

CANDIDATE  
NAME

CENTRE  
NUMBER

--	--	--	--	--

CANDIDATE  
NUMBER

--	--	--	--



**COMPUTER SCIENCE**

**9608/22**

Paper 2 Fundamental Problem-solving and Programming Skills

**May/June 2018**

**2 hours**

Candidates answer on the Question Paper.

No Additional Materials are required.

No calculators allowed.

**READ THESE INSTRUCTIONS FIRST**

Write your Centre number, candidate number and name in the spaces at the top of this page.

Write in dark blue or black pen.

You may use an HB pencil for any diagrams, graphs or rough working.

Do not use staples, paper clips, glue or correction fluid.

**DO NOT WRITE IN ANY BARCODES.**

Answer **all** questions.

No marks will be awarded for using brand names of software packages or hardware.

At the end of the examination, fasten all your work securely together.

The number of marks is given in brackets [ ] at the end of each question or part question.

The maximum number of marks is 75.

This document consists of **16** printed pages.

- 1 (a) A farm has a number of greenhouses used to grow vegetables. Each greenhouse has a different identification number. A program is needed to store temperature information for each greenhouse throughout the day.

Give a suitable **identifier name** for each of the data items.

Description of data item	Suitable identifier name
The temperature inside the greenhouse	
The temperature outside the greenhouse	
The greenhouse identification number	
The time of the temperature measurements	

[4]

- (b) (i) Program variables have values as follows:

Variable	Value
Mark	60
Subject	"Computer Science"
Grade	'B'
CourseCompleted	TRUE
AverageMark	49.5

Evaluate each expression in the following table.

If an expression is invalid, write ERROR.

For the built-in functions list, refer to the **Appendix** on page 16.

Expression	Evaluates to
"Fas" & MID(Subject, 6, 3)	
LEFT(Mark, 1)	
10 + ASC(Grade)	
MOD(AverageMark * 2, 3)	
CourseCompleted AND (Mark >= 60)	

[5]

(ii) Programming languages support different data types.

Give an appropriate data type for each of these variables from **part (b)(i)**.

Variable	Data type
Mark	
Subject	
Grade	
CourseCompleted	
AverageMark	

[5]

2 The following is a function design in pseudocode.

Line numbers are given for reference only.

```

01 FUNCTION CountDigits(InString : STRING) RETURNS CHAR
02
03 DECLARE nc : CHAR
04 DECLARE c : INTEGER
05 DECLARE n : INTEGER
06
07
08 c ← 0
09 n ← LENGTH(InString) // get number of characters for loop
10
11 WHILE n > 0 // repeat until no more characters left
12
13 nc ← LEFT(InString,1)
14 n ← n - 1
15 InString ← RIGHT(InString,n) // remove first character
16
17 IF (nc < '0') OR (nc > '9')
18 THEN
19 // do nothing
20 ELSE
21 c ← c + 1
22 ENDIF
23
24 ENDWHILE
25
26 RETURN c
27
28 ENDFUNCTION

```

(a) (i) This pseudocode includes features that make it easier to read and understand.

State **two** such features.

Feature 1 .....

Feature 2 ..... [2]

(ii) State **two** additional features that should be used to make this pseudocode easier to read and understand.

Feature 1 .....

Feature 2 ..... [2]

(b) Study the function `CountDigits()`. Identify the features of the function in the following table.

Feature	Answer
A line number containing an example of an assignment statement	
A line number containing the start of a 'pre-condition' loop	
A line number containing the end of a 'pre-condition' loop	
A line number containing the start of a selection statement	
The number of parameters passed to the <code>LEFT()</code> function	
The Boolean operator used	
The number of times the function <code>LEFT()</code> is called from within <code>CountDigits()</code> resulting from the call: <code>Result ← CountDigits("AB27C4")</code>	
The number of local variables	

[8]

(c) (i) There is a mistake in the pseudocode that would produce a data type mismatch error if a programmer were to write similar program code.

Describe this mistake and how it may be corrected.

.....

.....

.....

..... [2]

(ii) Lines 17 to 22 of the pseudocode could be written in a simplified form.

Re-write the lines in a simplified form.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

3 A chocolate factory produces bars of chocolate. A computer program controls the process.

The weight of each bar is stored in an array, `BarWeight`. The array contains 100 elements, representing the weights of 100 bars that make up one shipping box.

A procedure, `CheckWeight()`, is required to:

1. examine each array element and count how many times the weight has exceeded `MaxWeight`
2. compare the count obtained with a limit value, `Threshold`. Call procedure `ServiceCheck()` if the count exceeds the `Threshold`
3. output a message if the count does not exceed the `Threshold`. For example:

```
"ShippingBox OK - maximum weight exceeded 3 times."
```

Draw a program flowchart on the next page to represent the algorithm for the `CheckWeight()` procedure.

Assume that:

- the array contains 100 valid weight values and the first element is `BarWeight[1]`
- `MaxWeight`, `Threshold` and `BarWeight` are global variables.

Variable declarations are not required in program flowcharts.



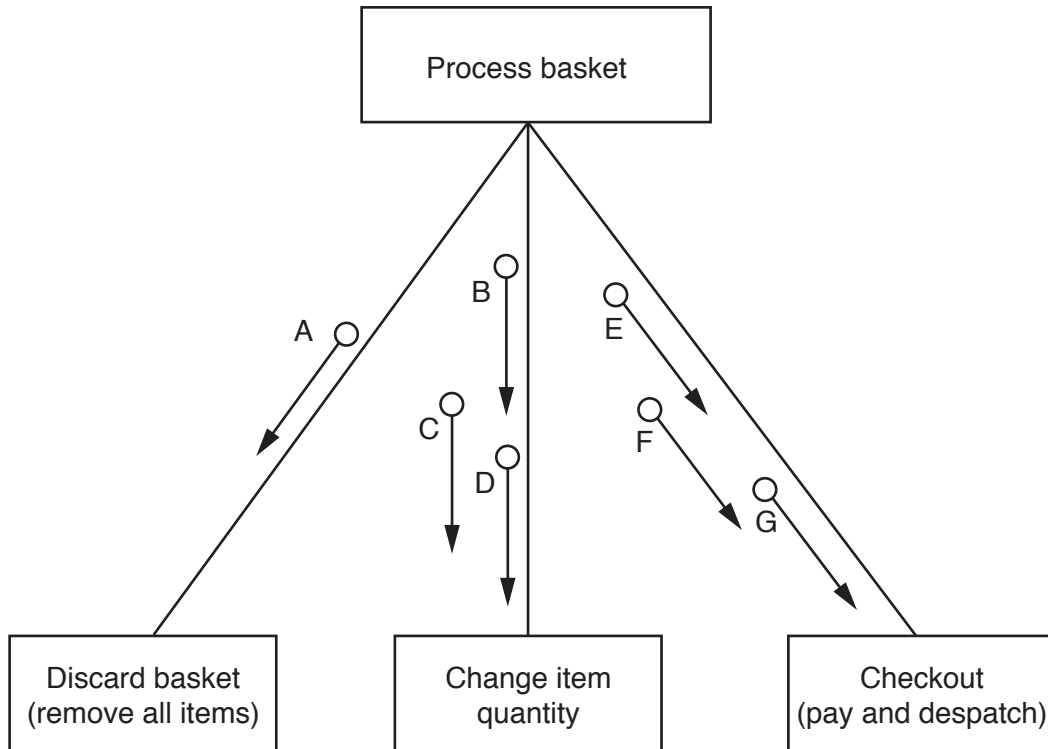
[10]

**[Turn over**

**Question 4 begins on the next page.**



4 The structure chart shows part of the design of a program for an online shopping system.



(a) (i) Draw on the chart to show the following facts.

- Each of the modules at the lower level returns a Boolean parameter, X.
- Process basket will call only one of the modules shown at the lower level.

[2]

(ii) The parameters A to G shown on the chart will be used to pass the following information.

PaymentDetails  
Quantity  
BasketID  
DeliveryAddress  
ItemID

Complete the following table to show the parameter and the information it represents.

Parameter	Information
A	
B	
C	
D	
E	
F	
G	

[3]

- 5 A golf club holds information about its members. When a member completes a round of golf, their score is stored along with their membership number and the date of the round.

A program is to be written to store and process the score information. The information to be stored is formed into a string as follows:

<MembershipNumber><Date><Score>

- (a) The program designer considers storing the strings in either a 1D array, `RoundScore` or as a separate variable for each round, for example, `RoundScore01`, `RoundScore02`, `RoundScore03` and so on.

Describe **two** advantages of storing the strings in a 1D array rather than as separate variables.

Advantage 1 .....

.....

Advantage 2 .....

.....

[2]

- (b) A procedure, `AddNewScores()` is being developed. The procedure will be coded using an Integrated Development Environment (IDE).

Name **two** features provided by an IDE that assist in the **initial detection of errors** in the procedure.

Feature 1 .....

.....

Feature 2 .....

.....

[2]



(d) The program designer decides to store the strings from **part (a)** in a text file, `ScoreDetails.txt`

Each string is formatted as follows:

`<MembershipNumber><Date><Score>`

- `MembershipNumber` is a four-digit numeric string between "1111" and "9999".
- `Date` is a six-digit numeric string in the format DDMMYY, for example "040716".
- `Score` is a two-digit numeric string in the range "50" to "99".

A procedure `OutputLowestScore()` is required to output the lowest score for an individual member.

Assume that there is at least one string stored for each member.

The program needs a procedure, `OutputLowestScore()`, to process the `ScoreDetails.txt` file and output the lowest score for an individual member.

1. Use the `GetNumber()` function to obtain a valid membership number.
2. Search the `ScoreDetails.txt` file for the lowest score for that member.
3. Output a message giving the lowest score for that member and the date of the round. For example: "The lowest score was 66 on 300917"

Write **program code** for the `OutputLowestScore()` procedure.

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



- 6 (a) Individual elements in a 1D array are referenced using an integer value that is used as the subscript to the array.

Give the technical terms for the minimum and maximum values the subscript may take.

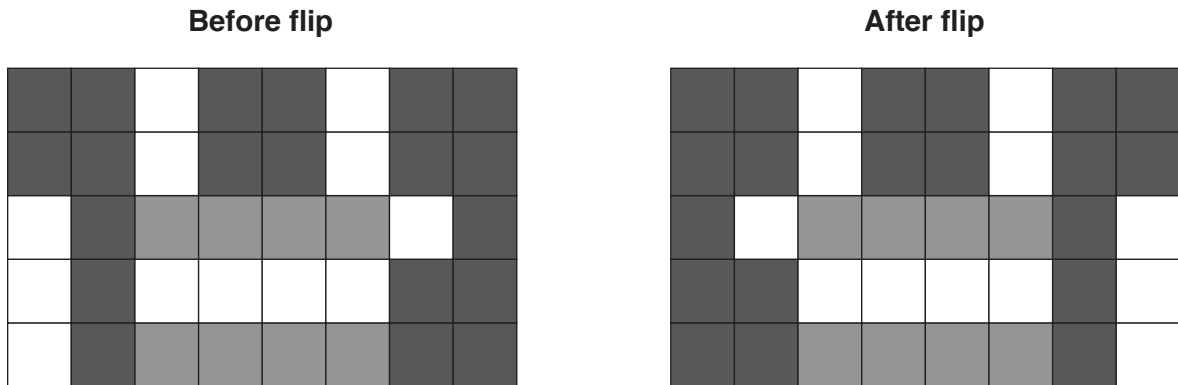
Minimum value .....

Maximum value .....

[2]

- (b) A 2D array, `Picture`, contains data representing a bitmap image. Each element of the array represents one pixel of the image. The image is grey-scale encoded where the value of each pixel ranges from 0 (representing black) to 255 (representing white) with intermediate values representing different levels of grey.

A graphics program needs a procedure, `Flip()`, to flip (reflect) the image. An example of an image before and after the function is:



The values contained in the 2D array before the flip are as follows:

**Data values**

80	80	255	80	80	255	80	80
80	80	255	80	80	255	80	80
255	80	120	120	120	120	255	80
255	80	255	255	255	255	80	80
255	80	120	120	120	120	80	80

In pseudocode, the array is declared as follows:

```
DECLARE Picture : ARRAY[1:5, 1:8] OF INTEGER
```



## 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 string "BCD"

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

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

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

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

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

`MOD(ThisNum : INTEGER, ThisDiv : INTEGER)` RETURNS INTEGER  
returns the integer value representing the remainder when `ThisNum` is divided by `ThisDiv`

Example: `MOD(10, 3)` returns 1

### 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

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 International Examinations Copyright Acknowledgements Booklet. This is produced for each series of examinations and is freely available to download at [www.cie.org.uk](http://www.cie.org.uk) after the live examination series.

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