

CANDIDATE  
NAME

--

CENTRE  
NUMBER

--	--	--	--	--

CANDIDATE  
NUMBER

--	--	--	--

## Secondary – 6

Computer Science

9618/4

Paper 4 - Practical

23<sup>rd</sup>, August 2024

1 hour 30 minutes

You must answer on the question paper.

No Additional Materials are needed.

### INSTRUCTIONS

Carry out every instruction in each task.

- Save your work using the file names given in the task as and when instructed.
- You must not have access to either the internet or any email system during this examination.
- You must save your work in the evidence document as stated in the tasks. If work is not saved in the evidence document, you will not receive marks for that task.
- You must use a high-level programming language from this list:
  - Java (console mode)
  - Python (console mode)
  - Visual Basic (console mode)
- A mark of zero will be awarded if a programming language other than those listed here is used.

### INFORMATION

The total mark for this paper is 35.

The number of marks for each question or part question is shown in brackets [ ].

For Teachers use only	
Qno	Mark
1	
2 (a)	
2(b)	
2(c)	
2 (d)	
2 (e)	
Total	



Open the document evidence.doc.

Make sure that your name, centre number and candidate number will appear on every page of this

document. This document will contain your answers to each question.

**Save this evidence** document in your work area as:

evidence\_ followed by your centre number\_candidate number, for example:

evidence\_john\_0101

---

**1.**

A program stores the following ten integers in a 1D array with the identifier `arrayData`.

12    6    7    9    1    10    15    18    21    8

The following bubble sort pseudocode algorithm sorts the data in `theArray` into descending numerical order. There are **five** incomplete statements.

```
PROCEDURE bubbleSort()  
  
    DECLARE temp : INTEGER  
  
    FOR x ← 0 to .....  
  
        FOR y ← 0 to .....  
  
            IF theArray[y] ..... theArray[y + 1] THEN  
  
                temp ← theArray[y]  
  
                theArray[y] ← .....  
  
                theArray[y + 1] ← .....  
  
            ENDIF  
  
        NEXT y  
  
    NEXT x  
  
ENDPROCEDURE
```

Write program code for the procedure `bubbleSort()` to sort the data in `arrayData` into descending order.

Save your program as **question 1**

Copy and paste the program code into **part 1(a)** in the evidence document.

## 2

A program uses a circular queue to store strings. The queue is created as a 1D array, `QueueArray`, with 10 integer items. The following data is stored about the queue:

- the head pointer initialised to 0
- the tail pointer initialised to 0
- the number of items in the queue initialised to 0.

(a) Declare the array, head pointer, tail pointer and number of items.

If you are writing in python, try to include attribute declarations using comments

Save your program as **question 2**

Copy and paste the program code into **part 2(a)** in the evidence document.

[2]

(b) The function `Enqueue` is written in pseudocode. The function adds `DataToAdd` to the queue. It returns `FALSE` if the queue is full and returns `TRUE` if the item is added.

The function is incomplete, there are **five** incomplete statements.

```
FUNCTION Enqueue(BYREF QueueArray[] : STRING, BYREF HeadPointer : INTEGER,
                BYREF TailPointer : INTEGER, NumberItems : INTEGER,
                DataToAdd : STRING) RETURNS BOOLEAN

    IF NumberItems = ..... THEN

        RETURN .....

    ENDIF

    QueueArray[.....] ← DataToAdd

    IF TailPointer >= 9 THEN

        TailPointer ← .....

    ELSE

        TailPointer ← TailPointer + 1

    ENDIF

    NumberItems ← NumberItems .....

    RETURN TRUE

ENDFUNCTION
```

Write program code for the function `Enqueue()`.

Save your program

Copy and paste the program code into **part 2(b)** in the evidence document.

[7]

- (c) The function `Dequeue()` returns "FALSE" if the queue is empty, or it returns the next data item in the queue.

Write program code for the function `Dequeue()`.

Save your program

Copy and paste the program code into **part 2(c)** in the evidence document.

[8]

- (d) Amend the main program to:

- take as input 10 integer values from the user
- use the `Enqueue()` function to add each element to the queue
- output an appropriate message to state whether each addition was successful, or not
- call `Dequeue()` function twice and output the return value each time.

Save your program

Copy and paste the program code into **part 2(d)** in the evidence document.

[7]

- (e) Test your program with the input data:

10      20      13      6      5      9      1      0      879      99

Take a screenshot to show the output.

Copy and paste the screenshot code into **part 3(e)** in the evidence document.

[1]