Computer Science (J277)

J277/02, Computational thinking, algorithms and programming, Practice - 2023
Nadya French

Please note that you may see slight differences between this paper and the original.

Candidates answer on the Question paper.

OCR supplied materials:
Additional resources may be supplied with this paper.

Other materials required:
- Pencil
- Ruler (cm/mm)

Duration: 90 mins

INSTRUCTIONS TO CANDIDATES

- Write your name, centre number and candidate number in the boxes above. Please write clearly and in capital letters.
- Use black ink. HB pencil may be used for graphs and diagrams only.
- Answer **all** the questions, unless your teacher tells you otherwise.
- Read each question carefully. Make sure you know what you have to do before starting your answer.
- Where space is provided below the question, please write your answer there.
- You may use additional paper, or a specific Answer sheet if one is provided, but you must clearly show your candidate number, centre number and question number(s).

INFORMATION FOR CANDIDATES

- The quality of written communication is assessed in questions marked with either a pencil or an asterisk. In History and Geography a *Quality of extended response* question is marked with an asterisk, while a pencil is used for questions in which *Spelling, punctuation and grammar and the use of specialist terminology* is assessed.
- The number of marks is given in brackets **[ ]** at the end of each question or part question.
- The total number of marks for this paper is **80**.
- The total number of marks may take into account some 'either/or' question choices.

1 Tick (✓) **one** box in each row to identify if each operator is a comparison operator or an arithmetic operator.

Operator	Comparison	Arithmetic
==		
+		
DIV		
>		

[4]

2(a) Give the output of the algorithm.

```
temp = 7
if temp < 10 then
       print(temp)
endif
```

[1]

(b)
```
value = 3
value = value - 1
print(value * 2)
```

[1]

(c)
```
scores = [3, 6, 6, 9, 2, 8]
number = scores[2]
print(number)
```

[1]

(d)
```
day = "Monday"
x = day.length
print(x)
```

[1]

3(a) An array stores a collection of words. The array has the identifier `data`.

```
data = ["or", "and", "it", "when", "and", "or", "and", "it"]
```

A searching algorithm is used to find a word in the array `data`. The algorithm used to search for a word is shown.

```
word = input("Enter a word to find")
found = False
for i = 0 to 7
  if data[i] == word then
    found = True
  endif
next i
print(found)
```

State the name of this searching algorithm.

---

[1]

(b) Some words appear in the array more than once.

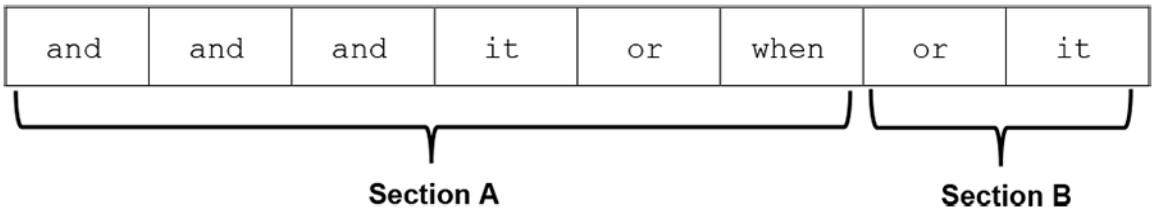
Refine the algorithm to:

- count the number of times a word appears in the array
- output this number.

For example, the algorithm should output 3 if the input is `"and"`.

[4]

(c) An insertion sort is used to put the array with the identifier `data` into alphabetical order.
The diagram shows the array when the insertion sort is almost complete.



i. Explain why the array is shown with two sections.

---

[2]

ii. Describe the steps an insertion sort takes to complete sorting the array `data`.

---

[3]

(d) One sorting algorithm splits an array into single items and then combines them into an ordered array.

State the name of this sorting algorithm.

---

[1]

4 Complete the description of computational thinking using the given list of terms.

Not all terms will be used.

abstraction	algorithm	computation	decomposition
evaluation	flowchart	origin	program
pseudocode	research	sequence	thinking

Computational thinking is the process of analysing problems so that they can be solved in a logical way.

The process of …… breaks down a problem into smaller, more manageable parts.

The process of …….. removes unnecessary detail from the problem, so that the main components can be focused on.

Algorithmic ……. identifies the main steps needed to solve the problem and the ……. that the steps are completed.

[4]

5(a) A programmer uses a high-level language and an Integrated Development Environment (IDE) to create a computer game.

Describe **two** advantages to the programmer of using a high-level language instead of a low-level language.

1 ----------

---

2 ----------

---

[4]

(b) The IDE includes a translator, such as a compiler or an interpreter.

Tick (✓) **one** box in each row to identify whether each statement describes the use of a compiler, an interpreter, or both.

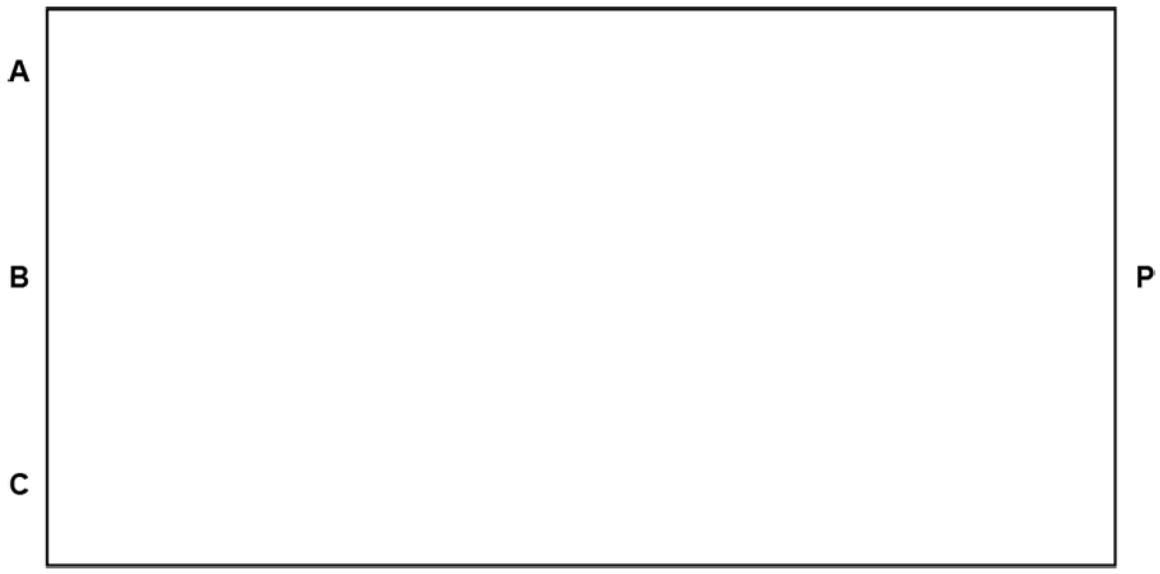
Statement	Compiler	Interpreter	Both
Translates high-level code to low-level instructions.			
Produces an executable file.			
Program needs to be translated every time it is run.			

[3]

6(a) A computer game uses a controller with three inputs, **A**, **B** and **C**.

An action **P** is carried out if the player presses **A** with either **B** or **C** at the same time.

Draw the logic system for **P**.



[3]

(b) Complete the truth table for **P = NOT A OR (B AND C)**.

A	B	C	P
0	0	0	
0	0	1	
0	1	0	
0	1	1	
	0	0	
	0	1	
	1	0	
	1	1	

[3]

(c) In a computer game, a player collects items. Each item has a score. The score for each item is stored using the data type **real**.

State what is meant by the data type **real**.

---

[1]

(d) The result of a computer game can be a win, a loss or a draw. The result is stored in a variable with the identifier `result`.

i. State why `result` **cannot** be stored using a Boolean data type.

---

[1]

ii. Give a suitable data type for `result`. Explain your answer, showing how each result could be stored.

Data type ---

Explanation ---

---

[2]

(e) A player enters a code at the beginning of a computer game. The code is three characters long. The code sets the level of the game. The game has three levels.

A variable `level` is assigned a value based on the code entered.

i. Write an algorithm that:

- asks the user to enter a three-character code
- repeats the previous step if the code is not 3 characters long
- if the code entered is SVA, assigns the value 2 to `level`
- if the code entered is UTV, assigns the value 3 to `level`
- assigns the value 1 to `level` if any other three-character code is entered.

[6]

ii. When a player completes a level, they move to the next level.

Once level 3 is completed, the game starts again at level 1.

Write a function called `nextlevel()` that:

- takes the current level as a parameter
- works out the integer value of the next level the player moves to

- returns the integer value.

[3]

iii. Complete the line of code to output the value returned from the function `nextlevel()` when the integer value 3 is passed in as a parameter.

```
print(……..……...…..……..…..(……..…....……..….))
```

[1]

7(a) OCR Drones flies goods around the country using drones.

Details about the drones that pilots fly are stored in a database table called `TblDrone`.
Some of the data stored in this table is shown.

DroneID	DroneType	Mileage	LastCheck
001	Quadcopter	65 032	65 000
002	Quadcopter	32 128	21 000
003	Octocopter	98 021	98 000

TblDrone

i. Complete the SQL statement to display `DroneID` and `Mileage` for all Octocopter type drones that have a mileage of greater than 50 000 miles.

```
SELECT ……

…… TblDrone

WHERE DroneType = "Octocopter" ……..…… Mileage ……..……..……
```

[4]

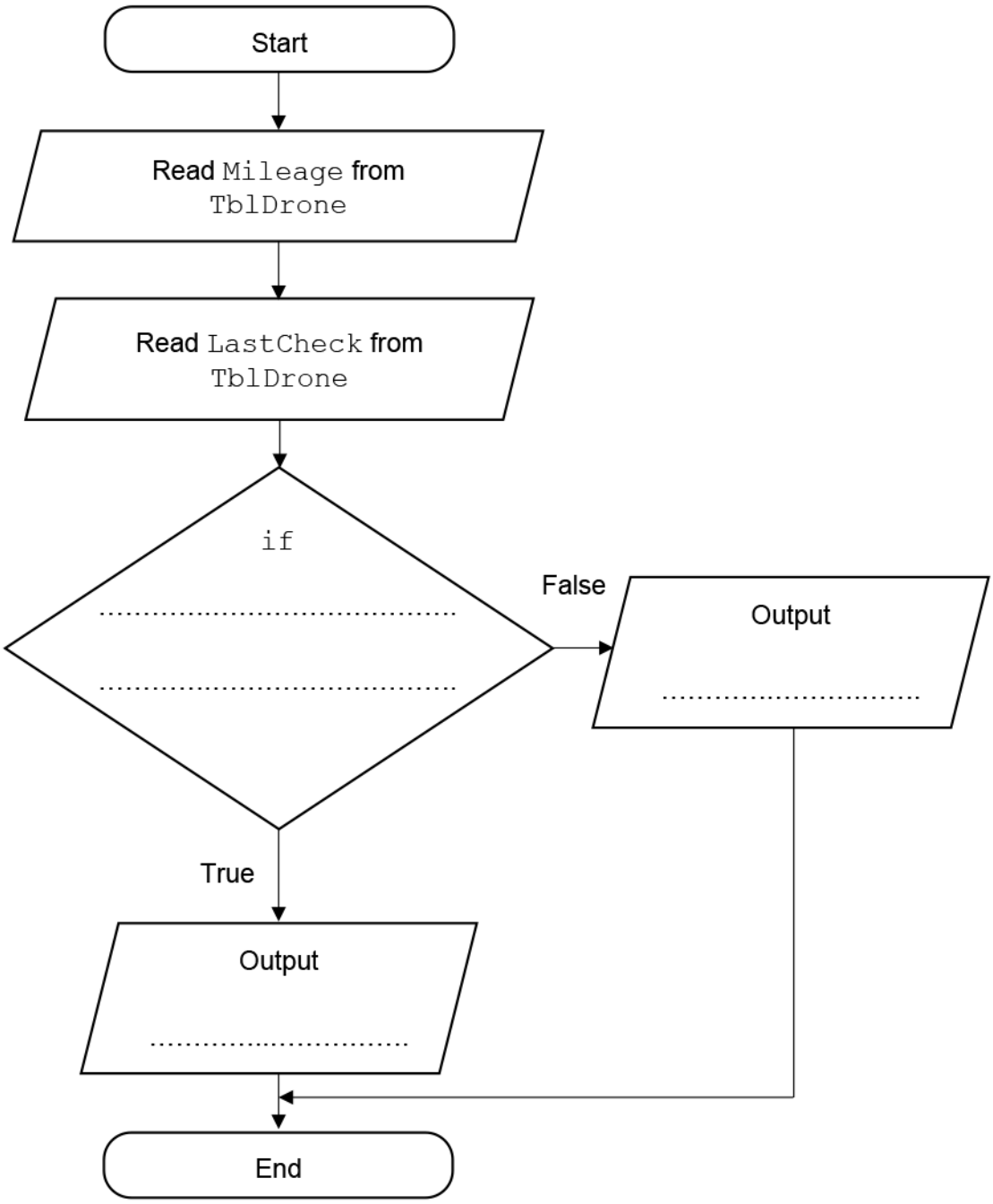
ii. Drones must be checked every 10 000 miles. If the difference between `Mileage` and `LastCheck` is greater than 10 000 then the drone needs to be checked.

A flowchart shows the steps needed to check a drone.

The flowchart outputs `"Check"` if the drone needs to be checked.

The flowchart outputs `"No Check"` if the drone does **not** need to be checked.

Complete the flowchart for the algorithm.



[2]

(b) A pilot code is automatically generated when a new pilot joins the company.

This algorithm generates a code for each pilot:

```
01 a = input("Enter first letter of first name")
02 b = input("Enter first letter of second name")
03 c = random(1,100)
04 while c < 100
05 c = c * 10
06 endwhile
07 pilotCode = a + b + str(c)
08 print(pilotCode)
```

Complete the trace table for the given algorithm.

Lines 01 to 03 have already been completed.

You may not need to use all rows in the trace table.

Line number	a	b	c	pilotCode	Output
01	H				
02		K			
03			9		

[4]

(c) A pilot's flying experience is validated. An algorithm checks that the experience is between 0 and 20 years.

```
exp = input("Enter number of years")
if exp >= 0 and exp <= 20 then
    print(True)
else
    print(False)
endif
```

Complete this test plan for the algorithm.

Experience in years	Type of test	Expected output
	Normal	True
20	Boundary	
32		

[4]

(d) Pilots are paid a set amount each day. Pilots also get an additional payment for each mile they have flown that day. These payments are shown in the table.

Flying experience	Pay per day	Pay per mile
Fewer than 2 years	£120.00	£0.45
2 years to 5 years inclusive	£150.00	£0.65
More than 5 years	£180.00	£0.85

For example, a pilot with 3 years' experience who flies 100 miles in one day will receive a total of £215.00 pay. This is calculated in the following way:

- £150.00 pay for the day
- £65.00 additional payment (£0.65 pay per mile x 100 miles)
- Add £150.00 and £65.00 together to get £215.00 total pay

i. Complete the algorithm to:

- Calculate the total pay for the pilot for that day

You must use either:

- OCR Exam Reference Language, or
- A high-level programming language that you have studied.

```
experience = input("Enter years of experience")

miles = input("Enter miles flown")

totalPay = 0

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

print(totalPay)
```

[4]

ii. The programmer decides to make a function to calculate the total pay for the pilot.
The function is called `calculatePay()`.

The function takes the values for `experience` and `miles` as two parameters and returns the total pay for the pilot.

Refine the algorithm to use this function and output the pay for the pilot.

You must use either:

- OCR Exam Reference Language, or
- A high-level programming language that you have studied.

```
experience = input("Enter years of experience")

miles = input("Enter miles flown")


…… = calculatePay( …… , ……)


print( totalPay )
```

[2]

(e) A text file is used to store pilot's codes and date of birth.

The text file is named `pilots.txt`. The contents of some of this file is shown:

```
JP670,17/05/1986
CC200,31/06/1980
SM720,01/04/1976
GC730,06/06/1978
```

Design an algorithm to:

- Ask the user to input a pilot code and a date of birth
- Write these inputs to the `pilots.txt` text file

[4]

(f) A pilot is allowed to fly a maximum of 9 hours per day. The pilot code and the hours flown for each flight in one day are stored in a 2D array of strings with the identifier `journeys`.

The data in `journeys` is shown.

	[0]	[1]
[0]	"SM720"	"4.5"
[1]	"GC730"	"3"
[2]	"JP670"	"2"
[3]	"GC730"	"3.5"
[4]	"CC200"	"9"
[5]	"RY320"	"12"

The value of `journeys[3,1]` is `"3.5"`.

Create a function, `pilotValid()`, that:

- takes a pilot code as a parameter
- uses the 2D array `journeys` to calculate the number of hours that the pilot has flown
- returns `valid` if the pilot has flown 9 hours or fewer or `warning` if the pilot has flown for more than 9 hours
- uses casting where needed.

You must use either:

- OCR Exam Reference Language, or
- A high-level programming language that you have studied.

[6]

END OF QUESTION PAPER