Surname _____

Forename(s) _____

Centre Number _____

Candidate Number _____

Candidate Signature _____

I declare this is my own work.

GCSE

COMPUTER SCIENCE

Paper 1 Computational thinking and programming skills – Python

8525/1B

Wednesday 15 May 2024 Afternoon

Time allowed: 2 hours

[Turn over]

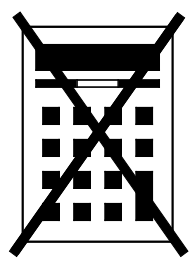
JUN248525 1B01

BLANK PAGE

On the front page of this book, write your surname and forename(s), your centre number, your candidate number and add your signature.

MATERIALS

For this paper you must have:



- **the Diagram Booklet.**

You must NOT use a calculator.

INSTRUCTIONS

- **Use black ink or black ball-point pen. Use pencil only for drawing.**
- **Answer ALL questions.**

[Turn over]

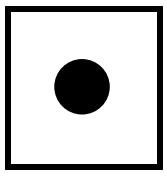
- **You must answer the questions in the spaces provided.**
- **If you need extra space for your answer(s), use the lined pages at the end of this book. Write the question number against your answer(s).**
- **Do all rough work in this book. Cross through any work you do not want to be marked.**
- **Questions that require a coded solution must be answered in Python.**
- **You should assume that all indexing in code starts at 0 unless stated otherwise.**

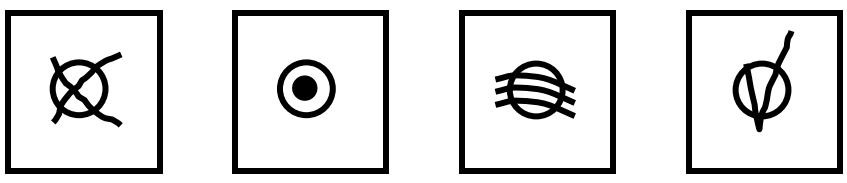
INFORMATION

The total number of marks available for this paper is 90.

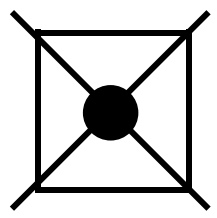
ADVICE

For the multiple-choice questions, completely fill in the lozenge alongside the appropriate answer.

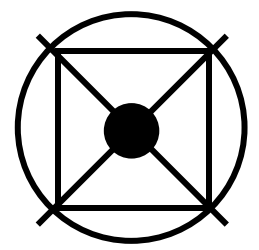
CORRECT METHOD 

WRONG METHODS 

If you want to change your answer you must cross out your original answer as shown.



If you wish to return to an answer previously crossed out, ring the answer you now wish to select as shown.



DO NOT TURN OVER UNTIL TOLD TO DO SO

Answer ALL questions.

0 1

FIGURE 1, provided on page 2 of the Diagram Booklet, shows an algorithm, represented using pseudo-code.

The algorithm assigns different values to two variables, then asks the user to input a letter.

0 1 . 1

Which pseudo-code statement assigns the length of the string `film` **to a variable called** `value`**?**

Shade ONE lozenge. [1 mark]

○ **A** `film ← LEN(value)`

○ **B** `film ← film + value`

○ **C** `value ← film`

○ **D** `value ← LEN(film)`

[Turn over]

0 1 . 2

The `POSITION` **subroutine returns the position of the first occurrence of a character in a string.**

For example:

- `POSITION("Godzilla vs. Kong", "o")` **would return** `1`
- `POSITION("Godzilla vs. Kong", "z")` **would return** `3`

`letter` **and** `film` **are variables used in the algorithm in FIGURE 1, provided on page 2 of the Diagram Booklet.**

0 9

Complete the pseudo-code statement to find the position of the first occurrence of the contents of `letter` **in** `film` **and store this position in the variable** `location`

You MUST use the `POSITION` **subroutine in your answer. [1 mark]**

`location` ← _____

_____

_____

[Turn over]

0 1 . 3

Which of the following would be the most suitable data type for the variable `year`**?**

Shade ONE lozenge. [1 mark]

- ○ **A Boolean**
- ○ **B character**
- ○ **C integer**
- ○ **D real**

0 1 . 4

Describe what is meant by an assignment statement in a program.
[1 mark]

0 1 . 5

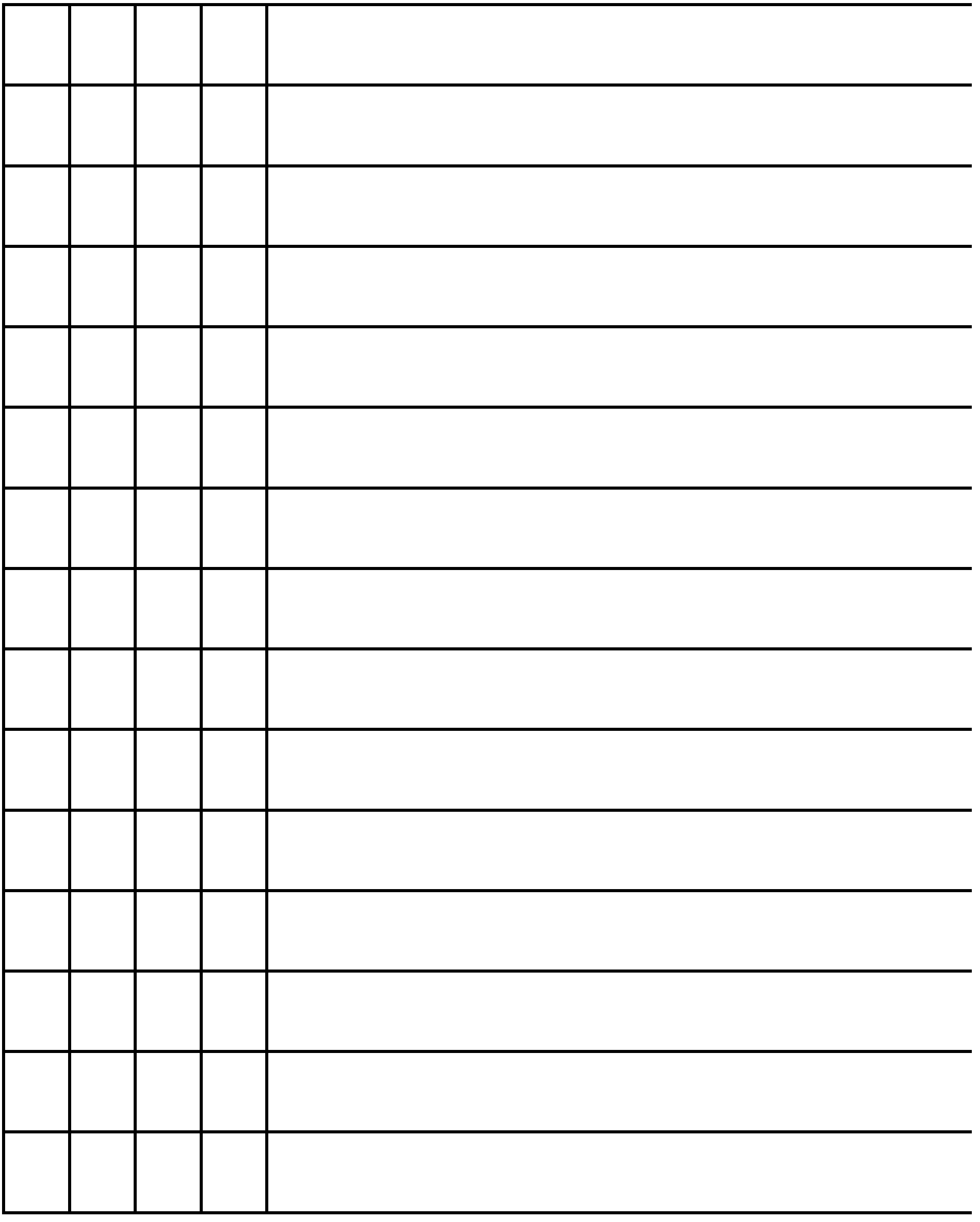
Write a Python program that:

- **gets the user to enter the name of a film**
- **displays** `You entered` **followed by the name of the film entered by the user.**

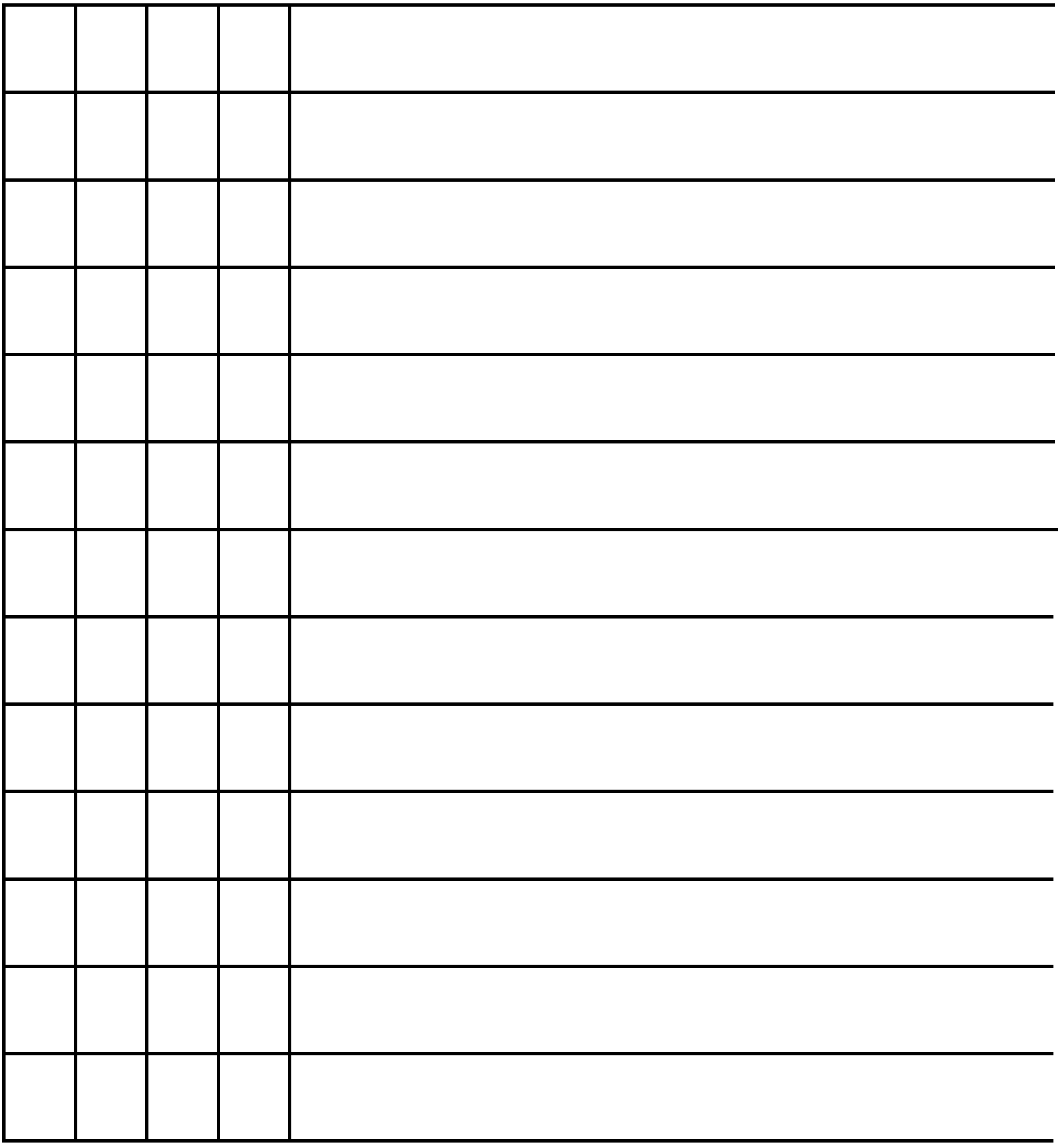
The output from the program MUST be on one line.

You SHOULD use indentation as appropriate, meaningful variable name(s) and Python syntax in your answer.

The answer grid, on pages 13–14, contains vertical lines to help you indent your code accurately.
[2 marks]



[Turn over]



6

BLANK PAGE

[Turn over]

0	2

FIGURE 2, provided on page 3 of the Diagram Booklet, shows an algorithm, represented using pseudo-code.

- **Line numbers are included but are not part of the algorithm.**

The modulus operator is used to calculate the remainder after dividing one integer by another.

For example:

- `14 MOD 3` **evaluates to** `2`
- `24 MOD 5` **evaluates to** `4`

0 2 . 1

Where is a relational operator FIRST used in the algorithm in FIGURE 2, provided on page 3 of the Diagram Booklet?

Shade ONE lozenge. [1 mark]

- ○ **A Line number 1**
- ○ **B Line number 2**
- ○ **C Line number 3**
- ○ **D Line number 6**

0 2 . 2

In the algorithm in FIGURE 2, provided on page 3 of the Diagram Booklet, what will be the output when the user input is 5**?**

Shade ONE lozenge. [1 mark]

- ○ **A** Almost
- ○ **B** False
- ○ **C** True
- ○ **D** Unknown

0 2 . 3

Which value input by the user would result in `True` being output by the algorithm in FIGURE 2, provided on page 3 of the Diagram Booklet?

Shade ONE lozenge. [1 mark]

- ○ **A** -1
- ○ **B** 10
- ○ **C** 20
- ○ **D** 21

[Turn over]

0 2 . 4

Rewrite LINE 2 from the algorithm in FIGURE 2 WITHOUT using the `NOT` **operator.**

FIGURE 2 is provided on page 3 of the Diagram Booklet.

The algorithm must still have the same functionality. [1 mark]

0 2 . 5

A user inputs a value into the algorithm in FIGURE 2, provided on page 3 of the Diagram Booklet.

State ONE value that the user could input that would result in an output of `Unknown` **[1 mark]**

0 3

FIGURE 3, provided on page 4 of the Diagram Booklet, shows an incomplete Python program for a number guessing game.

- **Line numbers are included but are not part of the program.**

0 3 . 1

The program should generate a random number between 1 and 100 (including 1 and 100). This will be the number the user has to guess.

Write the Python code that should be used on LINE 2 in FIGURE 3, provided on page 4 of the Diagram Booklet, to:

- **generate a random number between 1 and 100 inclusive**

- **assign this number to the appropriate variable from the program.**

You MUST use `random.randrange(a, b)` **in your Python code.**

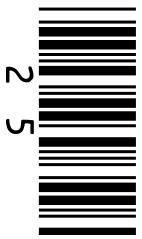
`random.randrange(a, b)` **generates a random integer in the range** `a` **to** `b` **starting at** `a` **but finishing one before** `b` **[2 marks]**

0 3 . 2

Complete the test plan in TABLE 1 to test the validation of `userNumber` in the program in FIGURE 3, provided on page 4 of the Diagram Booklet. [2 marks]

TABLE 1

TEST NUMBER	TEST TYPE	TEST DATA	EXPECTED RESULT
1	Erroneous	150	
2	Boundary		
3	Normal		Valid number entered



BLANK PAGE

[Turn over]

0 3 . 3

In an earlier version of the program in FIGURE 3, provided on page 4 of the Diagram Booklet, LINE 5 contained one syntax error and one logic error:

```
whil userNumber < 1 or userNumber >= 100:
```

Complete the table, on the opposite page, to describe the errors in the program on LINE 5. [2 marks]

ERROR TYPE	DESCRIPTION
Syntax error	
Logic error	

0 4 . 1

Define the term ABSTRACTION. [1 mark]

0 4 . 2

State the name for the process of breaking a problem down into sub-problems. [1 mark]

BLANK PAGE

[Turn over]

0 5

FIGURE 4, provided on pages 6–7 of the Diagram Booklet, shows an algorithm, represented using pseudo-code.

The algorithm calculates the total cost of hiring a hotel for a wedding.

Complete the table, on the opposite page, using the algorithm in FIGURE 4. [3 marks]

Input value for `numberOfGuests`	**Input value for** `numberOfRooms`	**OUTPUT**
`50`	`30`	
`20`	`10`	
`500`	`5`	

[Turn over]

5

0 6

A university is writing a program to calculate a student's total mark for three essays.

If any essays are handed in late, the total mark is reduced.

Write a Python program to calculate the total mark.

You should assume there are three integer variables called `e1`**,** `e2` **and** `e3` **which have already been given values to represent the marks of the three essays.**

The program should:

- **get the user to enter the number of essays handed in late and store the number in a variable**

- calculate the total mark for the three essays
  - if only one essay is handed in late, the total mark is reduced by 10
  - if more than one essay is handed in late, the total mark should be halved
  - the total mark should NOT be less than 0
- output the total mark.

[Turn over]

You SHOULD use indentation as appropriate, meaningful variable name(s) and Python syntax in your answer.

The answer grid, below and on pages 35–37, contains vertical lines to help you indent your code. [7 marks]

[Turn over]

[Turn over]

0 7

A shop owner wants to create stock codes for each type of sweet they sell.

FIGURE 5, provided on page 8 of the Diagram Booklet, shows some of the sweets.

A stock code is made up of the:

- `sweetID`
- **first letter and the second letter in** `sweetName`
- **first letter of the** `brand`

For example:

- **the stock code for** `WINE GUMS` **would be** `S1WIM`
- **the stock code for** `STARBURST` **would be** `S3STW`

Write a Python program to create the stock code for a sweet.

The program should:

- **get the user to enter the** `sweetID`**,** `sweetName` **and** `brand`
- **create the stock code**
- **assign the stock code to a variable called** `code`

[Turn over]

You SHOULD use indentation as appropriate, meaningful variable name(s) and Python syntax in your answer.

The answer grid, below and on pages 41–42, contains vertical lines to help you indent your code. [4 marks]

[Turn over]

42

11

42

BLANK PAGE

[Turn over]

0 8

FIGURE 6, provided on page 9 of the Diagram Booklet, shows an algorithm, represented using pseudo-code.

The `DIV` operator is used for integer division.

Complete the trace table, on the opposite page, for the algorithm in FIGURE 6.

Part of the table has already been filled in.

You may not need to use all the rows in the table.
[6 marks]

i	daysTotal	Weeks			weeksTotal
		[0]	[1]	[2]	
		0	0	0	

[Turn over]

0 9 . 1

Which of the following best describes a DATA STRUCTURE?

Shade ONE lozenge. [1 mark]

○ **A A number with a fractional part**

○ **B A value such as a whole number**

○ **C All of the data used and stored within a program**

○ **D An organised collection of values**

0 9 . 2

FIGURE 7, on page 49, shows an INCOMPLETE algorithm, represented using pseudo-code.

The algorithm is used to store and manage books using records.

The algorithm should do the following:

- **create a record definition called** `Book` **with the fields** `bookName`**,** `author` **and** `price`
- **create a variable for each book using the record definition.**

[Turn over]

Complete FIGURE 7, on the opposite page, by filling in the gaps using the items in TABLE 2, provided on page 10 of the Diagram Booklet.

- **You may need to use some of the items in TABLE 2 more than once.**
- **You will NOT need to use all the items in TABLE 2.**

[3 marks]

FIGURE 7

```
RECORD ________________
   bookName : String
   ________________ : String
   price : ________________
ENDRECORD

B1 ← Book("The Book Thief", "M Zusak", 9.99)
B2 ← ________________ ("Divergent", "V Roth", 6.55)
```

[Turn over]

0 9 . 3

Write an algorithm using pseudo-code to display the name of the most expensive book.

The algorithm should:

- **compare the price of** `B1` **and the price of** `B2`
- **output the book name of the most expensive book**
- **output** `Neither` **if the books are the same price.**

The algorithm should work for any values stored in `B1` **and** `B2` **[3 marks]**

[Turn over]

13

1 0

FIGURE 8, provided on page 12 of the Diagram Booklet, shows a Python program.

1 0 . 1

State what will be displayed by the `print` **statement when the subroutine** `First` **is called with the values** `3`**,** `4` **and** `4` **for the parameters** `p1`**,** `p2` **and** `p3`
[1 mark]

1 0 . 2

State what will be displayed by the `print` **statement when the subroutine** `First` **is called with the values** `3`**,** `4` **and** `8` **for the parameters** `p1`**,** `p2` **and** `p3`

[1 mark]

1 1

A program is to be written to authenticate a username and password entered by the user.

FIGURE 9, provided on page 13 of the Diagram Booklet, shows the only two pairs of valid usernames and passwords.

Write a Python program to authenticate a username and password.

The program should:

- **get the user to enter a username**
- **get the user to enter a password**
- **display the message** `Access denied` **if the username and password pair entered is not valid**

- **display the message** `Access granted` **if the username and password pair entered is valid**
- **repeat until a valid username and password pair is entered.**

[Turn over]

You SHOULD use indentation as appropriate, meaningful variable name(s) and Python syntax in your answer.

The answer grid, below and on pages 57–59, contains vertical lines to help you indent your code accurately. [7 marks]

[Turn over]

1	2

A program is being written to solve a sliding puzzle.

- **The sliding puzzle uses a 3 x 3 board.**
- **The board contains eight tiles and one blank space.**
- **Each tile is numbered from 1 to 8**
- **On each turn, a tile can only move one position up, down, left, or right.**
- **A tile can only be moved into the blank space if it is next to the blank space.**
- **The puzzle is solved when the tiles are in the correct final positions.**

FIGURE 10, provided on page 14 of the Diagram Booklet, shows an example of how the tiles might be arranged on the board at the start of the game with the blank space in the position (0, 1).

FIGURE 11, provided on page 14 of the Diagram Booklet, shows the correct final positions for the tiles when the puzzle is solved.

The blank space (shown in black) is represented in the program as number 0

[Turn over]

TABLE 3, provided on pages 16–19 of the Diagram Booklet, describes the purpose of three subroutines the program uses.

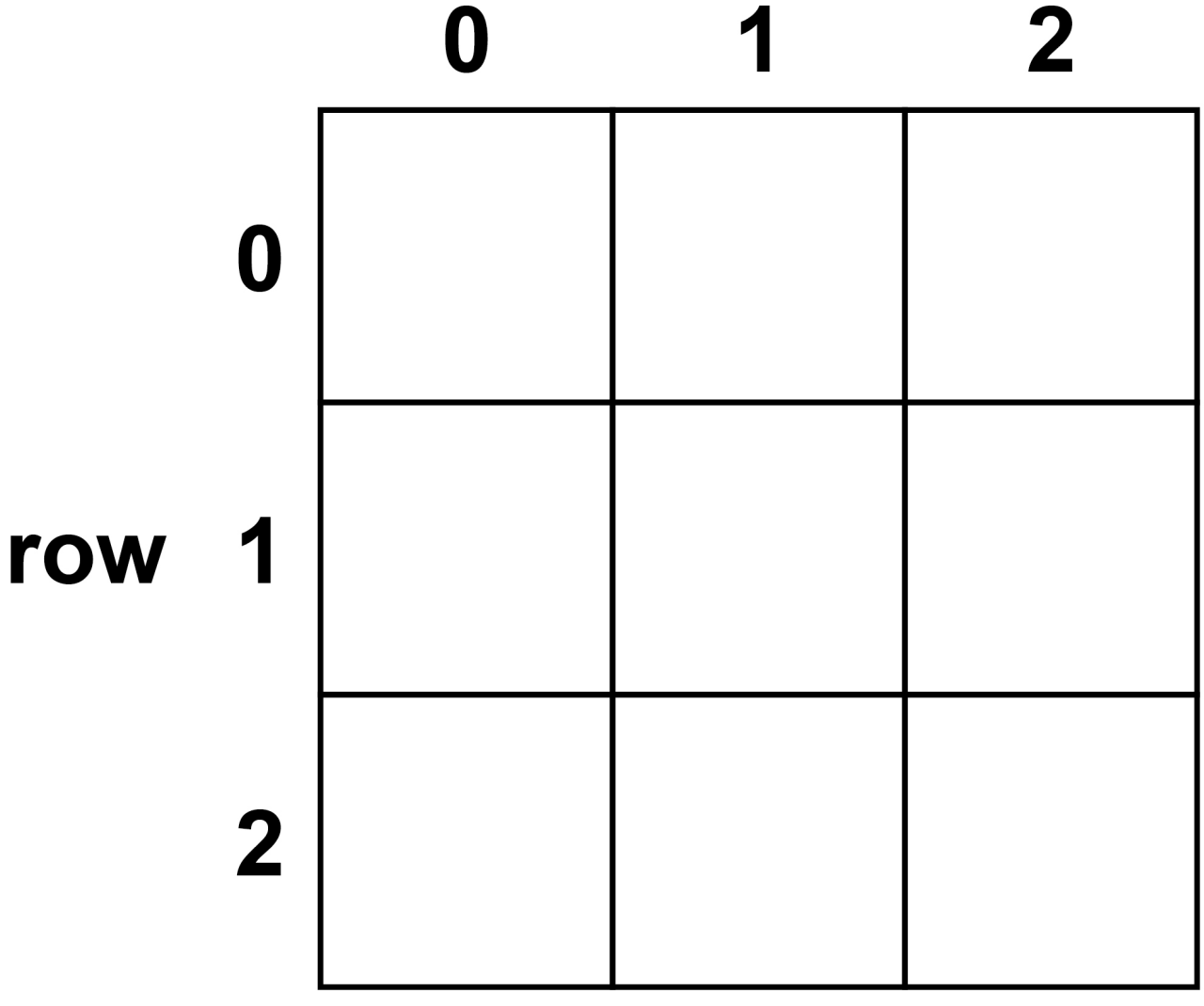
FIGURE 12 and FIGURE 13 are provided on page 20 of the Diagram Booklet.

12.1

The Python program shown in FIGURE 14, provided on page 21 of the Diagram Booklet, uses the subroutines in TABLE 3, provided on pages 16–19 of the Diagram Booklet.

The program is used with the board shown in FIGURE 15, provided on page 21 of the Diagram Booklet.

Complete the board to show the new positions of the tiles after the program in FIGURE 14 is run. [2 marks]



[Turn over]

FIGURE 16, provided on page 22 of the Diagram Booklet, shows part of a Python program that uses the `getTile` **subroutine from TABLE 3, provided on pages 16–19 of the Diagram Booklet.**

The program is used with the board shown in FIGURE 17, provided on page 22 of the Diagram Booklet.

1 2 . 2

Which TWO of the following statements about the program in FIGURE 16, provided on page 22 of the Diagram Booklet, are TRUE when it is used with the board in FIGURE 17, provided on page 22 of the Diagram Booklet?

Shade TWO lozenges, on the opposite page. [2 marks]

○ **A Nested iteration is used.**

○ **B The final value of** `ref1` **will be** `0`

○ **C The number of comparisons made between** `getTile(i, j)` **and** `0` **will be nine.**

○ **D The outer loop,** `for i in range(3)`**, will execute nine times.**

○ **E The values of** `i` **and** `j` **do not change when the program is executed.**

[Turn over]

FIGURE 16 and FIGURE 17 are provided on page 22 of the Diagram Booklet.

12.3 **Explain the purpose of the FIRST iteration structure in the program in FIGURE 16, provided on page 22 of the Diagram Booklet. [1 mark]**

12.4 Explain the purpose of the SECOND iteration structure in the program in FIGURE 16, provided on page 22 of the Diagram Booklet. [1 mark]

12.5 State the purpose of the program in FIGURE 16, provided on page 22 of the Diagram Booklet. [1 mark]

1 2 . 6

TABLE 4, provided on page 24 of the Diagram Booklet, shows a description of the `getTile` subroutine previously described in more detail in TABLE 3, provided on pages 16–19 of the Diagram Booklet.

FIGURE 18 and FIGURE 19, provided on page 25 of the Diagram Booklet, show example boards.

Write a Python program to:

- **check that in the first row:**
  - **the second tile number is one more than the first tile number**
  - **the third tile number is one more than the second tile number**
- **display `Yes` when the row meets both conditions above**

- **display** `No` **when the row does not meet both conditions above.**

For example:

- **for the board in FIGURE 18, the program would display** `No`
- **for the board in FIGURE 19, the program would display** `Yes`

You MUST use the `getTile` **subroutine in your Python code.**

[Turn over]

You SHOULD use indentation as appropriate, meaningful variable name(s) and Python syntax in your answer.

The answer grid, below and on pages 71–72, contains vertical lines to help you indent your code accurately. [4 marks]

[Turn over]

BLANK PAGE

[Turn over]

12.7

TABLE 5, provided on page 26 of the Diagram Booklet, describes the purpose of another two subroutines the program uses.

TABLE 6, provided on page 27 of the Diagram Booklet, shows a description of the `move` subroutine previously described in more detail in TABLE 3, provided on pages 16–19 of the Diagram Booklet.

Write a Python program to help the user solve the puzzle.

The program should:

- **get the user to enter the row number of a tile to move**

- **get the user to enter the column number of a tile to move**
- **check if the tile in the position entered is next to the blank space**
  - **if it is, move that tile to the position of the blank space**
  - **if it is not, output** `Invalid move`
- **repeat these steps until the puzzle is solved.**

You MUST use the subroutines in TABLE 5 and TABLE 6.

[Turn over]

You SHOULD use indentation as appropriate, meaningful variable name(s) and Python syntax in your answer.

The answer grid, below and on pages 77–78, contains vertical lines to help you indent your code accurately. [6 marks]

[Turn over]

78

17

7 8

1 3

Explain how the linear search algorithm works. [3 marks]

14.1

State ONE property of local variables that is NOT true for all variables.
[1 mark]

BLANK PAGE

[Turn over]

1 4 . 2

Using Python, write a subroutine to help a museum review the number of visitors in a month.

The subroutine must:

- **have the identifier** `countDays`
- **have the number of days a museum was open in the last month as a parameter**
- **get the user to enter the number of visitors to the museum for each of those days**
- **count how many of those days the museum had more than 200 visitors**
- **return the count.**

You SHOULD use indentation as appropriate, meaningful variable name(s) and Python syntax in your answer.

The answer grid, below and on pages 84–87, contains vertical lines to help you indent your code. [6 marks]

[Turn over]

[Turn over]

10

1	5

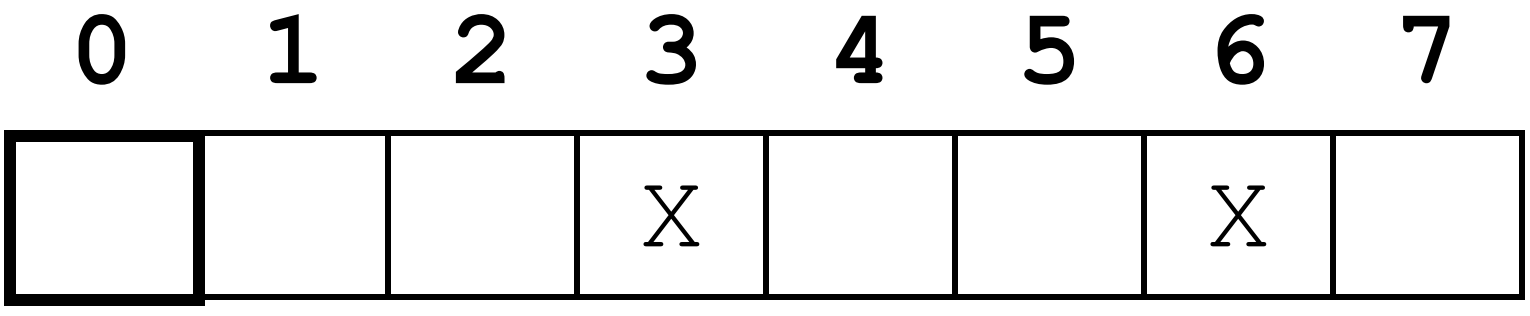
A programmer is writing a game.

The game uses a row of cells represented as a list. FIGURE 20, provided on page 29 of the Diagram Booklet, shows an example.

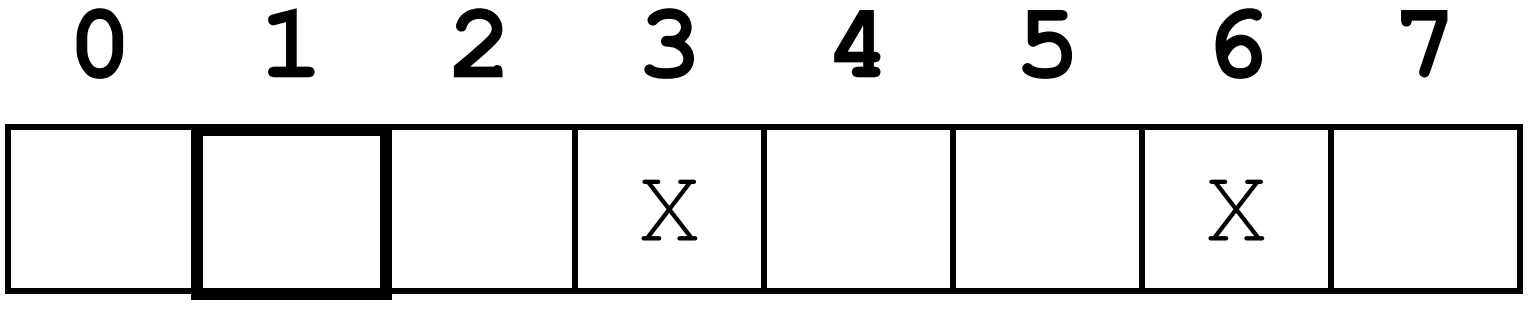
FIGURE 21, provided on pages 30–31 of the Diagram Booklet, describes how the game is to be played.

For example, using the list in FIGURE 20:

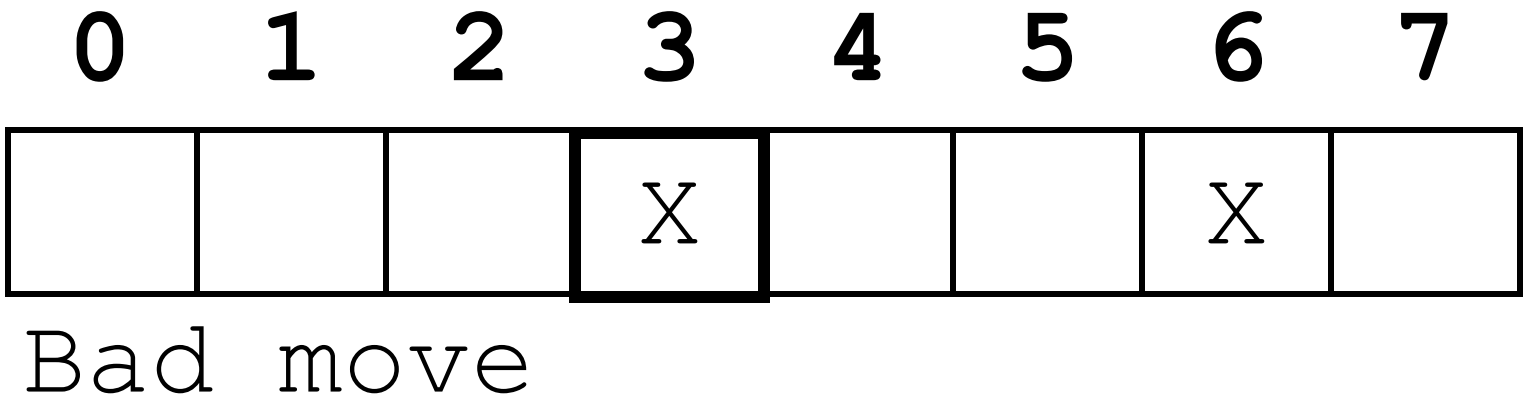
- **the player starts in position** 0



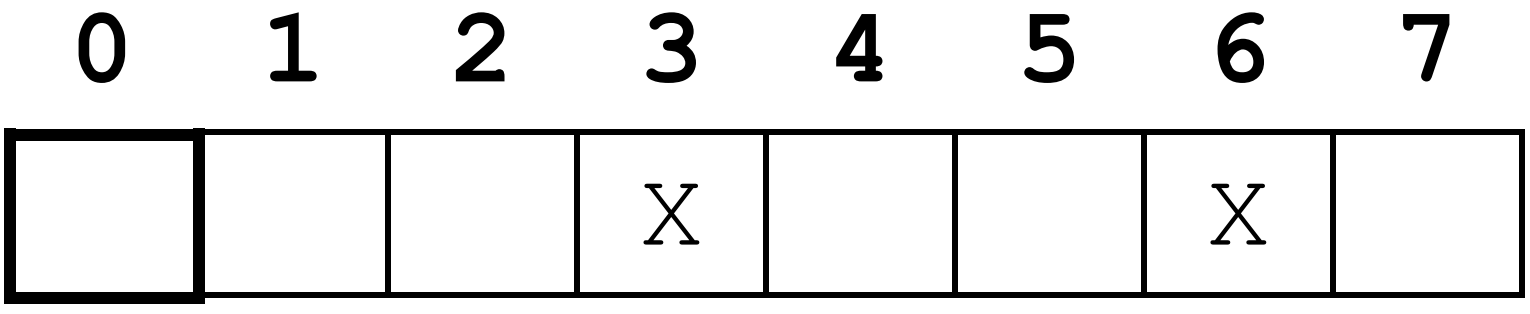
- **if the player enters a** `1`**, then they move to position** `1`



- **if the player then enters a** `2`**,** `Bad Move` **is displayed as position** `3` **contains an** `X`

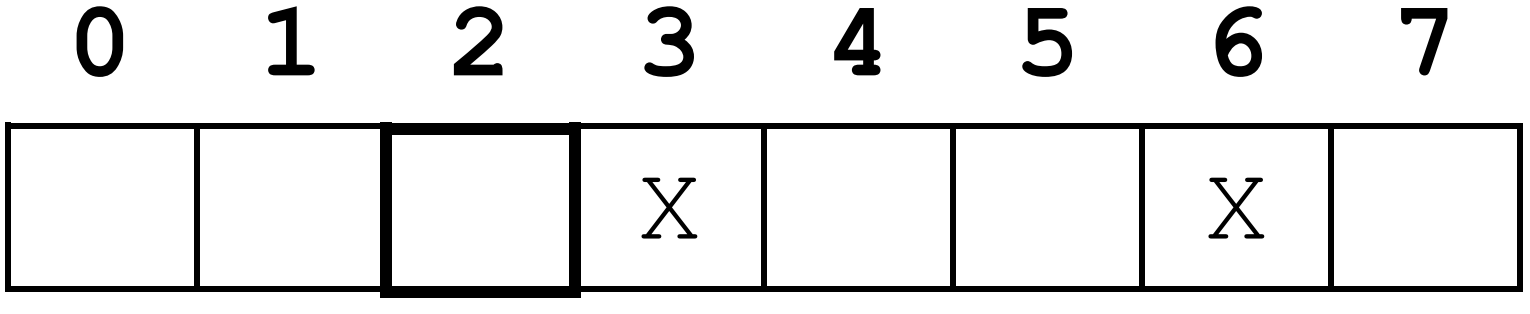


- **the player then goes back to position** `0`

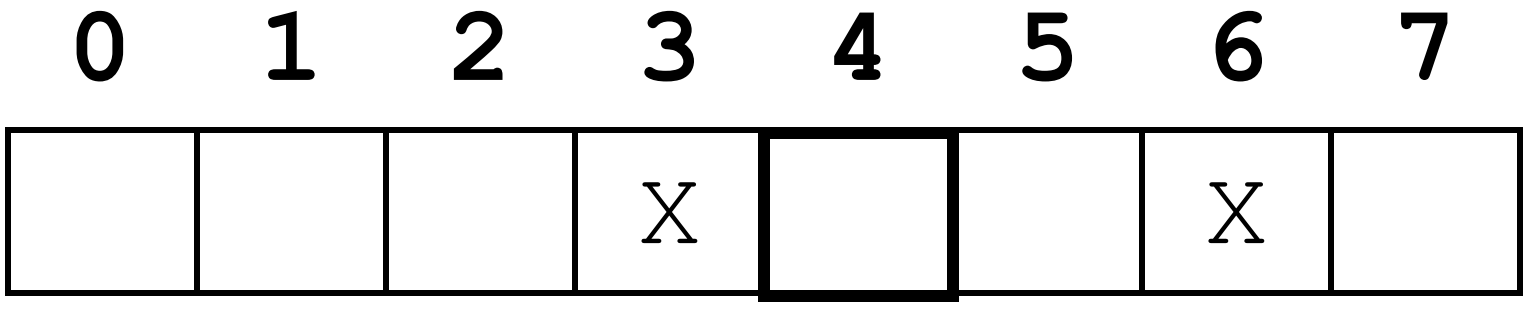


[Turn over]

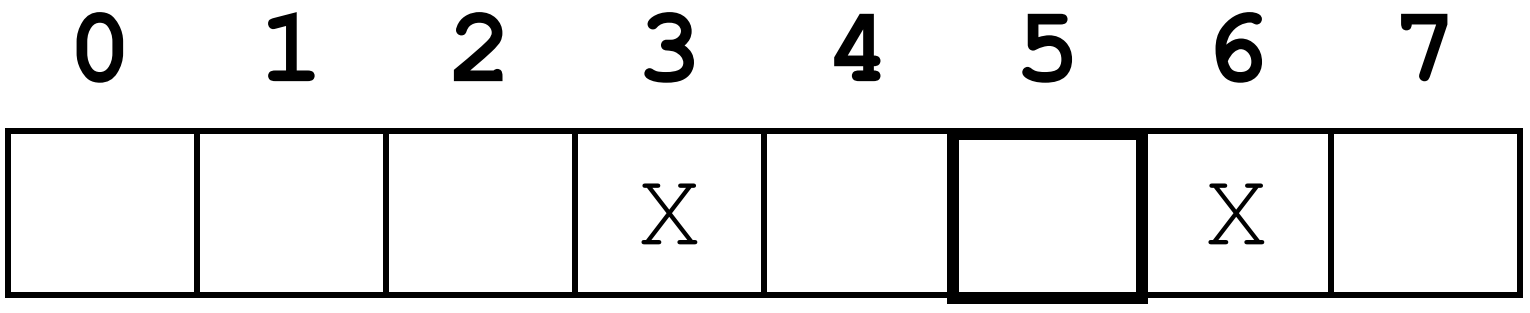
- **if the player then enters a** `2`**, they move to position** `2`



- **if the player then enters a** `2`**, they move to position** `4`



- **if the player then enters a** `1`**, they move to position** `5`



- **if the player then enters a `2`, the game finishes.**

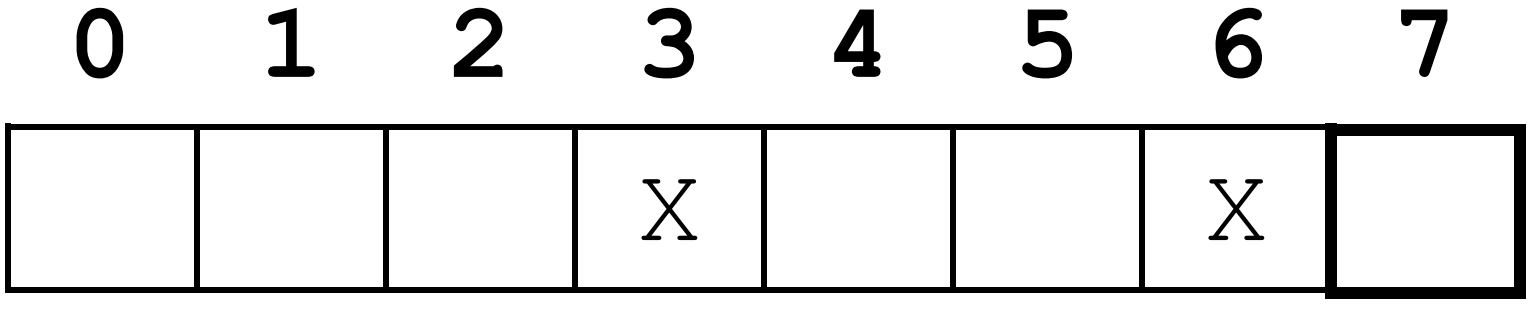


FIGURE 22, provided on page 31 of the Diagram Booklet, shows part of a Python program that will be used for the game.

`pos` **is a variable that contains the player's current position.**

Extend the program from FIGURE 22 so that the game works as described in FIGURE 21, provided pages 30–31 of the Diagram Booklet.

[Turn over]

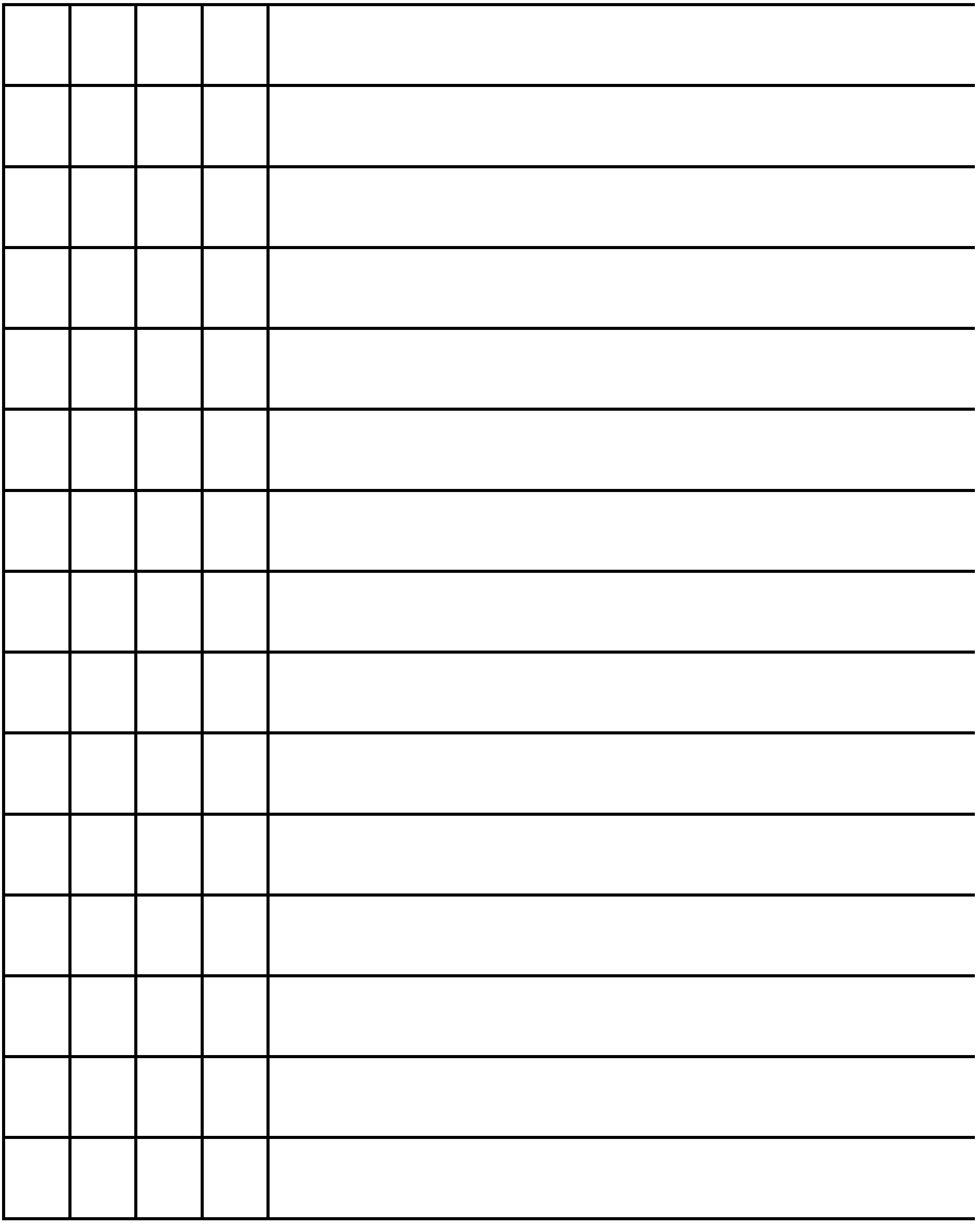
When writing your program you should assume:

- **there is a list called** `row`
- **the number of** `X` **characters in** `row` **can vary**
- **the position of the** `X` **characters in** `row` **can vary**
- **the** `X` **characters have already been added to the list called** `row`
- **the** `row` **list can be of any length.**

You SHOULD use indentation as appropriate, meaningful variable name(s) and Python syntax in your answer.

The answer grid, on pages 93–96, contains vertical lines to help you indent your code. [8 marks]

```
pos = 0
lastPos = len(row) - 1
while pos < lastPos:
```

[Turn over]

END OF QUESTIONS

8

Additional page, if required.

Write the question numbers in the left-hand margin.

Additional page, if required.

Write the question numbers in the left-hand margin.

Additional page, if required.

Write the question numbers in the left-hand margin.

100

BLANK PAGE

For Examiner's Use	
Question	Mark
1	
2–3	
4–5	
6–7	
8–9	
10–11	
12	
13–14	
15	
TOTAL	