



Cambridge International AS & A Level

CANDIDATE
NAME

--

CENTRE
NUMBER

--	--	--	--	--

CANDIDATE
NUMBER

--	--	--	--



COMPUTER SCIENCE

9608/22

Paper 2 Fundamental Problem-solving and Programming Skills

May/June 2021

2 hours

You must answer on the question paper.

No additional materials are needed.

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.

This document has **20** pages. Any blank pages are indicated.

- 1 (a) Maintenance of programs may be needed for a number of different reasons.

State **two** types of maintenance **and** give a reason why each may be needed.

Type

Reason

.....

Type

Reason

.....

[4]

- (b) State why characters need to be represented in ASCII or Unicode before they can be processed.

.....

..... [1]

- (c) Each line of a text file contains several data items. A special character is inserted between data items before the line is written to the file.

Explain why a special character is used in this way.

.....

.....

.....

..... [2]

- (d) Each pseudocode statement in the following table may contain an error due to the incorrect use of the function or operator.

Describe the error in each case, **or** write 'NO ERROR' if the statement contains no error.

Refer to the **Appendix** on page 18 for the list of built-in pseudocode functions and operators.

Statement	Error
Code ← RIGHT("Cap" & "art", 4)	
Status ← MID("Computer", 7, 5)	
Size ← LENGTH("Password") * 2	
NextChar ← CHR('A')	
Index ← Index & 3	

[5]

2 Study the following pseudocode.

```
DECLARE Overload : BOOLEAN
```



```
PROCEDURE LEM()
```

```
    DECLARE Status : BOOLEAN
```

```
    DECLARE Landed : INTEGER
```

```
    Overload ← FALSE
```

```
    Landed ← FALSE
```

```
    WHILE Landed = FALSE
```

```
        Status ← Sample()
```

```
        IF Status = TRUE
```

```
            THEN
```

```
                Landed ← SubA(42)
```

```
            ELSE
```

```
                Overload ← SubB(37)
```

```
                IF Overload = TRUE
```

```
                    THEN
```

```
                        CALL Display("Alarm 1202")
```

```
                    ENDIF
```

```
            ENDIF
```

```
    ENDWHILE
```

```
ENDPROCEDURE
```

(a) Examine the pseudocode **and** complete the following table:

Answer

The identifier name of a global variable	
The name of the loop structure	
The identifier involved in a data type mismatch	
The name of a procedure that takes a parameter	
The name of a function	

[5]

(b) Draw a program flowchart to represent the pseudocode algorithm.

Variable declarations are not required in program flowcharts.



[5]

- 3 (a) (i) Module names and parameters are features that may be represented on a structure chart.

State **two other** features than can be represented on a structure chart.

Feature 1

Feature 2

[2]

- (ii) The headers for three modules in a program are defined in pseudocode as follows:


Pseudocode module header
PROCEDURE Create(S2 : INTEGER, P3 : STRING)
PROCEDURE Modify(S2 : INTEGER, BYREF P4 : STRING)
FUNCTION Delete(P4 : INTEGER, M4 : STRING) RETURNS INTEGER

A fourth module, `Membership()`, may call any one of the three modules.

Draw a structure chart to represent the information given about the **four** modules.

[5]

- (b) Draw a diagram to show the stages of the program development cycle. Use arrows to indicate how the stages are linked.



[2]

5 (a) An Integrated Development Environment (IDE) will be used to develop a program.

(i) An IDE includes features for program presentation.

State **two** of these presentation features.

Feature 1

.....

Feature 2

.....

[2]

(ii) Name **two** IDE features that can help with initial error detection.

Feature 1

.....

Feature 2

.....

[2]

(b) (i) A function, `Verify()`, is written in pseudocode.

Write the **two** missing lines to complete the pseudocode.

```

FUNCTION Verify(UserID : STRING) RETURNS BOOLEAN
    .....
    DECLARE Password : STRING
    OUTPUT "Please Input your password: "
    INPUT Password
    Response ← Validate(UserID, Password) AND Today()
    .....
ENDFUNCTION
    
```

[2]

6 A program stores data about stock items in four global 1D arrays as follows:

Array	Data type	Description	Example data value	Initial data value
StockID	STRING	the stock item ID (eight alpha-numeric characters)	"JBCD0002"	""
Description	STRING	a description of the item (alphabetic characters only)	"soap"	""
Quantity	INTEGER	the number in stock	9	0
Cost	REAL	the cost of the item	1.45	0.0

- Each array contains 10000 elements.
- Elements with the same index relate to the same stock item. For example, `StockID[3]` contains the ID for the product whose description is in `Description[3]`.
- The `StockID` array is not sorted.

The program will be modified so that the data from the arrays can be stored in a text file for backup. You may assume that a backup file contains only valid stock data.

The programmer has started to define program modules as follows:

Module	Description
<code>Unpack()</code>	<ul style="list-style-type: none"> • called with two parameters: <ul style="list-style-type: none"> ◦ an array index ◦ a string value read from one line of the backup file • extracts the four data values from the string and assigns each to the appropriate array
<code>Restore()</code>	<ul style="list-style-type: none"> • called with a string representing the name of a backup file • returns <code>FALSE</code> if the file is empty • sets all elements of each array to the initial data value as given in the table • reads the backup file line by line calls <code>Unpack()</code> to extract data from each line and assign values to the corresponding arrays • returns <code>FALSE</code> if the arrays are full but there are still lines in the file, otherwise returns <code>TRUE</code>
<code>StockSummary()</code>	<p>For all items where <code>StockID</code> does not contain the initial value:</p> <ul style="list-style-type: none"> • counts the number of stock entries in the <code>StockID</code> array • outputs the overall value of all items in stock (cost multiplied by the quantity) • outputs the number of stock entries

(a) Write **program code** for the module `StockSummary()`.

Visual Basic and Pascal: You should include the declaration statements for variables.

Python: You should show a comment statement for each variable used with its data type.

Programming language

Program code

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

[5]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [8]

(c) The module description of `GetValidFilename()` is as follows:

Module	Description
<code>GetValidFilename()</code>	<ul style="list-style-type: none">• prompts and inputs a filename• validates the filename by checking that it:<ul style="list-style-type: none">◦ is between 4 and 10 characters in length (inclusive)◦ contains only alphanumeric characters• if the filename is invalid, outputs a warning message and asks the user to try again• otherwise returns the valid filename

Write **program code** for the module `GetValidFilename()`.

Visual Basic and Pascal: You should include the declaration statements for variables.

Python: You should show a comment statement for each variable used with its data type.

Programming language

Program code

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Appendix

Built-in functions (pseudocode)

Each function returns an error if the function call is not properly formed.

`MID(ThisString : STRING, x : INTEGER, y : INTEGER)` RETURNS STRING
returns a string of length `y` starting at position `x` from `ThisString`

Example: `MID("ABCDEFGH", 2, 3)` returns "BCD"

`LENGTH(ThisString : STRING)` RETURNS INTEGER
returns the integer value representing the length of `ThisString`

Example: `LENGTH("Happy Days")` returns 10

`LEFT(ThisString : STRING, x : INTEGER)` RETURNS STRING
returns leftmost `x` characters from `ThisString`

Example: `LEFT("ABCDEFGH", 3)` returns "ABC"

`RIGHT(ThisString : STRING, x : INTEGER)` RETURNS STRING
returns rightmost `x` characters from `ThisString`

Example: `RIGHT("ABCDEFGH", 3)` returns "FGH"

`INT(x : REAL)` RETURNS INTEGER
returns the integer part of `x`

Example: `INT(27.5415)` returns 27

`LCASE(ThisChar : CHAR)` RETURNS CHAR
returns the character value representing the lower case equivalent of `ThisChar`
If `ThisChar` is not an upper case alphabetic character, it is returned unchanged.

Example: `LCASE('W')` returns 'w'

`ASC(ThisChar : CHAR)` RETURNS INTEGER
returns the ASCII value of character `ThisChar`

Example: `ASC('A')` returns 65

`CHR(x : INTEGER)` RETURNS CHAR
returns the character whose ASCII value is `x`

Example: `CHR(87)` returns 'W'

Operators (pseudocode)

Operator	Description
&	Concatenates (joins) two strings Example: "Summer" & " " & "Pudding" produces "Summer Pudding"
AND	Performs a logical AND on two Boolean values Example: TRUE AND FALSE produces FALSE
OR	Performs a logical OR on two Boolean values Example: TRUE OR FALSE produces TRUE

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 the Cambridge Assessment Group. Cambridge Assessment is the brand name of the University of Cambridge Local Examinations Syndicate (UCLES), which itself is a department of the University of Cambridge.