CAMBRIDGE
International Education

Cambridge International AS & A Level

CANDIDATE NAME

CENTRE NUMBER

CANDIDATE NUMBER

COMPUTER SCIENCE **9618/23**

Paper 2 Fundamental Problem-solving and Programming Skills **May/June 2026**

2 hours

You must answer on the question paper.

You will need: Insert (enclosed)

INSTRUCTIONS

- Answer **all** questions.
- Use a black or dark blue pen.
- Write your name, centre number and candidate number in the boxes at the top of the page.
- Write your answer to each question in the space provided.
- Do **not** use an erasable pen. Do **not** use correction fluid or tape.
- Do **not** write on any bar codes.
- You may use an HB pencil for any diagrams, graphs or rough working.
- Calculators must **not** be used in this paper.

INFORMATION

- The total mark for this paper is 75.
- The number of marks for each question or part question is shown in brackets [ ].
- No marks will be awarded for using brand names of software packages or hardware.
- The insert contains all the resources referred to in the questions.

This document has **24** pages. Any blank pages are indicated.

DC (PQ/CGW) 356767/4

[Turn over

BLANK PAGE

Refer to the **insert** for the list of pseudocode functions and operators.

1 Study the pseudocode.

```
i ← 1
REPEAT
    OUTPUT i*i
    i ← i + 1
UNTIL i = 6
```

(a) Identify the type of programming construct used in the pseudocode.

..... [1]

(b) State how many iterations are carried out by the pseudocode.

..... [1]

2 A control program is being developed for a robot floor vacuum cleaner. The settings for the robot can be changed and include:

- the strength of the suction (numerical values: 0, 1, 2, 3, 4 or 5)
- the total vacuum cleaning time (a whole number of minutes).

The robot has a number of sensors.

The control program includes the variables:

variable	description
`MainsOn`	`TRUE` when the mains power is on
`DisplayOn`	`TRUE` when the user display is shown
`Suction`	`TRUE` when the program detects that the user has selected a non-zero value for the strength of the suction

(a) Identify the data type used for the **three** variables.

..... [1]

(b) Write a pseudocode statement to set the `DisplayOn` variable to `TRUE` when the mains power is on.

.....

.....

..... [1]

(c) When the robot is ready to move, the program assigns the value `TRUE` to a variable, `Move`

The control program contains the pseudocode logic expression:

```
Move ← MainsOn AND Suction
```

Explain the pseudocode statement.

.....

.....

..... [2]

(d) The pseudocode for the control program contains the procedure `CleanProgram()`

```
PROCEDURE CleanProgram()
// declaration of variables has been omitted

    Finished ← FALSE
    OUTPUT "Total clean time (minutes)"
    INPUT  TotalCleanTime
    OUTPUT "Suction strength (0,1,2,3,4,5)"
    INPUT  SuctionStrength

    WHILE Finished = FALSE
        // TimeSoFar is a global variable which changes
        // as the procedure is executed
        // TimeSoFar is the time the robot has been cleaning (in seconds)

        TimeRemaining ← TotalCleanTime - INT(TimeSoFar/60)

    ENDWHILE

ENDPROCEDURE
```

(i) When the program is under development and running, the programmer can temporarily stop the execution at a set point in the program.

Identify this feature of the Integrated Development Environment (IDE).

..... [1]

(ii) The programmer runs the program.

- The user is prompted and inputs the value 40 (total clean time in minutes).
- The user is prompted and inputs the value 3 (for the suction strength).
- The programmer interrupts its execution; the `TimeSoFar` is shown in a report window as 125 (the cleaning time so far in seconds).

State the current value of `TimeRemaining`

..... [1]

3 A company wishes to replace the point-of-sale software that each shop uses at its checkout. A software company is developing a new system.

(a) Complete the summary about the development of the new system.

The initial ..... stage is carried out by studying in detail the current system.

A design for the proposed new system is produced.

Following the design and coding stages, it is decided to install the completed new software initially in only two of the shops. ..... testing is carried out.

The final stage of the project uses ..... testing to test all programs on the company's hardware. After successful testing, the company approves the new system.

The company then installs the new software in all shops. In the first week of its use, unexpected bugs are reported. ..... maintenance has to be carried out.

[4]

(b) For this large project, the task was divided between two programming teams.

Team A were responsible for the screen designs and all outputs; they involved the company at the earliest possible stage and obtained feedback.

Team B used a more traditional approach to the program development life cycle; the stages in the development followed an agreed sequence.

Identify the program development life cycle model adopted by each team, and describe **one** additional feature of each approach.

team A model .....

feature .....

.....

team B model .....

feature .....

.....

[4]

(c) State **three** activities carried out during the design stage of the program development life cycle.

1 .....

.....

2 .....

.....

3 .....

.....

[3]

BLANK PAGE

4 An algorithm is expressed as:

- A user inputs a sequence of integers.
- The user will input value –1 to indicate there are no more integers to be input.
- The program will calculate the total number of integers greater than 120.
- Output the total number of integers greater than 120 after the –1 terminator is input.

Write a structured English description for this algorithm as a sequence of steps.

Do **not** include pseudocode statements in your description.

Your answer might **not** require all **six** steps shown.

step 1 .....

.....

step 2 .....

.....

step 3 .....

.....

step 4 .....

.....

step 5 .....

.....

step 6 .....

.....

[4]

5 Data is stored about people who have attended a job interview. Each interview is held at one of three office locations.

Each person's data is stored as a data packet made up of four data items:

<initial><shortened surname><office code><summary of the interview>

- The initial is a single character.
- The shortened surname is the first five characters of the surname.
  - You can assume all surnames are at least five characters.
- The office code is a single character 'A', 'B' or 'C'.
- The interview summary text is of variable length.

Example:

Peter Watkins was interviewed at office 'B'. After the interview it was decided to invite him for a second interview.

His data packet is stored by variable `Applicant`:

```
Applicant ← "PWatkiBSecond interview"
```

(a) A pseudocode function `InterviewSummary()` will:

- take the data packet as a single parameter
- return the interview summary text.

Write the pseudocode for the function `InterviewSummary()`

.....

.....

.....

.....

.....

..... [4]

(b) The data for all interviews is saved in this format in a text file `JobApps.txt`

Each line of the file stores one data packet.

A procedure `Search()` is written to search for a particular applicant.

If the applicant is found, the program will output 'Name found', otherwise it will output 'Name not found'.

Complete the pseudocode for procedure `Search()`

```
PROCEDURE Search(Initial : CHAR, ShortSurname : STRING)
    DECLARE ThisApplicant, ThisSurname : STRING
    DECLARE ThisInitial : CHAR
    DECLARE Found : BOOLEAN

    OPENFILE "JobApps.txt" .....
    Found ← FALSE
    REPEAT

        ..... "JobApps.txt", ThisApplicant
        ThisInitial ← LEFT(ThisApplicant, 1)

        ThisSurname ← .....

        IF ThisSurname = ShortSurname ..... THEN

            .....
            OUTPUT "Name found"
        ENDIF

    UNTIL Found = TRUE .....

    IF Found = FALSE THEN
        OUTPUT "Name not found"
    ENDIF

    .....
ENDPROCEDURE
```

[7]

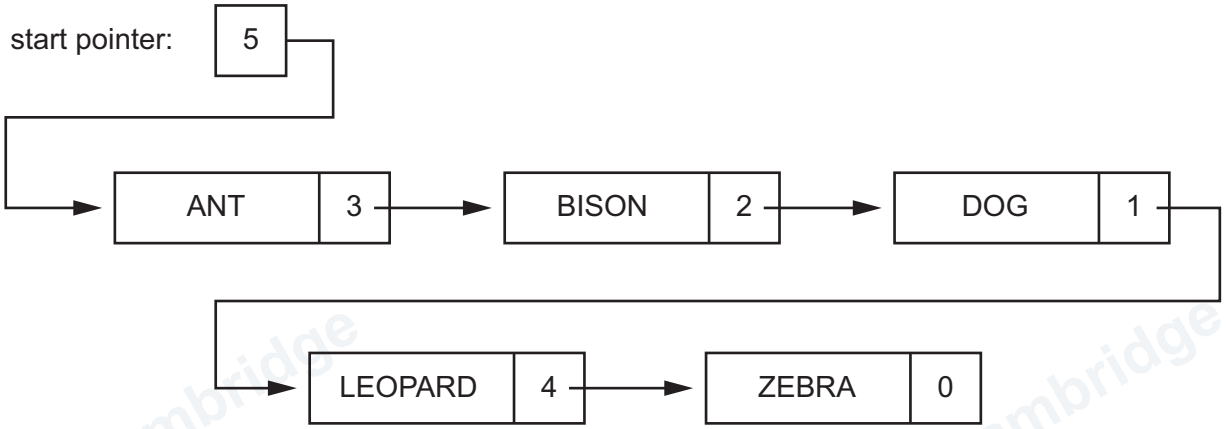
6 A linked list is an example of an Abstract Data Type (ADT).

A linked list is implemented using:

- a 1D array for the data items
- a 1D array for the link pointers
- a start pointer.

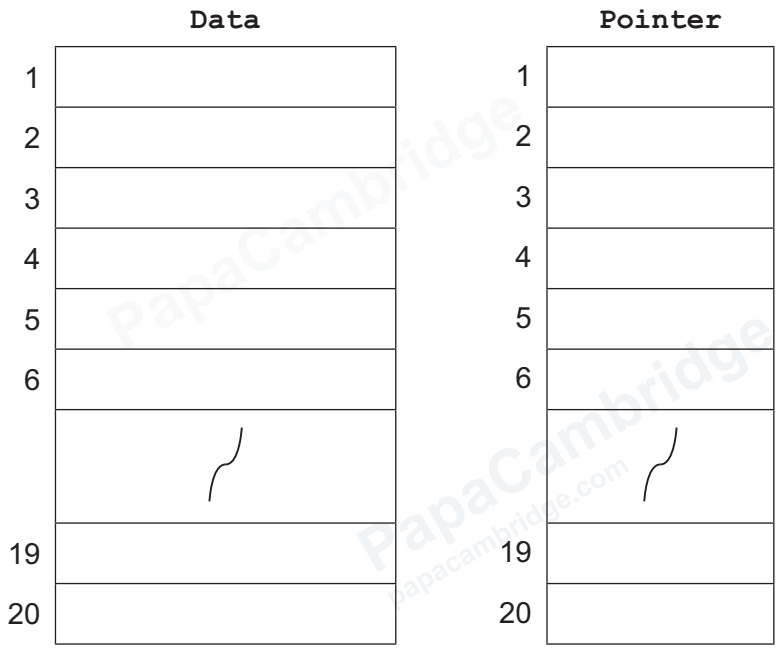
The linked list maintains the data items in alphabetical order.

(a) The state of the linked list after five data items have been added is:



Complete the diagram to show the `StartPointer` and the contents of the arrays `Data` and `Pointer`

`StartPointer` .....



[3]

(b) State what the pointer value 0 (null pointer) indicates.

..... [1]

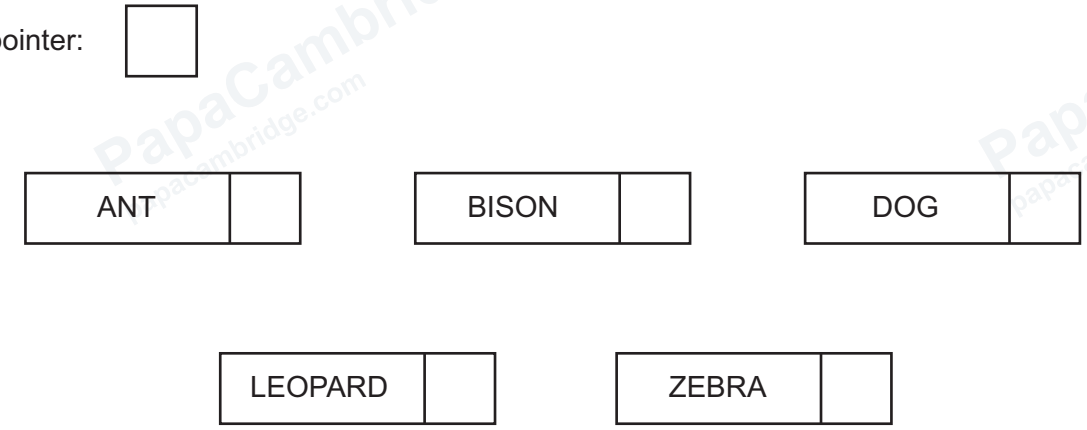
(c) To maintain the linked list, algorithms are required to:

- insert a new data item into the linked list
- delete a data item from the linked list.

(i) The data item 'DOG' is to be deleted from the linked list given in **6(a)**.

Complete the diagram to show the new state of the linked list.

start pointer:



[1]

(ii) To delete a data item, the algorithm will have special cases. One special case is the linked list is empty.

Describe **two** other special cases which must be considered when deleting a data item.

1 .....

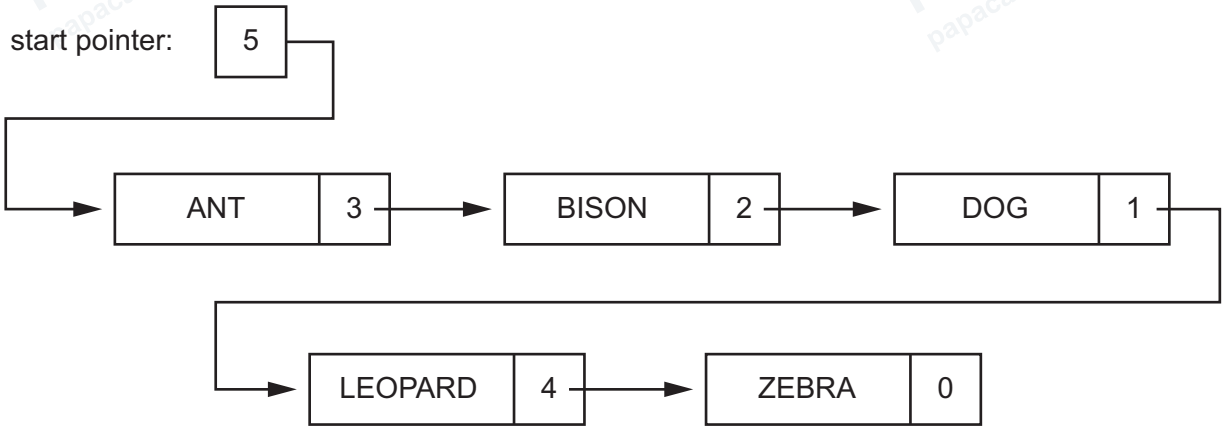
.....

2 .....

.....

[2]

(d) The linked list given in **6(a)** is repeated here. Data item 'BAT' is to be added to the linked list.



Describe how the data item 'BAT' is added to the linked list.

Do **not** attempt to write pseudocode.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

7 Study the pseudocode algorithm. Numbers will be copied into the `Data` array.

```
// STAGE 1
Num1 ← 12
Index ← 1
FOR i ← 2 TO Num1 DIV 2
    IF Num1 MOD i = 0 THEN
        Data[Index] ← i
        Index ← Index + 1
    ENDIF
NEXT i
Index ← Index - 1
OUTPUT Index, " values were found"
```

(a) (i) Complete the trace table for `STAGE 1`

			Data					
Num1	**Index**	**i**	1	2	3	4	5	6

[4]

(ii) State how the numbers in the `Data` array are related.

..... [1]

(b) The complete algorithm has a second stage.

```
// STAGE 2

Num2 ← 8
Match ← FALSE

REPEAT
    IF Num2 MOD Data[Index] = 0 THEN
        Match ← TRUE
    ELSE
        Index ← Index - 1
    ENDIF
UNTIL Match = TRUE OR Index = 0

IF Match = TRUE THEN
    OUTPUT Data[Index]
ELSE
    OUTPUT "Nothing found"
ENDIF
```

One line of the `STAGE 1` code is changed:

```
Num1 ← 20
```

The programmer creates a final program with the `STAGE 1` code followed immediately by the `STAGE 2` code.

The `STAGE 1` trace table shows the relevant final variable values after the completion of `STAGE 1`

STAGE 1				Data					
Num1	**Num2**	**Index**	**Match**	1	2	3	4	5	6
20		4		2	4	5	10		

This is followed by running of the `STAGE 2` code.

(i) Complete the trace table for `STAGE 2`

STAGE 2				Data					
Num1	**Num2**	**Index**	**Match**	1	2	3	4	5	6
20		4		2	4	5	10		

[3]

(ii) State the purpose of the `STAGE 2` algorithm.

.....

..... [1]

8 A warehouse stores stock in bins. A bin is a container which stores the stock items; each bin can store many identical items. The warehouse is organised on three different levels – level 0 (ground level), then level 1 and level 2. Each level has 120 bins, numbered 1 to 120.

A bin is identified by the level and a bin number. The data item stored for each bin is a Boolean value; this indicates whether the bin is in use (contains stock items) or not in use (contains no stock items).

(a) (i) Complete the pseudocode to declare a 2D array `Bins`

```
DECLARE .....
```

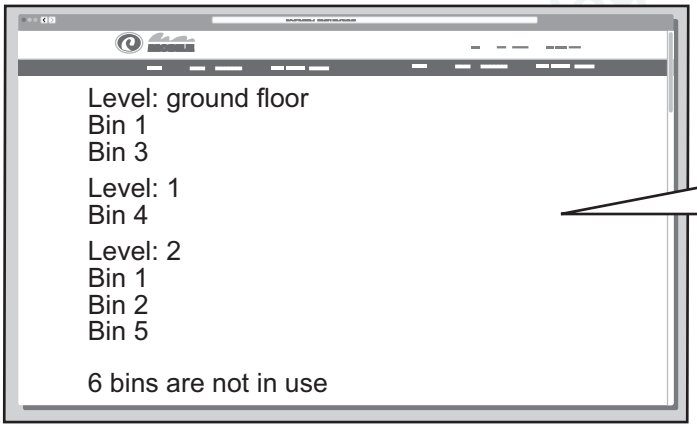
[2]

(ii) A pseudocode module `InactiveReport()` will produce a report that will show:

- a header for each level
- the bin number for each bin that is **not** in use on that level
- the total number of bins that are **not** in use on all levels.

The module is initially developed for only **five** bins on each level.

An example program output:



Shows:

- the header for each level
- a list of bin numbers for bins that are not in use on that level
- the total number of bins not in use.

Complete the pseudocode for the procedure `InactiveReport()`

Assume that the `Bins` array is a global variable and has already been populated with data.

```
PROCEDURE InactiveReport()
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

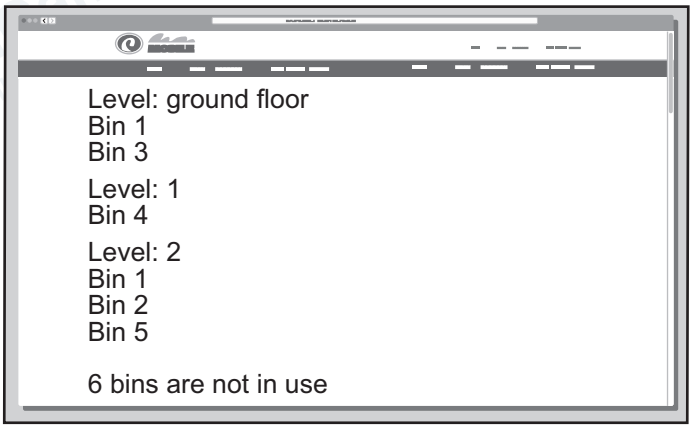
.....

.....

```
ENDPROCEDURE
```

[5]

(iii) Test data were chosen to produce this output:



Complete the pseudocode test data which would produce the output given.
Reminder: The module is initially developed for only **five** bins on each level.

```
// Ground floor
Bins[0, 1] ← FALSE
Bins[0, 2] ← TRUE

Bins[0, 3] ← .....
Bins[0, 4] ← TRUE
Bins[0, 5] ← TRUE

// Level 1
Bins[1, 1] ← TRUE

Bins[1, 2] ← .....
Bins[1, 3] ← TRUE
Bins[1, 4] ← FALSE
Bins[1, 5] ← TRUE

// Level 2
```

.....

.....

.....

.....

.....

[3]

(b) It is decided to change the 2D array `Bins` so that more information is available about each bin.

Data items for each bin are:

- bin size (coded as 'Small', 'Medium' or 'Large')
- active – is the bin currently in use?
- stock item code for the stock item stored (four-digit string)
- stock level (0 to 500 items).

(i) Complete the pseudocode to create a record structure with the identifier `BinData`

```
TYPE BinData
.....
.....
.....
.....
ENDTYPE
```

[2]

(ii) A stock item is to be allocated to an unused bin. A function `FindBin()` is required that finds and returns the location of the first bin not in use which has a suitable size. The function will search on the ground floor first, then search level 1 and finally search level 2.

The function uses the global 2D array `Bins` using the revised record structure `BinData`

The function `FindBin()`:

- takes a parameter for the bin size required (coded as 'Small', 'Medium' or 'Large')
- searches for an available bin that matches the size required
- outputs the level and bin number (as a string) of the first unused bin with the matching size
  - for example, 'Level: 0 Bin: 4'
  - or if there are no available bins 'No bins available'.

Write the pseudocode function `FindBin()`

.....

.....

.....

.....

.....

..... [7]

BLANK PAGE

