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

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER					

3626721815

COMPUTING **9691/22**

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 **12** printed pages.

IB11 11_9691_22/2RP

UNIVERSITY of CAMBRIDGE
International Examinations

1 Nathan is designing a software solution for stock control in a mobile phone shop. He has a colleague called Andre who will help him write the program. Nathan decides to modularise the solution.

(a) State why modularisation is a good idea.

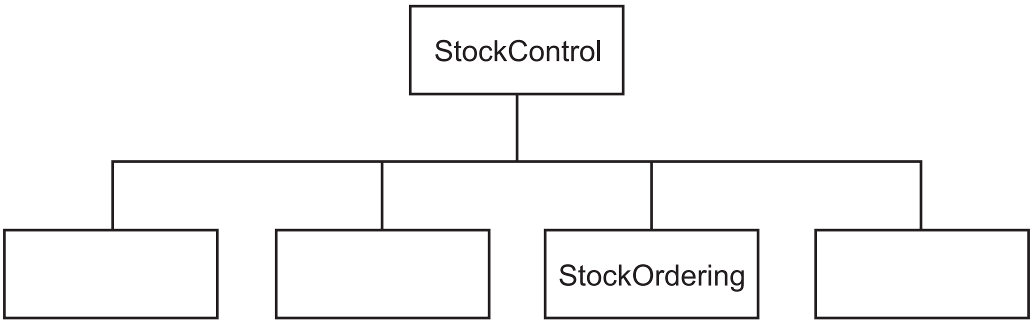
....

.... [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.



[1]

(c) StockOrdering is made up of two modules, PlaceOrder and ReceiveGoods.

Add them to the structure diagram in **(b)**. [2]

(d) Nathan will write the PlaceOrder module and Andre will write the ReceiveGoods module. Nathan will use the identifier *OrderValue* for the value of an order placed. Andre will use the identifier *OrderValue* for the total value of the received goods.

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

....

....

....

.... [2]

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

....

.... [1]

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

Language ....

Invalid identifier .... [1]

(iii) Both programmers need to choose other identifiers that they use.

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

Language ....

Rule 1 ....

....

Rule 2 ....

....

Rule 3 ....

.... [3]

(f) One line in the program reads

SalesTax = a – b * 0.1

(i) Work out the value of this expression when a is 600 and b is 200.

....

.... [1]

(ii) The programmer has made an error in the code. When a is 600 and b is 200 the expected answer is 40.

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) Nathan has written the PhoneSales module, which now needs testing.

- ContractLength, the number of months of the contract, can be only 12, 18 or 24
- FreeTexts, the number of free text messages per month must be in the range 0 to 600

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.

ContractLength	**FreeTexts**	**Reason**
20		Invalid data for ContractLength
	1000	Invalid data for FreeTexts

[6]

(h) (i) Write a Boolean expression to validate ContractLength.

....

....

....

.... [2]

(ii) Write a Boolean expression to validate FreeTexts.

....

....

.... [2]

(iii) Describe a possible problem that could occur if ContractLength is not validated.

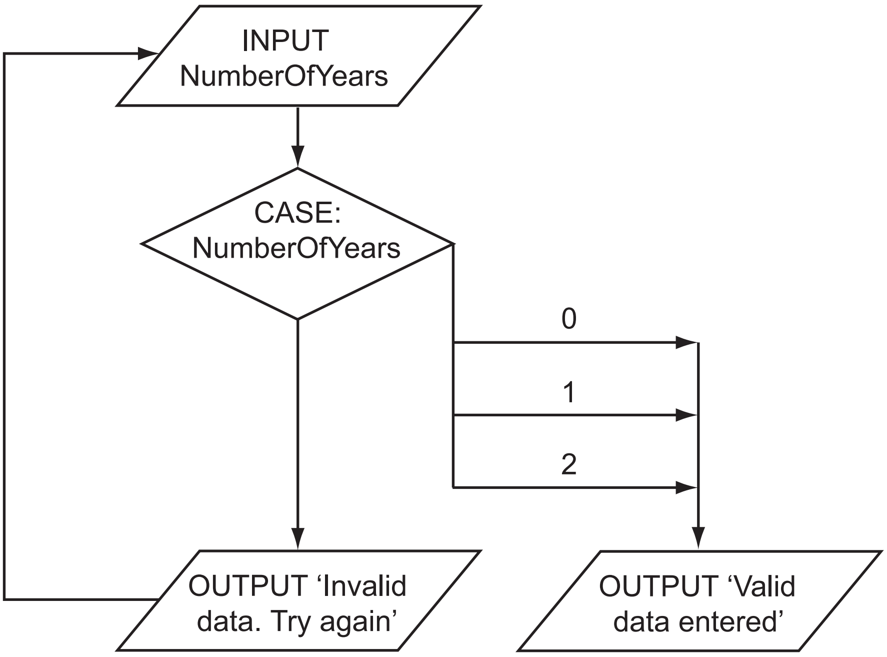
....

....

....

.... [2]

2 Ahmed is writing a program to record the data of members of the school football squad. The input data will be validated. One input is the number of years a member has played for the team. This will be 0, 1 or 2.
The flowchart for the validation of number of years is shown below.



(a) **(i)** What is the output when the input is 2?

.... [1]

(ii) What is the output when the input is 3?

.... [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.

....

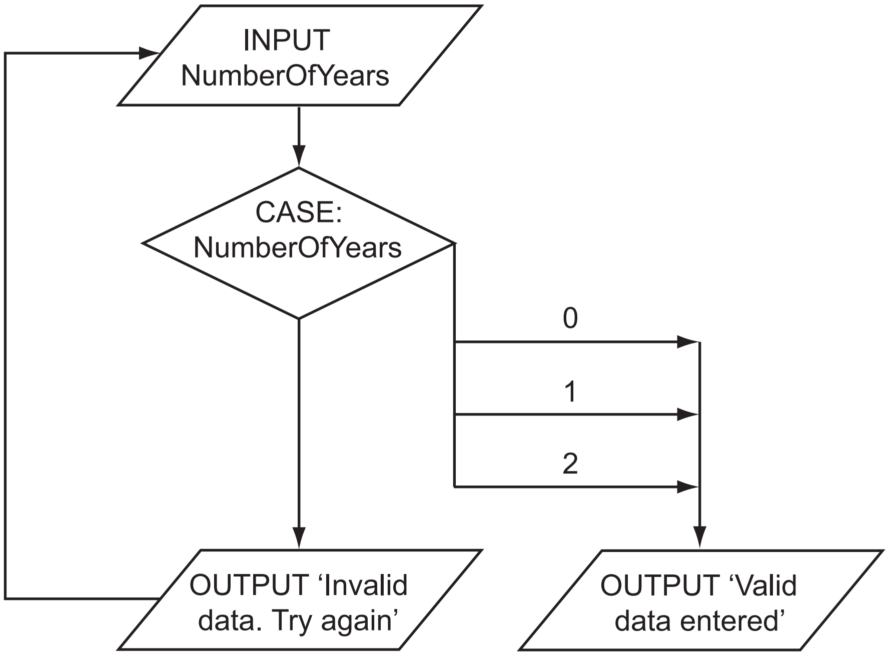
....

....

.... [2]

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

Modify the flowchart to include this additional check.



[5]

(f) Ahmed needs to store more information about the players. He creates a record structure that contains PlayerID (a whole number between 1 and 50), Sex (m or f), PlayerName, NumberOfYears and DateOfBirth. Complete the table.

Field Name	Data Type	Field Size (bytes)
PlayerID		
Sex		
PlayerName		
NumberOfYears		
DateOfBirth		

[10]

(g) The squad has 30 members. Ahmed stores the records in an array called Squad. To calculate how many females there are he designs this pseudocode.

```
NoOfFemales ← 0
Index ← 1
WHILE Index < 30
    IF Squad[Index].Sex  = 'f'
        THEN
            NoOfFemales ← NoOfFemales + 1
    ENDIF
    Index ← Index + 1
ENDWHILE
```

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) Rewrite the line to ensure that the pseudocode will consider all 30 records.

.... [1]

(h) Write this updated pseudocode using a FOR loop ensuring that it will check all records in the array.

....

....

....

....

....

.... [3]

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 pseudocode:

```
FUNCTION Surprise(MyWord : STRING) : STRING
    LOOP FOR i ← 1 TO LENGTH(MyWord)
        TempNo ← ASCII(MyWord[i]) + 1
        TempWord[i] ← CHAR(TempNo)
    ENDLOOP
    Surprise ← TempWord
ENDFUNCTION
```

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

MyWord	**i**	**LENGTH (MyWord)**	**TempNo**	**TempWord[1]**	**TempWord[2]**	**TempWord[3]**	**Surprise**
cab							
	1						
		3					
			100				

[3]

(b) State what this function does.

....

.... [1]

(c) Describe what `Surprise ← TempWord` 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 lines:

```
TempNo ← ASCII(MyWord[i]) + 1
TempWord[i] ← CHAR(TempNo)
```

....

....

....

....

....

.... [3]

(f) Aisha finds out that there is a string manipulation function called `CONCATENATION`.

State what is meant by concatenation.

....

....

....

.... [2]

For Examiner's Use