

Please check the examination details below before entering your candidate information

Candidate surname					Other names				
Centre Number					Candidate Number				
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

Pearson Edexcel Level 1/Level 2 GCSE (9–1)

Specimen Assessment Material for first teaching Xx 2024

Morning/Afternoon (Time: 2 hours)	Paper reference	1CP2/02
-----------------------------------	-----------------	----------------

Computer Science
PAPER 2: Application of Computational Thinking

You must have: a computer workstation with appropriate programming language code editing software and tools, including an IDE that you are familiar with which shows line numbers, a 'STUDENT CODING' folder containing code and data files, printed and electronic copies of the Programming Language Subset (PLS) document

Total Marks

Instructions

- Answer all the questions on your computer.
- Save new or amended code using the file name provided and place it in the 'COMPLETED CODING' folder.
- You must **not** use the internet at any time during the examination.

Information

- The 'STUDENT CODING' folder in your user area includes all the code and data files you need.
- The total mark for this paper is 75.
- The marks for each question are shown in brackets.

Advice

- Read each question carefully before you start to answer it.
- Try to answer every question.
- Check your answers if you have time at the end.

Turn over ►

S83038A

©2024 Pearson Education Ltd.
1/1/



Answer ALL questions.

Suggested time: 10 minutes

1 A program is being developed to show the pattern of traffic light colours.

The pattern is Red, Red and Amber, Green, Amber.

The program displays the full pattern three times.

Open file **Q01.py**

Amend the lines at the bottom of the code to give the:

- name of a library used in this program
- name of an array used in this program
- line number of a variable initialisation
- line number of a repetition
- name of a data type conversion function used in this program
- name of a built-in subprogram used on line 21
- name of an arithmetic operator used in this program.

Do **not** add any additional functionality.

Save your amended code file as **Q01FINISHED.py**

(Total for Question 1 = 7 marks)

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

BLANK PAGE



S 8 3 0 3 8 A 0 3 1 6

Suggested time: 10 minutes

- 2 A program is required to determine if years are leap years or not leap years. First, the user enters the number of years to be checked. Then, the program prompts the user to enter a year. Once the user enters a year, the program displays if that year is a leap year or not a leap year. The program then loops for the number of years entered.

When a year is divisible by 400 and divisible by 100, it is a leap year.

When a year is divisible by 4 and not divisible by 100, it is a leap year.

All other years are not leap years.

This tables shows the correct result for test data.

Year	Output
2000	Is a leap year
1920	Is a leap year
1900	Is not a leap year

Open file **Q02.py**

Amend the code to:

- fix the syntax error on original line
`numYears =`
- fix the syntax error on original line 11
`numYears = float (input ("How many years do you want? "))`
- fix the syntax error on original line 20
`else`
- fix the TypeError on original line 11
`numYears = float (input ("How many years do you want? "))`
- fix the NameError on original line 13
`for count in rang (numYears):`
- fix the NameError on original line 14
`theyear = int (input ("Enter a year: "))`
- fix the NameError on original line 23
`print (Goodbye)`

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



- fix the logic error on original line 23 that causes goodbye to be displayed for every year entered
`print (Goodbye)`
- fix the logic error on original line 21 that causes the year 0 to be displayed when 1920 is entered
`print (theYear % 4, "is not a leap year")`
- fix the logic error on original line 18 that causes 1920 to be reported as not a leap year
`elif (theYear % 4 == 0) and (theYear % 100 == 0):`
- fix the logic error on original line 16 that causes 1900 to be reported as a leap year
`if (theYear % 400 == 0) or (theYear % 100 == 0):`

Execute the code with the provided test data.

Do **not** change the functionality of the given lines of code.

Do **not** add any additional functionality.

Save your amended code file as **Q02FINISHED.py**

(Total for Question 2 = 11 marks)



S 8 3 0 3 8 A 0 5 1 6

Suggested time: 15 minutes

3 A program is being developed to convert imperial measurements to metric measurements.

The program displays a menu for the user. The user enters a choice from the menu.

Here are the functions for each menu item.

Menu item	Function
1	Prompts the user to enter the number of miles. The program converts the number of miles to kilometres and displays the result.
2	A message is displayed that the option is not implemented.
9	The program displays a goodbye message and ends.

When the user enters a number not on the menu, a message is displayed asking them to choose from the menu.

The program must display this output for inputs of 7, 2, 1, 11.5, 9

```
=====
1 - Miles to kilometres
2 - Inches to centimetres
9 - Exit
Enter an option: 7
Enter an option from the menu
Enter an option: 2
Going to change inches to centimetres
=====
1 - Miles to kilometres
2 - Inches to centimetres
9 - Exit
Enter an option: 1
Going to change miles to kilometres
Enter the number of miles: 11.5
11.5 miles is 18.5069 kilometres
=====
1 - Miles to kilometres
2 - Inches to centimetres
9 - Exit
Enter an option: 9
Goodbye
```

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



Open file **Q03.py**

Amend the code to make the program work and produce the correct output.

You will need to:

- add some lines of code
- amend some lines of code
- choose between alternative lines of code. Make a choice by removing the # at the beginning of the line you choose to execute.

Do **not** change the functionality of the given lines of code.

Do **not** add any additional functionality.

Save your amended code as **Q03FINISHED.py**

(Total for Question 3 = 15 marks)



Suggested time: 25 minutes

4 A program is required to process data about the stock of ice creams available for sale.

The ice cream data is stored in a comma-separated value text file named Q04_INPUT.TXT

Each record in the file is composed of the flavour, the number in stock and the selling price.

The program must meet these requirements:

- read each line from the data file
- display the flavour, the number in stock and the selling price for each item
- calculate the total value of all the stock
- display the total value of all the stock.

Here is the output produced by a fully functional program.

```
Vanilla 22 2.65
Chocolate 15 2.85
Strawberry 18 2.55
Mint 20 2.75
Caramel 17 2.35
Total 241.90
```

Open file **Q04.py**

Amend the code to meet the requirements.

Use the constants and variables provided.

Do **not** add any additional functionality.

Save your amended code as **Q04FINISHED.py**

(Total for Question 4 = 12 marks)



DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

BLANK PAGE



S 8 3 0 3 8 A 0 9 1 6

Suggested time: 30 minutes

5 A program is needed to display quarterly and annual sales data.

Here is the output produced by a fully functional program.

Name	ID	Q 1	Q 2	Q 3	Q 4	Yr Total
Adams	434	942.45	865.78	973.25	535.84	3317.32
Baker	161	1566.99	337.10	466.54	919.83	3290.46
Collins	427	924.59	1597.64	288.54	661.63	3472.40
Dalton	285	197.71	171.13	979.36	852.07	2200.27
East	460	764.20	552.8	2780.42	315.02	4412.53
Ford	889	279.43	495.23	898.52	120.78	1793.96
Green	275	867.03	637.09	522.45	2748.53	4775.10
Hill	789	880.43	469.96	979.11	755.71	3085.21
Year Total						26347.25
Mean Sales						3293.41

The pipe symbol (|) separates columns.

The program must use string formatting functions to control the output.

The table shows the format for the column headers.

Field	Align	Width	Precision	Type
Name	Left	9		String
ID	Centre	5		String
Q 1 to Q 4	Centre	9		String
Yr Total	Centre	10		String



The table shows the format for each row of employee data.

Field	Align	Width	Precision	Type
Name	Left	9		String
ID	Centre	5		Integer
Q 1 to Q 4	Right	9	2	Real
Yr Total	Right	10	2	Real

The data is stored in one-dimensional arrays.

Open file **Q05.py**

Amend the code to:

- calculate the total sales for each employee and add it to an array
- display the column headers
- display the data for each employee
- calculate and display the total sales for the year
- calculate and display the mean sales for the year.

The program must work if the number of items in the arrays is changed.

Use the constants and variables provided.

Use string formatting functions to align the columns.

Do **not** add any additional functionality.

Save your amended code file as **Q05FINISHED.py**

(Total for Question 5 = 15 marks)



S 8 3 0 3 8 A 0 1 1 1 6

Suggested time: 30 minutes

6 A program is being developed to create a noughts and crosses game. The game grid has nine cells. The user symbols are \times and \bigcirc . A blank cell has an underscore ($_$) symbol in it.

The game functions are being developed by different team members. One part of the development is to find if there is a winner.

A winning symbol is one that appears in three horizontal, vertical or diagonal cells in a line.

The table shows example grids and winners.

× wins	○ wins	× wins	No winner																																				
<table><tr><td>—</td><td>○</td><td>○</td></tr><tr><td>×</td><td>×</td><td>×</td></tr><tr><td>—</td><td>—</td><td>—</td></tr></table>	—	○	○	×	×	×	—	—	—	<table><tr><td>○</td><td>×</td><td>—</td></tr><tr><td>○</td><td>—</td><td>×</td></tr><tr><td>○</td><td>—</td><td>—</td></tr></table>	○	×	—	○	—	×	○	—	—	<table><tr><td>×</td><td>—</td><td>○</td></tr><tr><td>○</td><td>×</td><td>—</td></tr><tr><td>—</td><td>—</td><td>×</td></tr></table>	×	—	○	○	×	—	—	—	×	<table><tr><td>—</td><td>×</td><td>○</td></tr><tr><td>×</td><td>—</td><td>—</td></tr><tr><td>○</td><td>—</td><td>—</td></tr></table>	—	×	○	×	—	—	○	—	—
—	○	○																																					
×	×	×																																					
—	—	—																																					
○	×	—																																					
○	—	×																																					
○	—	—																																					
×	—	○																																					
○	×	—																																					
—	—	×																																					
—	×	○																																					
×	—	—																																					
○	—	—																																					

Open file **Q06.py**

Write a program to meet these requirements:

- process each of the game grids given in the code
- display each game grid by row, no grid lines required
- determine if there is a winner for each game
- display the winning symbol and an appropriate message when there is a winner
- display an appropriate message when there is no winner
- include at least one user-devised subprogram.

Use comments, white space and layout to make the program easier to read and understand.

Do **not** add any additional functionality.

Save your amended code as **Q06FINISHED.py**

(Total for Question 6 = 15 marks)

TOTAL FOR PAPER = 75 MARKS



DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

BLANK PAGE



S 8 3 0 3 8 A 0 1 3 1 6



DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

BLANK PAGE



DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

BLANK PAGE



S 8 3 0 3 8 A 0 1 5 1 6

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

BLANK PAGE

