing the current value of its timer. When its timer reaches zero, the cell's symbol changes to an @.

A countdown cell should be added to the board each time the user's score is increased. The location for the countdown cell should be an empty cell on the grid and selected randomly.

[Turn over]

WHAT YOU NEED TO DO

TASK 1

Create a new class called `CountdownCell` **that is a subclass of the** `BlockedCell` **class.**

Create a constructor for the `CountdownCell` **class that sets the initial value of the timer and makes sure that the symbol displayed for the cell will be the value of its timer. The first symbol that should be displayed for the timer value of a** `CountdownCell` **is** `3`

Create a method `UpdateCell` **in the** `CountdownCell` **class that:**

- **overrides the method from the base class**
- **decreases the value of the timer by one**
- **changes the symbol to** `@` **when the timer has a value of zero.**

TASK 2

Modify the `AttemptPuzzle` **method in the** `Puzzle` **class so that, if the amount to add to the score is greater than zero, it selects an empty cell in the grid at random and replaces that cell with a new** `CountdownCell`**.**

TASK 3

Modify the `AttemptPuzzle` **method in the** `Puzzle` **class so that, immediately before it checks if the number of symbols left is zero, it calls the** `UpdateCell` **method for each cell in the grid.**

[Turn over]

TASK 4

Test that the changes you have made work:

- **run the Skeleton Program**
- **load the file** `puzzle3`
- **enter** `1`
- **enter** `4`
- **enter** `X`
- **place a T in an empty cell**
- **place a T in an empty cell**
- **place a T in an empty cell.**

EVIDENCE THAT YOU NEED TO PROVIDE

Include the following evidence in your Electronic Answer Document.

1 4 . 1

Your PROGRAM SOURCE CODE for the new class `CountdownCell` **and the amended** `AttemptPuzzle` **method. [13 marks]**

1 4 . 2

SCREEN CAPTURE(S) showing the requested test. [1 mark]

END OF QUESTIONS

BLANK PAGE