GCE A LEVEL

1500U30-1

S19-1500U30-1

MONDAY, 3 JUNE 2019 – MORNING

COMPUTER SCIENCE – A2 unit 3
Programming and System Development

2 hours

1500U301
01

ADDITIONAL MATERIALS

A WJEC pink 16-page answer booklet.
A calculator.

INSTRUCTIONS TO CANDIDATES

Answer **all** questions.
Write your answers in the separate answer booklet provided.

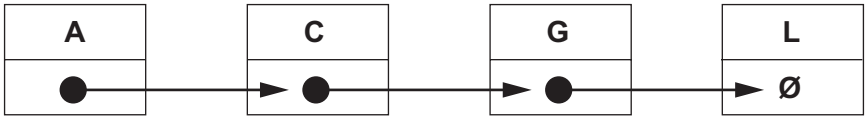
INFORMATION FOR CANDIDATES

The number of marks is given in brackets at the end of each question or part-question; you are advised to divide your time accordingly.
The total number of marks available is 100.
Assessment will take into account the quality of written communication used in your answers.

Answer **all** *questions.*

1. *(a)* Explain the differences between stack and queue data structures. [4]

(b) This is a diagram of a linked list in alphabetical order.



(i) Redraw the linked list after the data item 'E' has been added. [2]

(ii) Redraw the amended linked list after the data item 'C' has been deleted. [2]

2. Clearly showing each step, simplify the following Boolean expressions using Boolean algebra, identities and De Morgan's Law.

(a) $A.\overline{A} + A.B + A.\overline{B} + B.\overline{B}$ [5]

(b) $(\overline{A.B}) + A.C + B$ [5]

3. This is an eight-bit number:

$$01101001_2$$

Include this number in a worked example to demonstrate how masking can be used to determine the state of the most significant bit. [3]

4. *(a)* Describe the term natural language interface. [2]

(b) Using an example, describe a potential use for natural language interfaces. [2]

(c) Describe the potential problems that can be associated with natural language interfaces. [3]

5. Below is an algorithm.

```
Declare subprocedure myAlgorithm(myArray is integer, indexLow is
integer, indexHi is integer)

Declare pivot is integer
Declare tmpSwap is integer
Declare tmpLow is integer
Declare tmpHi is integer

set tmpLow = indexLow
set tmpHi = indexHi

set pivot = myArray[(int((indexLow + indexHi)/2))]

while (tmpLow <= tmpHi)

     while (myArray[tmpLow] < pivot and tmpLow < indexHi)
           set tmpLow = tmpLow + 1
     end while

     while (pivot < myArray[tmpHi] and tmpHi > indexLow)
           set tmpHi = tmpHi – 1
     end while

     if (tmpLow <= tmpHi) then
           set tmpSwap = myArray[tmpLow]
           set myArray[tmpLow] = myArray[tmpHi]
           set myArray[tmpHi] = tmpSwap
           set tmpLow = tmpLow + 1
           set tmpHi = tmpHi - 1
     end if
end while

if (indexLow < tmpHi) then myAlgorithm(myArray, indexLow, tmpHi)
if (tmpLow < indexHi) then myAlgorithm(myArray, tmpLow, indexHi)
```

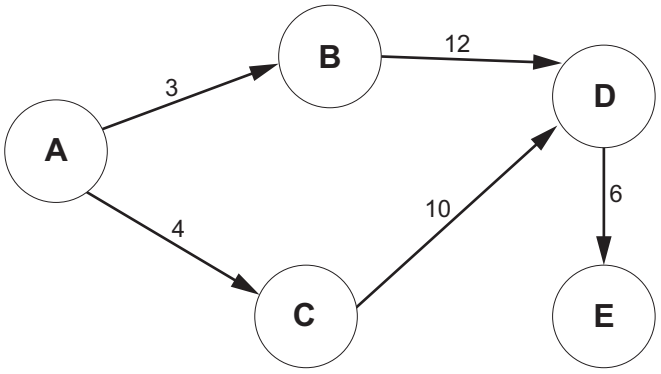
(a) Describe the purpose of this algorithm. [2]

(b) Describe the characteristics of this type of algorithm. [3]

(c) Describe the advantages arising from the elegance of this algorithm. [3]

6. *(a)* Explain the purpose of a shortest path algorithm. [4]

(b) This is a diagram of the costs of traversing a network.



The traversal cost for each node is 2.

(i) Show how this network and its traversal costs can be represented using a two dimensional array. [2]

(ii) State the shortest path from node A to E and calculate its cost. [2]

7. *(a)* Explain what is meant by the term programming paradigm. [2]

(b) Describe the difference between procedural and event-driven programming paradigms. [4]

(c) Using examples, describe which programming paradigms would be most suitable when developing different types of software applications. [4]

8. Draw a truth table to prove the following: [4]

A NOR B = NOT A AND NOT B

9. Below are two algorithms that search for a data item in a one dimensional array. You can assume that the data in the array is in ascending order and that the data item being searched is present. The Search_A algorithm has a time performance of O(n).

```
Algorithm Search_A

declare searchKey, i as integer
declare flag as Boolean
declare myArray[] as integer[]
Input searchKey
Set i = 0
flag = FALSE

repeat
      if myArray[i] = searchKey then
            set flag = TRUE
      end if
      set i = i + 1
until (flag = TRUE)
output myArray[i]
```

```
Algorithm Search_B

declare searchKey, first, last, m as integer
declare myArray[] as integer[]
Input searchKey
Set first = 1
Set last = len(myArray[])

repeat
      set m = (first + last) DIV 2
      if searchKey < myArray[m] then
            set last = m – 1
      else
            set first = m + 1
      end if
until (myArray[m] = searchKey)
output myArray[m]
```

(a) Evaluate the efficiency of the Search_B algorithm and using Big O notation, determine the growth rate for time performance. [5]

(b) Draw a graph of the algorithms above to illustrate their order of time performance. Graph paper is not required. [4]

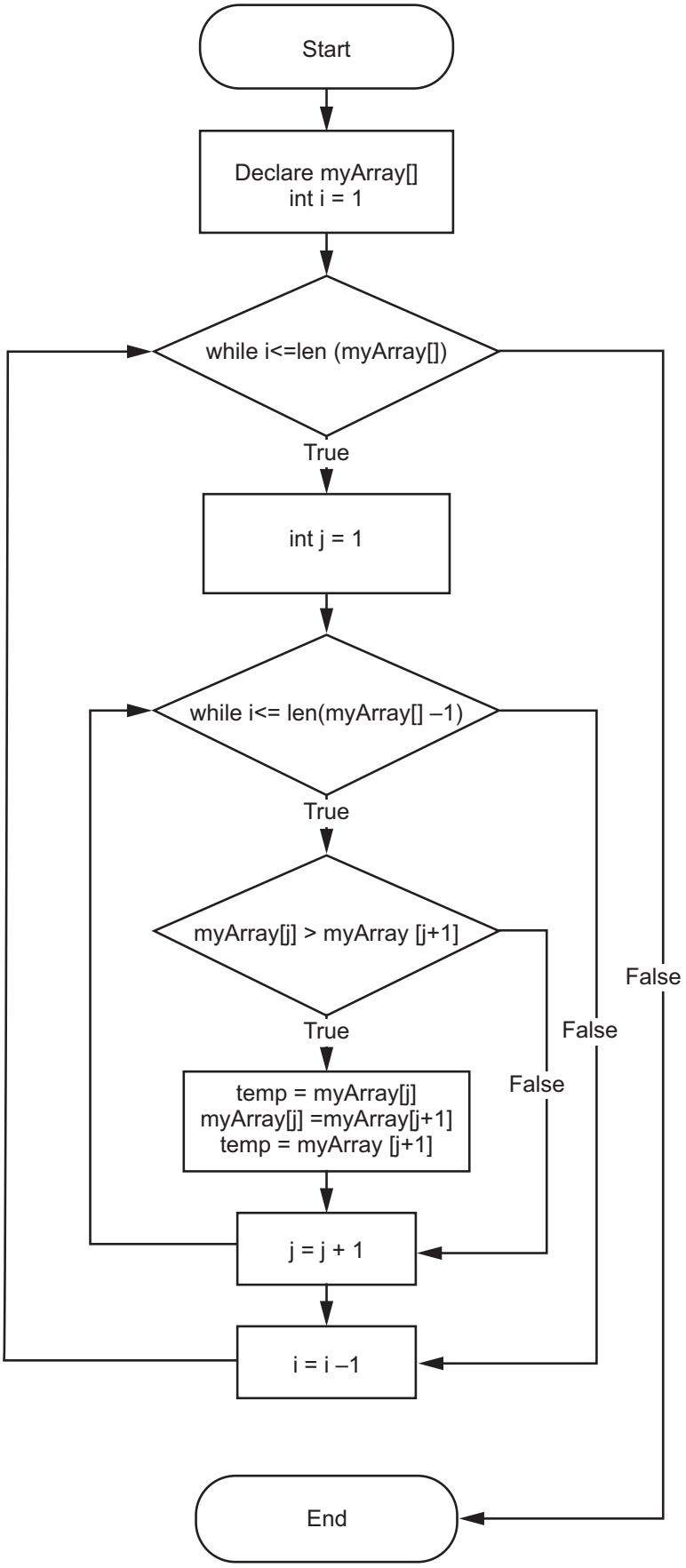
(c) State which algorithm is more efficient when searching for a data item. [1]

10. Describe the term data compression and explain how data compression algorithms are used.[6]

11. *(a)* Describe what is meant by a class and an object, and describe the relationship between them. [4]

(b) Describe what is meant by the term method, and describe the relationship between object and method. [2]

12. The algorithm in the flowchart is intended to sort an array in descending order. However, the algorithm contains several errors. Write a corrected version of the algorithm in pseudo-code. [7]



13. Describe the purpose, and give examples of, the use of compilers, interpreters and assemblers and distinguish between them. [13]

END OF PAPER