Cambridge Assessment International Education

Cambridge International AS & A Level

COMPUTER SCIENCE **9618/41**

Paper 4 Practical **October/November 2022**

2 hours 30 minutes

You will need: Candidate source files (listed on page 2)
evidence.doc

0549214835

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 75.
- The number of marks for each question or part question is shown in brackets [ ].

This document has **12** pages. Any blank pages are indicated.

© UCLES 2022

Open the evidence document, **evidence.doc**

Make sure that your name, centre number and candidate number will appear on every page of this document. This document must 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_zz999_999**

A class declaration can be used to declare a record.

If the programming language does not support arrays, a list can be used instead.

A source file is used to answer **Question 1**. The file is called **`IntegerData.txt`**

1 The text file `IntegerData.txt` stores 100 integer numbers between 1 and 100 inclusive. A program is required to read in this data and perform searching and sorting on the data.

(a) Write program code to declare a global 1D array, `DataArray`, with space for 100 integer values.

Save your program as **Question1_N22**. Copy and paste the program code into **part 1(a)** in the evidence document.

[2]

(b) The procedure `ReadFile()` must read in the numbers from the text file and store each one in the array. Use appropriate exception handling.

Write program code for the procedure `ReadFile()`.

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

[6]

(c) The function `FindValues()` asks the user to enter a number to search for in the array. The number input must be a whole number between 1 and 100 inclusive. The function then returns the number of times the number input appears in the array.

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

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

[7]

(d) (i) Write program code to call `ReadFile()` and `FindValues()` from the main program. The return value from `FindValues()` must be output with an appropriate message.

Save your program. Copy and paste the program code into **part 1(d)(i)** in the evidence document.

[3]

(ii) Test your program using the number 61 as input.

Take a screenshot to show the output.

Copy and paste the screenshot into **part 1(d)(ii)** in the evidence document.

[1]

(e) The procedure `BubbleSort()` needs to perform a bubble sort on the array and print the contents of the sorted array.

Write program code for the procedure `BubbleSort()` **and** call it from the main program.

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

[4]

2 A computer program is being developed that uses a set of cards. The program is written using object-oriented programming.

The program has two classes: `Card` and `Hand`.

The methods and attributes of these classes are shown:

Card	
`Number : INTEGER`	stores the card number from 1 to 5 inclusive
`Colour : STRING`	stores the card colour: red, blue or yellow
`Constructor()`	takes a number and colour as parameters and sets the private values to these parameters
`GetNumber()`	returns the card number
`GetColour()`	returns the card colour

Hand	
`Cards : ARRAY[0:9] OF Card`	1D array of type `Card`
`FirstCard : INTEGER`	stores the position of the first card in the hand
`NumberCards : INTEGER`	stores the number of cards in the hand
`Constructor()`	takes five card objects as parameters, assigns each card to the array `Cards[]`, initialises `FirstCard` to 0 and `NumberCards` to 5
`GetCard()`	takes an index as a parameter and returns the card at that index in the array

(a) (i) Write program code to declare the class `Card`, its attributes and constructor.

Do **not** write program code for the get methods.

Use your programming language appropriate constructor.

All attributes must be private. If you are writing in Python, include attribute declarations using comments.

Save your program as **Question2_N22**.

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

[5]

(ii) Write program code for the class methods `GetNumber()` and `GetColour()`.

Save your program.

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

[3]

© UCLES 2022

(iii) The program is tested with the following cards:

Number	Colour
1	red
2	red
3	red
4	red
5	red
1	blue
2	blue
3	blue
4	blue
5	blue
1	yellow
2	yellow
3	yellow
4	yellow
5	yellow

Write program code to declare each of these cards as a variable of type `Card` in the main program.

Save your program.

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

[2]

(b) (i) Write program code to declare the class `Hand`, its attributes and constructor.

Do **not** write the get methods.

Use your programming language appropriate constructor.

All attributes must be private. If you are writing in Python, include attribute declarations using comments.

Save your program.

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

[6]

(ii) The get method `GetCard()` takes an index as a parameter and returns the card stored at that index in the array.

Write program code for the method `GetCard()`.

Save your program.

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

[2]

(iii) Two players are declared with 5 cards each.

Player 1 has the cards: 1 red, 2 red, 3 red, 4 red, 1 yellow.
Player 2 has the cards: 2 yellow, 3 yellow, 4 yellow, 5 yellow, 1 blue.

Write program code to declare player 1 and player 2 as objects of type `Hand`, with the cards indicated.

Save your program.

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

[2]

© UCLES 2022

(c) The function `CalculateValue()` takes a player's hand as a parameter and returns a score calculated using the following rules:

- If a card is red, 5 points are added to the player's score.
- If a card is blue, 10 points are added to the player's score.
- If a card is yellow, 15 points are added to the player's score.
- The number of each card in the hand is added to the player's score.

(i) Write program code for the function `CalculateValue()`.
Assume that there are only 5 cards in the player's hand in this function.

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

[6]

(ii) Amend the main program by writing program code to use the function `CalculateValue()` for each of the two players. The player with the highest score wins.

Output an appropriate message to identify the winning player, or if the game was a draw (both players have the same number of points).

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

[4]

(iii) Test your program.

Take a screenshot to show the output.

Copy and paste the screenshot into **part 2(c)(iii)** in the evidence document.

[1]

3 A binary tree consists of nodes. Each node has 3 integer values: a left pointer, data and a right pointer.

The binary tree is stored using a global 2D array.

The pseudocode declaration for the array is:

```
DECLARE ArrayNodes : ARRAY[0:19, 0:2] OF INTEGER
```

For example:

- `ArrayNodes[0, 0]` stores the left pointer for the first node.
- `ArrayNodes[0, 1]` stores the data for the first node.
- `ArrayNodes[0, 2]` stores the right pointer for the first node.

`-1` indicates a null pointer, or null data.

(a) Write program code to:

- declare the global 2D array `ArrayNodes`
- initialise all 3 integer values to `-1` for each node.

Save your program as **Question3_N22**.

Copy and paste the program code into **part 3(a)** in the evidence document.

[3]

(b) The binary tree stores the following values:

Index	Left pointer	Data	Right pointer
0	1	20	5
1	2	15	–1
2	–1	3	3
3	–1	9	4
4	–1	10	–1
5	–1	58	–1
6	–1	–1	–1

`FreeNode` stores the index of the first free element in the array, initialised to 6.

`RootPointer` stores the index of the first node in the tree, initialised to 0.

Amend your program by writing program code to store the given data in `ArrayNodes` **and** initialise the free node and root node pointers.

Save your program.

Copy and paste the program code into **part 3(b)** in the evidence document.

[2]

(c) The following recursive pseudocode function searches the binary tree for a given value. If the value is found, the function must return the index of the value. If the value is not found, the function must return –1.

The function is incomplete. There are **four** incomplete statements.

```
FUNCTION SearchValue(Root : INTEGER,
                     ValueToFind : INTEGER) RETURNS INTEGER

   IF Root = -1 THEN

      RETURN -1

   ELSE

     IF ArrayNodes[Root, 1] = ValueToFind THEN

        RETURN .....

     ELSE

        IF ArrayNodes[Root, 1] = -1 THEN

           RETURN -1

        ENDIF

     ENDIF

   ENDIF

   IF ArrayNodes[Root, 1] ..... ValueToFind THEN

      RETURN SearchValue(ArrayNodes[....., 0], ValueToFind)

   ENDIF

   IF ArrayNodes[Root, .....] < ValueToFind THEN

      RETURN SearchValue(ArrayNodes[Root, 2], ValueToFind)

   ENDIF

ENDFUNCTION
```

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

Save your program.

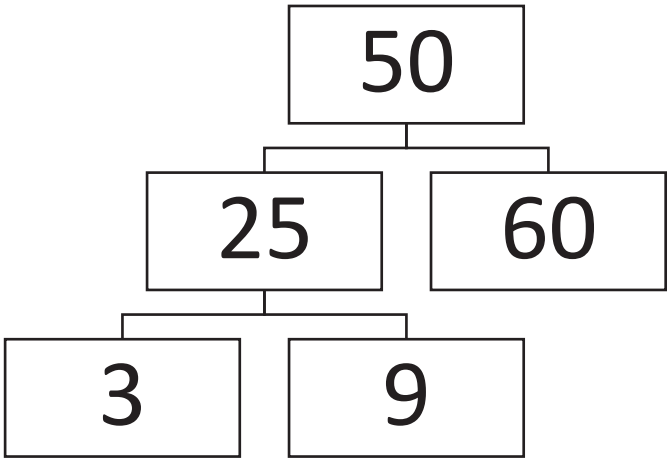
Copy and paste the program code into **part 3(c)** in the evidence document.

[5]

(d) A post order traversal performs the following operation:

- visit the left node
- visit the right node
- output the root.

For example, in the following tree, the output would be: **3 9 25 60 50**



An outline of the `PostOrder()` procedure is:

- If left node is not empty, make a recursive call with the left node as the root.
- If right node is not empty, make a recursive call with the right node as the root.
- Output the current root node.

The procedure `PostOrder()` takes the root node as a parameter.

Write program code for the procedure `PostOrder()`.

Save your program. Copy and paste the program code into **part 3(d)** in the evidence document.

[7]

(e) (i) Amend the main program by writing program code to:

- call the function `SearchValue()` to find the position of the number 15 in the tree
- use the result from `SearchValue()` to output either the index of the value if found, or an appropriate message to state that the value was not found
- call the procedure `PostOrder()`.

Save your program. Copy and paste the program code into **part 3(e)(i)** in the evidence document.

[3]

(ii) Test your program.

Take a screenshot to show the output.

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

[1]

BLANK PAGE

Permission to reproduce items where third-party owned material protected by copyright is included has been sought and cleared where possible. Every reasonable effort has been made by the publisher (UCLES) to trace copyright holders, but if any items requiring clearance have unwittingly been included, the publisher will be pleased to make amends at the earliest possible opportunity.

To avoid the issue of disclosure of answer-related information to candidates, all copyright acknowledgements are reproduced online in the Cambridge Assessment International Education Copyright Acknowledgements Booklet. This is produced for each series of examinations and is freely available to download at www.cambridgeinternational.org after the live examination series.

Cambridge Assessment International Education is part of Cambridge Assessment. Cambridge Assessment is the brand name of the University of Cambridge Local Examinations Syndicate (UCLES), which is a department of the University of Cambridge.

© UCLES 2022