



Cambridge International AS & A Level

CANDIDATE
NAMECENTRE
NUMBER

--	--	--	--	--

CANDIDATE
NUMBER

--	--	--	--

COMPUTER SCIENCE

9618/23

Paper 2 Fundamental Problem-solving and Programming Skills

May/June 2025

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 or correction fluid.
- 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 **20** pages. Any blank pages are indicated.



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

1 A program is being developed for the management of a sports centre.

(a) The programmer developing the software decides to use modules (procedures or functions) and local variables. They also decide not to use global variables.

(i) State **two** reasons why the programmer decides to use modules.

1

.....

2

.....

[2]

(ii) State **one** difference between local and global variables.

.....

..... [1]

(iii) State **two** benefits of using local variables.

1

.....

2

.....

[2]

(b) The pseudocode design contains a number of expressions.

Complete each pseudocode expression so that it evaluates to the value shown.

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

Expression	Evaluates to
..... (68)	'D'
..... (04/02/2025)	2
.....TRUE	FALSE
..... (..... ("Court1.4Upper" , ,))	1.4

[5]





- (c) The table lists some of the variables used in the program.

Complete the table by writing the most appropriate data type for each variable.

Variable	Use of variable	Data type
MemberCount	stores how many members have used the sports centre each day	
TotalTakings	stores the total amount of money taken each day	
BookingConfirmed	stores the state of a booking for an exercise class; a booking is either confirmed or not confirmed	
MemberDOB	to calculate when to send an email with a birthday message to a member	

[4]



- 2 A programmer uses a number of Abstract Data Types (ADT) in the programs that she is developing.

- (a) A stack using memory locations 400 to 409 is used in one program.

The diagram represents the current state of the stack.

A variable `TopOfStack` pointer indicates the last value added to the stack.

Stack

Memory location	Value
409	
408	
407	
406	
405	
404	
403	'P' ← <code>TopOfStack</code>
402	'X'
401	'D'
400	'B'

- (i) Complete the answer column in the following table:

	Answer
the memory location of the value that has been on the stack for the longest time	
the number of consecutive push operations that will result in the <code>TopOfStack</code> variable containing 409	

[2]



(ii) The diagram shows the current state of the stack.

Stack	
Memory location	Value
409	
408	
407	'A' ← TopOfStack
406	'C'
405	'F'
404	'K'
403	'B'
402	'S'
401	'R'
400	'D'

The following sequence of operations are performed:

PUSH 'T'
 POP
 POP
 PUSH 'Z'
 PUSH 'X'
 POP
 PUSH 'Y'

Complete the following diagram to show the state of the stack after the operations have been performed.

Stack	
Memory location	Value
409	
408	
407	
406	
405	
404	
403	
402	
401	
400	





- (b) In a different program, the programmer decides to implement a linked list using an array of records.

- (i) The array is represented in the diagram.

A pointer value of -1 indicates the end of the linked list.

Complete the pointer field to produce a linked list that is in alphabetical order.

Index	Data	Pointer
0	"Neptune"	
1	"Jupiter"	
2	"Saturn"	
3	"Earth"	
4	"Mercury"	
5	"Uranus"	

[3]

- (ii) The variable `StartPointer` contains the index of the first item in the linked list.

State the value of the variable `StartPointer`

..... [1]

- (c) A third program is also being created to manage a queue.

Describe **two** features of a queue.

Feature one

.....

Feature two

.....

[2]





The module will:

step 2 – simulate **one** roll of the dice by generating a random integer between 1 and 6 inclusive

step 4 – repeat, as required from **step 2**

step 5 – calculate the average value of the random integers generated

step 6 – return the average value.

Write pseudocode for the module `RollDice()`

..... [7

- DO NOT WRITE IN THIS MARGIN

- DO NOT WRITE IN THIS MARGIN

DO NOT WRITE IN THIS MARGIN

DO NOT WRITE IN THIS MARGIN

File line number	File data
1	"350"
2	"502"
3	"434"
...	...
49	"492"
50	"872"
51	"772"
52	"201"

The owner requires a module to output all week numbers where the number of items sold is greater than 500.

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

..... [5]



(b) Once the program has been completed, it is tested using the walkthrough method.

Describe the walkthrough method and explain how it can be used to identify errors.

.....

.....

.....

.....

.....

.....

.....

..... [3]



5 A module `Parity()` takes a string as a parameter. The parameter has the identifier `BitString` and it represents a binary value.

The module will concatenate a single character to the end of `BitString` by applying one of the two rules:

- '0' if Bitstring contains an even number of 1s
- '1' if Bitstring contains an odd number of 1s.

The modified value of `BitString` is then returned.

For example:

Parameter	String returned	Explanation
"0010010110"	"0010010110 0 "	there are an even number of 1s in the string passed to <code>Parity()</code> so a '0' is concatenated to the end of <code>BitString</code>
"101010"	"101010 1 "	there are an odd number of 1s in the string passed to <code>Parity()</code> so a '1' is concatenated to the end of <code>BitString</code>

Write pseudocode for the module `Parity()`

Assume the parameter `BitString` can only contain the characters '0' and '1'

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.



.....

.....

.....

.....

.....

.....

.....

.....

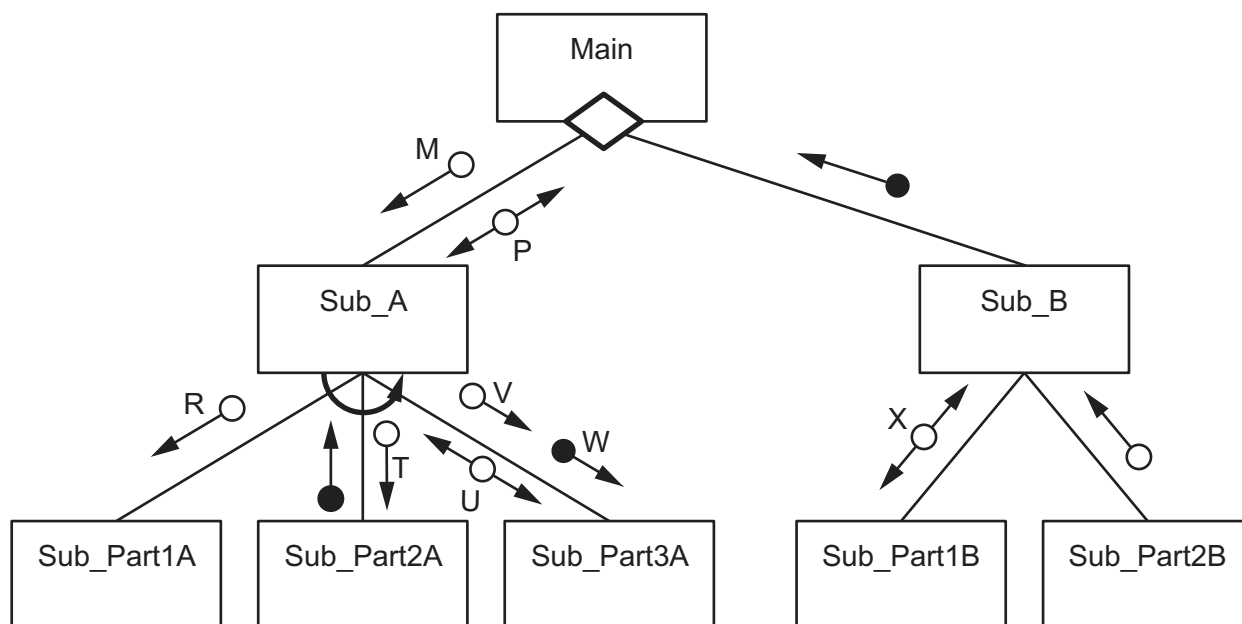
.....

..... [8]





6 Study the structure chart:



Some of the parameter data types are:

- R and V are of type `INTEGER`
- T is of type `REAL`
- U is of type `STRING`
- W is of type `BOOLEAN`.

Some of the modules in the structure chart are functions, others are procedures.

(a) Explain **one** difference between functions and procedures.

..... [1]

(b) Write the pseudocode to define the module headers:

Sub_Part1A

.....

Sub_Part2A

.....

Sub_Part3A



.....





(c) The structure chart uses the two symbols shown.

Complete the table to explain what each symbol means and how the relevant modules in the structure chart are affected.

Symbol	Explanation
	<div><div></div><div></div><div></div></div>
	<div><div></div><div></div><div></div></div>

[3]



DO NOT WRITE IN THIS MARGIN

DO NOT WRITE IN THIS MARGIN

DO NOT WRITE IN THIS MARGIN

DO NOT WRITE IN THIS MARGIN

Module	Description
UpdateVisit()	<ul style="list-style-type: none"> called with two parameters: <ol style="list-style-type: none"> loyalty points of type <code>INTEGER</code> last visit date of type <code>DATE</code> change the loyalty points: <ul style="list-style-type: none"> increase by four if the last visit date and the current date are in the same month otherwise increase by one change the last visit date to the current date the changed values must be available to the code that follows the call to <code>UpdateVisit()</code>

Assume the customer has made at least **one** previous visit.

[illegible]

[5]



- (ii) The programmer decides to amend the `UpdateVisit()` module so that loyalty points are further increased if the customer visits the coffee shop the same day of the week as their last visit.

Write the pseudocode for this condition.

.....
..... [1]





(b) A text file `Loyalty.txt` will be used to store the data items for the loyalty scheme.

The text file `Loyalty.txt` contains only one line of data for each customer.

Each data item is separated by a comma:

`<CustomerID>,<LoyaltyPointsString>,<LastVisitDateString>`

`LastVisitDateString` is always eight characters in length in the format: DDMMYYYY.

An example of how a line of data will be stored in the text file `Loyalty.txt` is:

`"100123,132,05032025"`

This example shows that the customer with an ID of 100123 had 132 loyalty points following their last visit to the coffee shop on 05/03/2025.

If a customer visits the coffee shop on a Monday, the value of their loyalty points is increased by 10.

The programmer has defined **two** more program modules:

Module	Description
<code>FindCustomer()</code> (is already written)	<ul style="list-style-type: none"> called with a parameter of type <code>STRING</code> that represents a customer ID returns the line of data read from the text file <code>Loyalty.txt</code> containing the data items for that customer
<code>MondayCheck()</code>	<ul style="list-style-type: none"> called with a parameter of type <code>STRING</code> that represents the customer ID of a customer calls <code>FindCustomer()</code> extracts the loyalty points from the string returned by <code>FindCustomer()</code> increases the loyalty points for the current customer by 10 if the day visited is a Monday returns an integer value representing the loyalty points





The module `FindCustomer()` must be used and assume it returns a valid string for the current customer.

[8]



(ii) A date is stored in `LastVisitDateString` in the format:

DDMMYYYY

DD is a 2-digit string

MM is a 2-digit string

YYYY is a 4-digit string.

For example, the date 03/09/2024 is stored as "03092024"

An algorithm is needed to convert the string stored in `LastVisitDateString` to data type `DATE` which is then stored in the variable `LastVisitDate`.

Complete the pseudocode for this algorithm:

Assume that `LastVisitDateString` has been declared and contains valid data.

```
DECLARE DayInt, MonthInt, YearInt : INTEGER
```

```
DECLARE LastVisitDate : DATE
```

.....

.....

.....

.....

.....

.....

.....

.....

[4]







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.

