



Cambridge International AS & A Level

CANDIDATE
NAME

CENTRE
NUMBER

--	--	--	--	--

CANDIDATE
NUMBER

--	--	--	--



COMPUTER SCIENCE

9608/22

Paper 2 Fundamental Problem-solving and Programming Skills

October/November 2020

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. Blank pages are indicated.

2

- 1 (a) Algorithms usually consist of three different stages.

One stage is INPUT.

Name the **other** stages.

- 1
- 2 [1]

- (b) An algorithm may be documented using different methods. These include structured English, a program flowchart, and pseudocode.

State what a program designer represents using one or more of these methods.

.....

..... [2]

- (c) Programming languages support different data types.

Complete the table by giving four **different** data types together with an example data value for each.

Data type	Example data value

[4]

(d) Draw lines to connect each of the following computing terms with the appropriate description.

Term	Description
Black-box testing	A structure for the temporary storage of data
File	A method used when the structure of the program is unknown
Assignment	A method of setting the value of a variable
Array	A structure for the permanent storage of data

[3]

(e) A pseudocode algorithm assigns values to three variables as follows:

```
FlagA ← TRUE
FlagB ← FALSE
FlagC ← TRUE
```

Evaluate the expressions given in the following table:

Expression	Evaluates to
NOT FlagB AND FlagC	
NOT (FlagB OR FlagC)	
(FlagA AND FlagB) OR FlagC	
NOT (FlagA AND FlagB) OR NOT FlagC	

[2]

.....

.....

.....

.....

.....

.....

..... [6]

(b) Complete the pseudocode expressions in the following table.

Use **only** functions and operators described in the **Appendix** on pages 18–19.

Expression	Evaluates to
"ALARM: " & ("Time: 1202" ,)	"ALARM: 1202"
..... ("Stepwise." , ,)	"wise"
1.5 * ("OnePointFive")	18
..... (27.5)	"27.5"
..... (9, 4)	2

[5]

(c) A problem may be decomposed into sub-tasks when designing an algorithm.

Give **three** benefits of using sub-tasks.

1

.....

2

.....

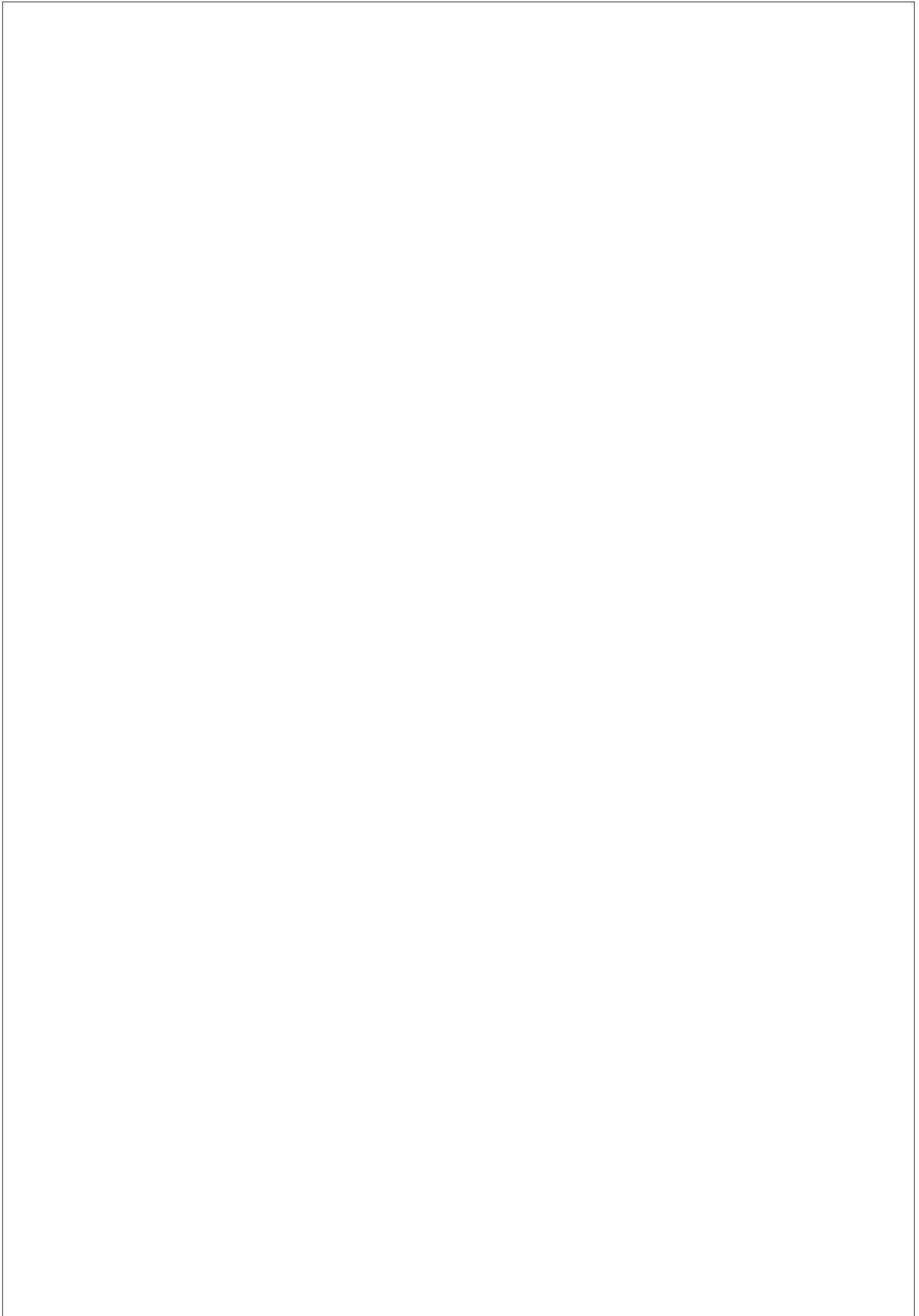
3

.....

[3]

- 4 (a) The following structured English describes an algorithm used to count the number of odd and even digits in an input sequence.
1. Initialise variables `OddCount` and `EvenCount` to zero.
 2. Prompt and input an integer.
 3. If the integer is not in the range 0 to 9 then go to step 7.
 4. If the integer is an even number then add 1 to `EvenCount`.
 5. Otherwise add 1 to `OddCount`.
 6. Repeat from step 2.
 7. Output "Same" if there are the same number of odd and even integers.
 8. Output "Odd" if there are more odd than even integers.
 9. Output "Even" if there are more even than odd integers.

Draw a flowchart on the following page to represent the algorithm.



- (b) The following pseudocode is an attempt to check whether two equal-length strings consist of identical characters.

Refer to the **Appendix** on pages 18–19 for the list of built-in functions and operators.

```

FUNCTION Compare(String1, String2 : STRING) RETURNS BOOLEAN
  DECLARE x, y, Len1, Len2 : INTEGER
  DECLARE RetFlag : BOOLEAN
  DECLARE NextChar : CHAR
  DECLARE New : STRING

  Len1 ← LENGTH(String1)
  RetFlag ← TRUE

  FOR x ← 1 TO Len1 // for each char in String1
    Len2 ← LENGTH(String2)
    NextChar ← MID(String1, x, 1) // get NextChar from String1
    New ← ""
    FOR y ← 1 TO Len2 // for each char in String2
      IF NextChar <> MID(String2, y, 1) // no match
        THEN
          New ← New & MID(String2, y, 1) // save this char from String2
        ENDIF
    ENDFOR
    String2 ← New // replace String2 with New
  ENDFOR

  IF LENGTH(String2) <> 0 // anything left in String2 ?
    THEN
      RetFlag ← FALSE
    ENDIF

  RETURN RetFlag

ENDFUNCTION

```


- (iii) There is an error in the algorithm, which means that under certain circumstances, the function will return an incorrect value.

Describe the problem. Give **two** test strings that would demonstrate it.

Problem

.....

.....

.....

.....

.....

Test String1

Test String2 [2]

- (iv) Describe the modification that needs to be made to the algorithm to correct the error.
Do **not** use pseudocode or program code in your answer.

.....

.....

.....

.....

..... [1]

- (v) State the name given to the type of testing that makes use of a trace table.

..... [1]

- (vi) State **two** features found in a typical Integrated Development Environment (IDE) that may be used for debugging a program.

1

2 [2]

Question 5 begins on the next page.

Appendix

Built-in functions (pseudocode)

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

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"

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

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"

DIV(ThisNum : INTEGER, ThisDiv : INTEGER) RETURNS INTEGER
returns the integer value representing the whole number part of the result when ThisNum is divided by ThisDiv

Example: DIV(10, 3) returns 3

NUM_TO_STRING(x : REAL) RETURNS STRING
returns a string representation of a numeric value.
Note: This function will also work if x is of type INTEGER

Example: NUM_TO_STRING(87.5) returns "87.5"

STRING_TO_NUM(x : STRING) RETURNS REAL
returns a numeric representation of a string.
Note: This function will also work if x is of type CHAR

Example: STRING_TO_NUM("23.45") returns 23.45

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.