Cambridge Assessment International Education

Cambridge International AS & A Level

COMPUTER SCIENCE **9618/42**

Paper 4 Practical **May/June 2023**

2 hours 30 minutes

4774742619

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

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.

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 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_9999

Two source files are used to answer **Question 3**. The files are called **`Employees.txt`** and **`HoursWeek1.txt`**

A class declaration can be used to declare a record.

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

1 A 1D array needs to store the names of 10 animals.

(a) Write program code to declare the global string array `Animals` to store 10 items.

Save your program as **Question1_J2023**. Copy and paste the program code into **part 1(a)** in the evidence document.

[2]

(b) The main program needs to store the following animals in the array:

horse
lion
rabbit
mouse
bird
deer
whale
elephant
kangaroo
tiger

Write program code to store these animal names in the array.
The names must be in lower case and stored in the order given in the list.

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

[2]

(c) The following pseudocode procedure sorts the array into a descending alphabetical order using only the first character in each animal name.

The function `LENGTH(DataArray)` returns the number of elements in the array `DataArray`.

The function `MID(String, Start, Quantity)` returns `Quantity` number of characters from `String` starting at index `Start`. The first character in the string is index 0, for example:

`MID("tiger", 0, 2)` will return `"ti"`

There are **four** incomplete statements in the procedure.

```
PROCEDURE SortDescending()

  DECLARE ArrayLength : INTEGER

  DECLARE Temp : STRING

  ArrayLength ← LENGTH(Animals)

  FOR X ← 0 TO ArrayLength - 1

    FOR Y ← …… TO ArrayLength - X - 1

      IF MID(Animals[Y], 0, 1) < MID(Animals[……], 0, 1) THEN

        Temp ← Animals[……]

        Animals[Y] ← Animals[Y + 1]

        Animals[Y + 1] ← ……

      ENDIF

    NEXT Y

  NEXT X

ENDPROCEDURE
```

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

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

[6]

(d) (i) Write program code to amend the main program to:

- call the procedure `SortDescending()`
- output the sorted contents of the array with each animal name on a new line.

Save your program.

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

[3]

(ii) Test your program.

Take a screenshot of the output.

Save your program.

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

[1]

2 A business sells a single product. Customers can purchase one or more of this product.

Each sale has an ID and a quantity, for example "ABC" and 2

The business needs a program to store the data about the sales in a circular queue data structure.

(a) Write program code to declare a record structure, `SaleData`, to store the data about each sale.

Save your program as **Question2_J2023**.

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

[2]

(b) Write program code to:

- declare a global array, `CircularQueue`, of 5 items to store the sale records
- declare the global pointers `Head` and `Tail`
- declare the global variable `NumberOfItems`
- initialise all elements of the array `CircularQueue` to an empty record, where the ID is null (`""`) and quantity is `-1`
- initialise `Head`, `Tail` and `NumberOfItems` to `0`

Save your program.

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

[4]

(c) The function `Enqueue()`:

- takes a new record as a parameter
- inserts the record in the circular queue at the element pointed to by `Tail`
- updates pointers and other variables as required
- returns `-1` if the circular queue is full
- returns `1` if the record is stored successfully.

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

Save your program.

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

[6]

(d) The function `Dequeue()`:

- returns a null or empty record if the circular queue is empty
- returns the first record in the queue if the circular queue is not empty
- updates pointers and other variables as required.

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

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

[6]

(e) The procedure `EnterRecord()`:

- takes an ID and quantity as input and creates a sale record
- uses `Enqueue()` to insert the record in the circular queue
- outputs `"Full"` if the record was not inserted in the circular queue
- outputs `"Stored"` if the record was inserted in the circular queue.

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

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

[5]

(f) The following sale records need to be entered into the program:

ID	Quantity
ADF	10
OOP	1
BXW	5
XXZ	22
HQR	6
LLP	3

(i) Amend the main program to:

- use `EnterRecord()` to input the six records in the table
- use `Dequeue()` to remove one record
- output either the ID and quantity of the removed record, or an error message if the circular queue is empty
- use `EnterRecord()` to input the record with the ID "LLP" for a second time
- output the ID and quantity for all the records currently stored in the array `CircularQueue`.

Write program code to perform these tasks.

Save your program.

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

[4]

(ii) Test your program.

Take a screenshot of the output.

Save your program.

Copy and paste the screenshot into **part 2(f)(ii)** in the evidence document.

[1]

3 A company needs a computer program to store data about its employees.

Part of the program is being written using object-oriented programming.

The class `Employee` stores data about the employees. Each employee has an employee number, a job title and hourly pay rate. The class will also store the amount they are paid each week over a 52-week year in a 1D array.

Employee	
`HourlyPay : REAL`	stores the amount each employee gets paid each hour
`EmployeeNumber : STRING`	stores the employee's unique number
`JobTitle : STRING`	stores the employee's job title
`PayYear2022 : ARRAY[0:51] OF REAL`	stores the amount the employee has been paid each week
`Constructor()`	initialises `HourlyPay`, `EmployeeNumber` and `JobTitle` from the values passed as parameters initialises all 52 elements in `PayYear2022` to `0.0`
`GetEmployeeNumber()`	returns the employee number
`SetPay()`	takes the week number and number of hours worked that week as parameters calculates and stores the pay for that week in `PayYear2022`
`GetTotalPay()`	returns the total of all the values in `PayYear2022`

(a) (i) Write program code to declare the class `Employee`.

You only need to declare the class and its constructor. Do **not** declare any other methods.

Use your programming language appropriate constructor.

If you are writing program code in Python, include attribute declarations using comments.

Save your program as **Question3_J2023**.

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

[5]

(ii) The method `GetEmployeeNumber()` returns the employee number.

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

Save your program.

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

[2]

(iii) The method `SetPay()`:

- takes a week number and the number of hours worked that week as parameters
- calculates the pay for that week by multiplying the hourly pay by the number of hours worked that week
- stores the calculated pay in the appropriate index for that week in `PayYear2022`.

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

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

[3]

(iv) The method `GetTotalPay()` returns the total of all the values in `PayYear2022`.

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

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

[2]

(b) The child class `Manager` inherits from the parent class `Employee`.

A manager gets a bonus. This bonus value is a percentage, for example 10.0%.
When calculating the pay, the number of hours the manager worked that week is increased by the bonus value.

Manager	
`BonusValue : REAL`	stores the bonus value, for example 10.0 represents a 10.0% increase
`Constructor()`	takes bonus value, hourly pay, employee number and job title as parameters initialises `BonusValue` to its parameter value
`SetPay()`	takes the week number and number of hours worked as parameters increases the number of hours worked by the bonus value calls the `SetPay()` method from the parent class

(i) Write program code to declare the class `Manager`.

You only need to declare the class and its constructor. Do **not** declare any other methods.

Use your programming language appropriate constructor.

If you are writing in Python, include attribute declarations using comments.

Save your program.

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

[4]

(ii) The `Manager` method `SetPay()` overrides the method from the parent class and:

- takes the week number and number of hours worked as parameters
- increases the number of hours worked by the bonus value
- calls `SetPay()` from the parent class.

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

Save your program.

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

[3]

(c) The main program has a global 1D array, `EmployeeArray`, to store data about eight employees. Each employee is stored as an object of type `Employee`.

The file `Employees.txt` stores data about the employees, in the order:

- hourly pay
- employee number
- bonus value (where included)
- job title.

Only employees who are managers have a bonus value saved. For example:

- The first employee is a Junior Developer, with employee number 12452 and an hourly pay of $15.22. This employee does not have a bonus value.
- The third employee is an Interface Manager, with employee number 02586 and an hourly pay of $22.50. This employee has a bonus value of 5.25%.

Write the main program to:

- declare the array to store data about 8 employees
- read in the data from the file for each employee
- instantiate each employee as either `Employee` (if the employee does not have a bonus value) or `Manager` (if the employee has a bonus value).

Save your program.

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

[7]

(d) The file `HoursWeek1.txt` stores the number of hours each employee has worked in week 1, in the order:

- employee number
- number of hours worked.

For example, the first set of data is for employee 21548 who has worked 50.0 hours.

The procedure `EnterHours()`:

- reads in the values from the file
- finds the location of each employee in `EmployeeArray`
- calls the method `SetPay()` for each employee.

Write program code for `EnterHours()`.

Save your program.

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

[4]

(e) (i) The main program needs to call `EnterHours()` and use the method `GetTotalPay()` to output the employee number and total pay for each of the eight employees.

Amend the main program to perform these tasks.

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

[2]

(ii) Test your program.

Take a screenshot of the output.

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

[1]