Cambridge International AS & A Level

Cambridge International Examinations
Cambridge International Advanced Subsidiary and Advanced Level

CANDIDATE NAME									
CENTRE NUMBER						CANDIDATE NUMBER			

COMPUTING **9691/22**

Paper 2 **May/June 2015**

2 hours

Candidates answer on the Question Paper.

No Additional Materials are required.

READ THESE INSTRUCTIONS FIRST

Write your Centre number, candidate number and name on all the work you hand in.
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.

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.

This document consists of **14** printed pages and **2** blank pages.

CAMBRIDGE International Examinations

1 A high-level programming language has the built-in string handling function `MID` defined as follows:

> `MID(ThisString : STRING, x : INTEGER, n : INTEGER) RETURNS STRING`
>
> returns a substring of `n` characters from the string `ThisString` starting at position `x`.
>
> For example: `MID("STOP", 3, 2)` returns `"OP"`
>
> If the function call is not properly formed, an error is generated.

(a) State what is returned by the following function calls.

(i) `MID("FRED",1,1)`

..... [1]

(ii) `MID("FRED",5,1)`

..... [1]

(b) A date is stored in the format DDMMYYYY in the string variable `TodaysDate`.

Use the function `MID` to separate the day, month and year.

`ThisDay` ← .....

`ThisMonth` ← .....

`ThisYear` ← ..... [3]

2 The pseudocode below is intended to calculate the sum of a sequence of integers input.

The dummy value –1 ends the input.

```
DECLARE x : INTEGER
DECLARE Result : INTEGER
x ← 0
Result ← 0
WHILE x <> -1
  INPUT x
  Result ← Result + x
ENDWHILE
OUTPUT Result
```

(a) (i) The sequence of numbers 3, 5, 2, 1 is input and terminated with –1.

Complete the trace table.

x	Result	x <> -1
0	0	

Output ..... [4]

(ii) Give the expected result from the sum of the numbers 3, 5, 2, 1.

..... [1]

(iii) What is the error in the given pseudocode?

.....

.....

..... [1]

(iv) State the type of error.

..... [1]

(b) Rewrite the pseudocode so that it works correctly.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [3]

[Turn over

3 A board game is designed for two players, O and X.

At the beginning, all cells of a 3 x 3 grid are empty.

The players take turns in placing their marker in an empty cell of the grid; player O always starts.

The game ends when one player completes a row, column or diagonal or the grid is full.

Here is one example after three turns:

		O
	O	X

Ali wants to write a program to play the game.

(a) The array `Grid` is to be used to represent the contents of the grid.

Rows and columns are to be numbered from 1 to 3.

(i) To take their turn, the player inputs a row number and a column number to place their marker in an empty cell.

Write the values player X has input to place their marker, 'X', in the above diagram:

Row .....

Column ..... [1]

(ii) State the value Ali could use to represent an empty cell. ..... [1]

(iii) In a high-level programming language, write statements to:

- declare the array `Grid`
- assign the value for an empty cell to all cells

Programming language .....

Code .....

.....

.....

.....

.....

.....

.....

.....

.....

..... [7]

(b) Ali decides to validate the player input.

The input is valid if:
- the row and column numbers are within the range 1 to 3 inclusive
- the cell is empty

Ali chooses a sequence of seven pairs of integer values to simulate player input. The test starts with an empty grid.

(i) Show the contents of the grid after the input of each pair of integer values. Circle whether the input is valid or invalid. If the input is invalid state the reason.

Row	Column	Grid content	Reason (if invalid)
2	2		valid / invalid
0	1		valid / invalid
1	1		valid / invalid
1	4		valid / invalid
4	1		valid / invalid
2	0		valid / invalid
2	2		valid / invalid

[6]

(ii) In a high-level programming language, write the function `IsInputValid` which takes two integers (row number and column number) as parameters and returns:

- `TRUE` if the cell reference exists and is empty
- `FALSE` if the cell reference is out of range or the cell contains a marker already

Programming language .....

Code .....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

[5]

(c) Ali uses the top-down design approach for his overall program solution.

His design is as follows:

```
01 GameEnd ← FALSE
02 CurrentPlayer ← 'O'
03 CALL DisplayGrid()
04
05 REPEAT
06    CALL PlayerTakesTurn(CurrentPlayer)
07    CALL DisplayGrid()
08    IF HasPlayerWon() = TRUE
09       THEN
10          GameEnd ← TRUE
11          OUTPUT "Player", CurrentPlayer, "has won"
12       ELSE
13          IF GridFull() = TRUE
14             THEN
15                GameEnd ← TRUE
16                OUTPUT "Draw"
17             ELSE
18                CALL SwapPlayer(CurrentPlayer)
19          ENDIF
20    ENDIF
21 UNTIL GameEnd = TRUE
```

(i) Identify **one** feature in the above pseudocode which indicates that top-down design has been used.

.....

..... [1]

(ii) State **one** benefit of top-down design.

.....

..... [1]

(iii) Give the line number of a statement which shows:

Assignment .....

Selection .....

Iteration .....

a Function call .....

a Procedure call .....

[5]

(iv) Ali has written the pseudocode with features that make it easier to understand.

State **two** such features.

Feature 1 .....

.....

Feature 2 .....

.....

[2]

(v) Complete the identifier table below.

Identifier	**Variable or Procedure or Function or Array**	**Data type**	**Description**
`GameEnd`	Variable	`BOOLEAN`	`FALSE` if game in progress `TRUE` if there is a winner or the grid is full
`Grid`	`ARRAY`		To store the current state of the game
`CurrentPlayer`			The marker value ('O' or 'X') of the current player
`PlayerTakesTurn`			Current player chooses cell Program checks if it is valid and stores marker
`DisplayGrid`			Outputs the contents of the grid
`HasPlayerWon`			Checks if the current player has completed a row, column or diagonal
`GridFull`			Checks if the grid is full
`SwapPlayer`	`PROCEDURE`		Swaps the value of `CurrentPlayer`

[5]

(d) Write the pseudocode required for the procedure `SwapPlayer`:

.....

.....

.....

.....

.....

.....

..... [5]

(e) The current player is a winner if they have placed their markers in each cell of a row or a column or a diagonal. Ali's solution checks for a winner after every turn.

Complete the pseudocode for the subroutine `HasPlayerWon`:

```
..... HasPlayerWon() .....
DECLARE WinningLine : .....
DECLARE i : .....
WinningLine ← .....
   // check both diagonals
IF Grid[1,1] = Grid[2,2] AND Grid[1,1] = Grid[3,3]
   OR Grid[.....] = Grid[.....]
      AND Grid[.....] = Grid[.....]
   THEN WinningLine ← TRUE
   ELSE
      i ← 0
      .....
         i ← i + 1
           // check a row
         IF   Grid[i,1] = Grid[i,2] AND Grid[i,1] = Grid[i,3]
           // check a column
           OR (Grid[.....] = Grid[.....]
             AND Grid[.....] = Grid[.....])
           THEN WinningLine ← TRUE
         ENDIF
      UNTIL WinningLine = TRUE OR .....
.....
RETURN WinningLine
ENDFUNCTION
```

[10]

(f) The subroutine `DisplayGrid` is to output the state of play at any time.

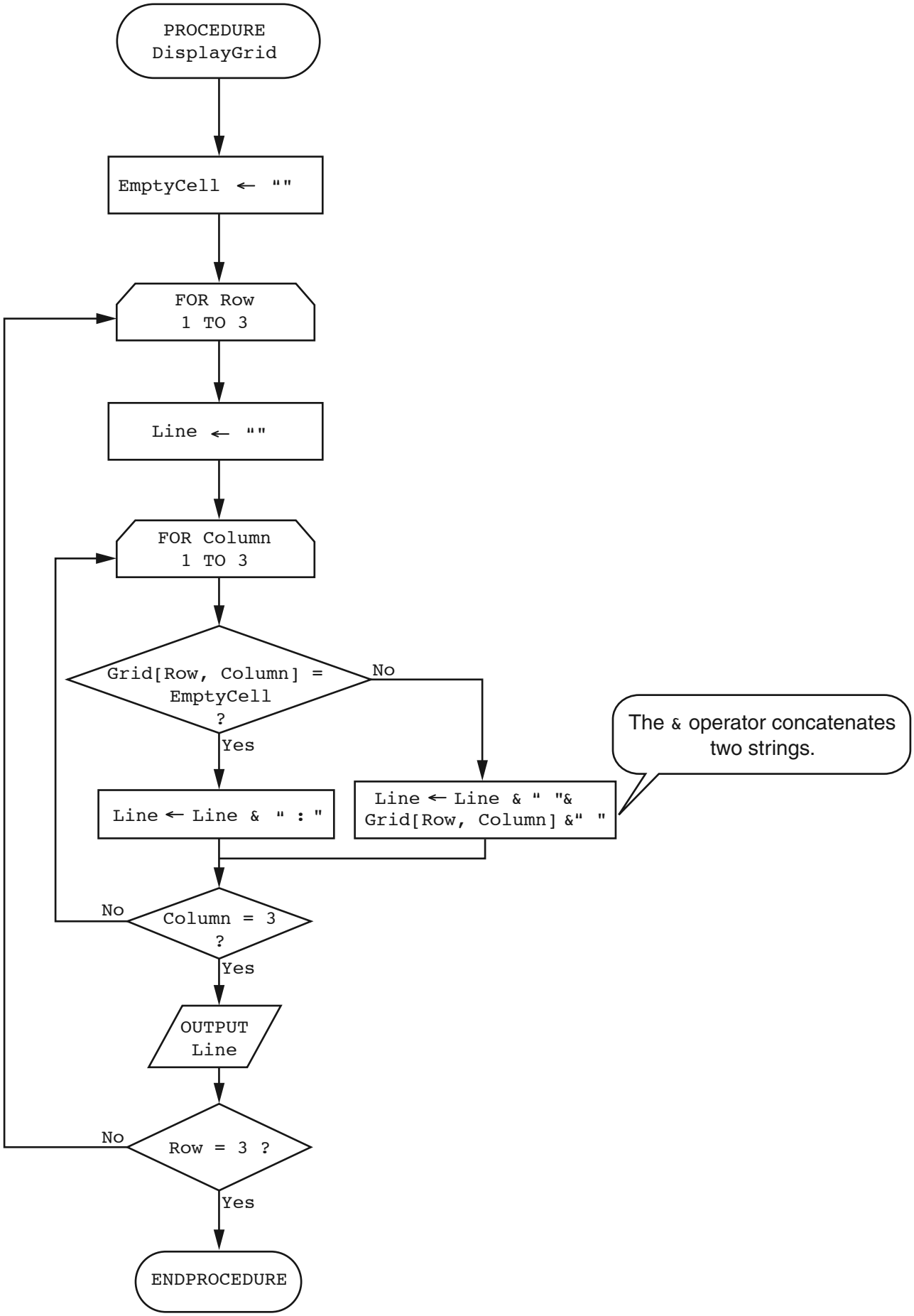
For example, after three turns the display should look like:

```
:  :  O
:  O  X
:  :  :
```

where the character ':' shows an empty cell.

Question 3(f) continues on page 12.

Write **program code** for the subroutine algorithm represented by the flowchart:



Programming language .....

Code .....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(g) Design a suitable form-based screen interface for the current player to input the row number and column number to place their marker when it is their turn.

[4]

(h) When Ali has tested all individual modules he plans to do further testing.

Give **two** types of testing Ali should do.

1.....

.....

2 .....

.....

[2]

BLANK PAGE

BLANK PAGE