UNIVERSITY OF CAMBRIDGE INTERNATIONAL EXAMINATIONS
General Certificate of Education
Advanced Subsidiary Level and Advanced Level

CANDIDATE
NAME

CENTRE
NUMBER

CANDIDATE
NUMBER

**COMPUTING** **9691/21**

Paper 2 **October/November 2011**

**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 a soft pencil for any diagrams, graphs or rough working.
Do not use staples, paper clips, highlighters, 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.

UNIVERSITY *of* CAMBRIDGE
International Examinations

**[Turn over**

1   Ahmed is writing a program to record the data of members of the school football
    The input data will need to be validated.  One input will be the position of each player
    he/she a forward, defender or goalkeeper?  These will be entered as f, d or g.
    The flowchart for the validation of the position is shown below.

**(a) (i)** What is the output when the input is 'g'?

................................................................................................................ [

**(ii)** What is the output when the input is 'x'?

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

**(b)** In a high-level language, write the code that will produce the validation process shown in the flowchart.

Language ................................................................................................................

Code ................................................................................................................

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

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

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

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

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

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

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

................................................................................................................ [5]

**(c)** The three basic programming constructs used to control the flow of information are: sequence, selection and iteration.

State the **two** constructs that are used in your code.

1 ................................................................................................................

2 ................................................................................................................ [1]

**(d)** Describe what is meant by iteration.

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

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

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

................................................................................................................ [2]

**(e)** Ahmed thinks it will be a good idea to allow only three attempts at getting the input correct.  If it is not a valid entry after three attempts, a message 'Please check what values are allowed' should be output.

*For ~~iner's~~*

Modify the flowchart to include this additional check.



[5]

*For*
* miner's*
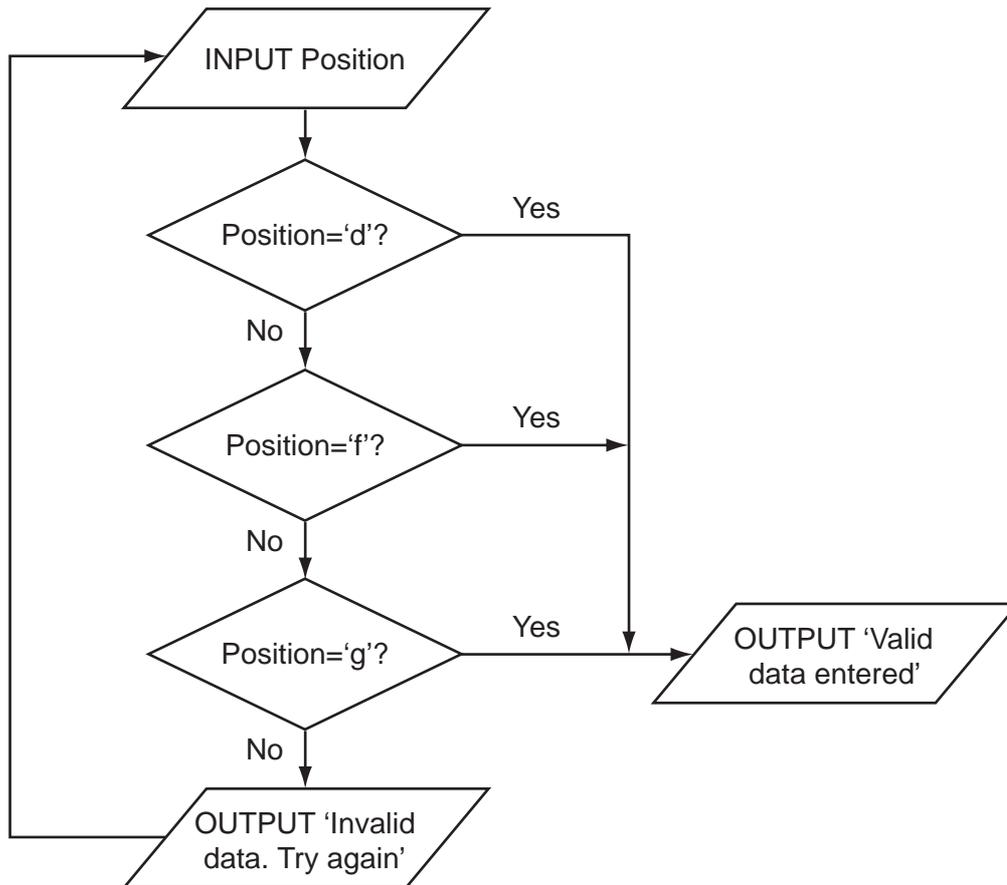
**(f)** Ahmed needs to store more information about the players.  He creates a
structure that contains PlayerID (a whole number between 1 and 50), Sex (m o
PlayerName, Position (f, d or g), and DateOfBirth.

Complete the table.

| Field Name | Data Type | Field Size (bytes) |
|---|---|---|
| PlayerID | | |
| Sex | | |
| PlayerName | | |
| Position | | |
| DateOfBirth | | |

[10]

**(g)** The squad has 30 players. Ahmed stores the records in an array called Squa... calculate how many players are defenders he designs this pseudocode:

```
Dtotal ← 0
ArrayPosition ← 1
REPEAT
    IF Squad[ArrayPosition].Position = 'd'
        THEN
                Dtotal ← Dtotal + 1
    ENDIF
    ArrayPosition ← ArrayPosition + 1
UNTIL ArrayPosition = 30
```

This pseudocode will only consider the first 29 records in the array.

**(i)** State the name of this type of error.

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

**(ii)** State the line that needs changing.

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

**(iii)** Re-write this line to ensure the pseudocode will consider all 30 records.

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

**(h)** Re-write this updated pseudocode using a WHILE loop ensuring that it will check all records in the array.

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

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

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

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

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

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

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

.......................................................................................................................... [3]

**2** Nathan is designing a software solution for stock control in a mobile phone shop. H~~e~~ colleague, called Andre, who will help him write the program. Nathan decide~~s~~ modularise the solution.
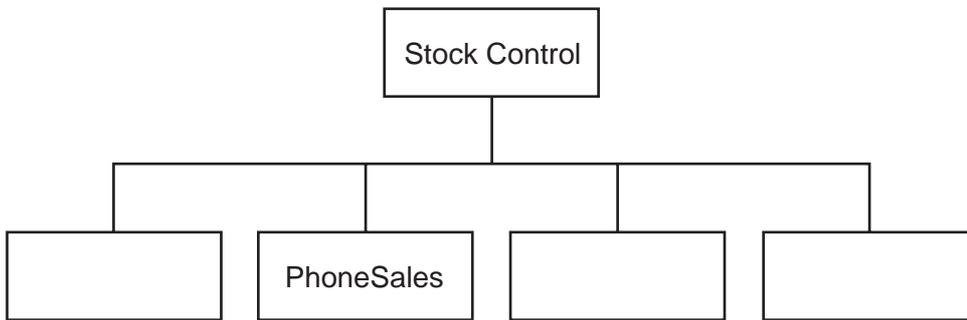
(a) State why modularisation is a good idea.

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

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

(b) As the first step in his design he splits the solution into the following main areas:

Initialisation, PhoneSales, StockOrdering, Accounts.

Complete the following structure diagram.

```
                        ┌─────────────────┐
                        │  Stock Control  │
                        └────────┬────────┘
          ┌──────────────┬───────┴───────┬──────────────┐
    ┌──────────┐   ┌──────────┐    ┌──────────┐    ┌──────────┐
    │          │   │PhoneSales│    │          │    │          │
    └──────────┘   └──────────┘    └──────────┘    └──────────┘
```

[1]

(c) PhoneSales is made up of two modules, ShopSales and OnlineSales. Add them to the structure diagram shown in **(b)**. [2]

(d) Nathan will write the ShopSales module and Andre will write the OnlineSales module. Nathan will use the identifier *Sale* for a sale in the shop, and Andre will use the identifier *Sale* for an online order.

Explain how they can both use the same identifier and not cause a problem when the program is run.

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

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

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

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

..................................................................................................................... [2]

For
iner's

**(e)** Both programmers need to choose other identifiers that they will use.

    **(i)** Explain why there are some words that cannot be used as identifiers.

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

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

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

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

    **(ii)** State **three** other rules of a high-level programming language that restrict the choice of identifiers.

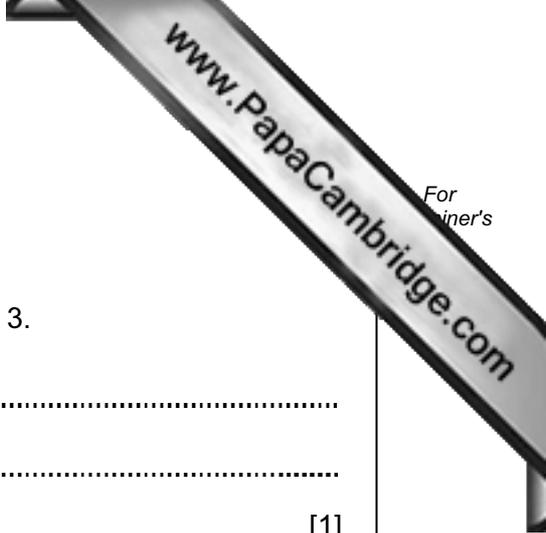        Language ...............................................................................................................

        Rule 1 ...............................................................................................................

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

        Rule 2 ...............................................................................................................

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

        Rule 3 ...............................................................................................................

        .................................................................................................................... [3]

    **(iii)** Give an example of an **invalid** identifier.

        Language ...............................................................................................................

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

For
iner's

**(f)** One line in the program reads:

ForecastStock = 5*a-b.

**(i)** Work out the value of ForecastStock when a is 4 and b is 3.

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

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

.................................................................................................................... [1]
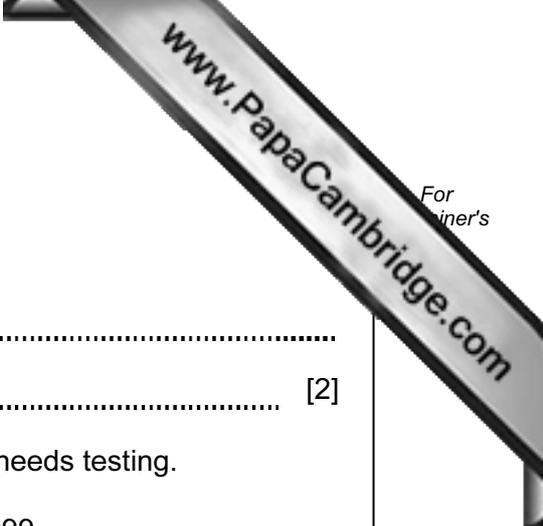
**(ii)** The programmer has made an error in the code. When a is 4 and b is 3 the expected result is 5.

Rewrite the line of code with added parentheses to give the expected result.

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

**(iii)** Name the type of testing strategy which identified this error.

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

**(g)** One type of test data is invalid data.

   **(i)** Name the other **two** types.

     1 ...............................................................................................................................

     2 .................................................................................................................. [2]

   **(ii)** Andre has written the StockOrdering module, which now needs testing.

- The StockID is a whole number between 1000 and 9999
- The ReOrderLevel is between 10% and 20%

     Give **six** different items of test data, other than invalid data, which thoroughly test the two rules given above. Give a reason for each choice.

| StockID | ReOrderLevel | Reason |
|---------|--------------|--------|
| 50 | | Invalid data for StockID |
| | 21% | Invalid data for ReOrderLevel |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

[6]
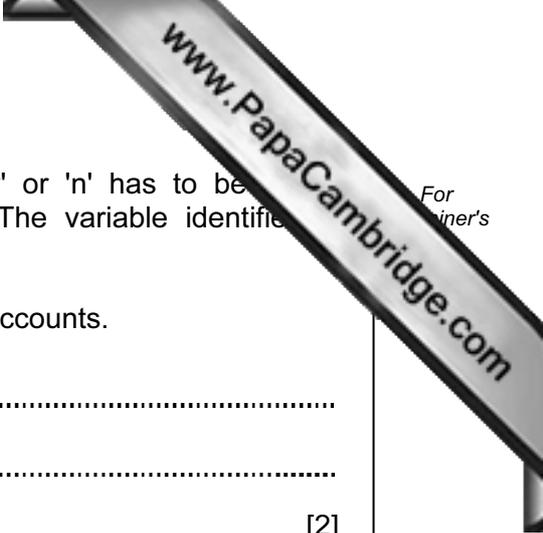
**(h) (i)** When dealing with the Accounts module, an entry, 'y' or 'n' has to be █ indicating whether the accounts should be printed. The variable identifi█ PrintAccounts.

Write a Boolean expression to validate an input for PrintAccounts.

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

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

................................................................................................................................ [2]

**(ii)** Write a Boolean expression to validate StockID, as described in (g).

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

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

................................................................................................................................ [2]

**(iii)** Describe a possible problem that could occur if the StockID is not validated.

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

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

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

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

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

................................................................................................................................ [2]

For
...iner's

**3** Aisha is learning about manipulating strings in a high-level programming language.
She has an idea that she wants to try. She produces the following design in pseudocod...

```
FUNCTION Surprise(MyWord : STRING): STRING
   LOOP FOR Counter ← 1 TO LENGTH(MyWord)
      Temp[Counter] ← MyWord[LENGTH(MyWord) – (Counter-1)]
   ENDLOOP
   Surprise ← Temp
ENDFUNCTION
```

**(a)** Complete the trace of this function for the function call `Surprise('rac')`.

| MyWord | Counter | LENGTH (MyWord) | Temp[1] | Temp[2] | Temp[3] | Surprise |
|--------|---------|-----------------|---------|---------|---------|----------|
| rac |  |  |  |  |  |  |
|  | 1 |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

[3]

**(b)** State the purpose of this function.

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

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

**(c)** Describe what `Surprise ← Temp` does.

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

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

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

................................................................................................................................................................ [2]
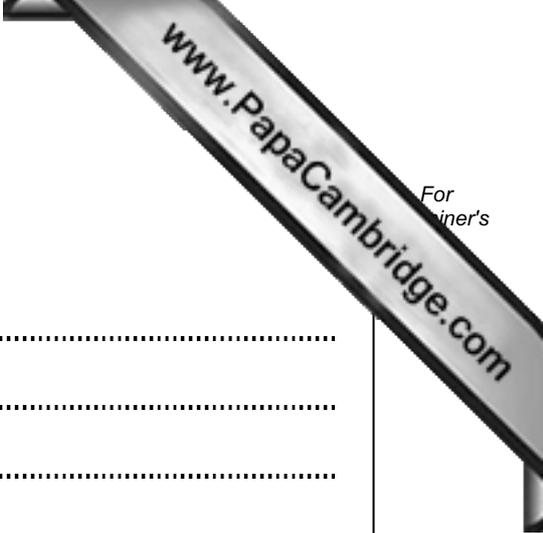
**(d)** High-level languages also use procedures.

Describe the features of a procedure.

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

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

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

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

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

............................................................................................................................... [3]

**(e)** Aisha has written her pseudocode so that it should be easily understood.

**(i)** State **two** techniques that she has used in her pseudocode to do this.

1 ...............................................................................................................................

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

2 ...............................................................................................................................

............................................................................................................................... [2]

**(ii)** One other technique to help understanding is to annotate using comments.

Write a suitable comment to annotate the line:

```
Temp[Counter] ← MyWord[LENGTH(MyWord) – (Counter-1)]
```

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

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

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

............................................................................................................................... [2]

**(f)** Aisha uses an expression:

```
String1 > String2
```

Explain how strings are compared by the processor.

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

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

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

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

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

................................................................................................................................ [3]

**BLANK PAGE**

**BLANK PAGE**