UNIVERSITY OF CAMBRIDGE INTERNATIONAL EXAMINATIONS
General Certificate of Education Advanced Level

CANDIDATE NAME			
CENTRE NUMBER		CANDIDATE NUMBER	

3270941184

COMPUTING **9691/32**

Paper 3 **May/June 2013**

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 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.
No marks will be awarded for using brand names for 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.

This document consists of **20** printed pages.

IB13 06_9691_32/FP

UNIVERSITY of CAMBRIDGE
International Examinations

For
iner's

1 A hospital has a number of doctors and a number of wards. A ward has a num patients. A patient may be treated by more than one doctor and a doctor treats m patients.

Data is to be recorded in a relational database and the tables include DOCTOR and PATIENT.

(a) **(i)** What is the relationship between DOCTOR and PATIENT?

.... [1]

(ii) Show this relationship with an entity-relationship (E-R) diagram.

[1]

(iii) Draw an E-R diagram showing a database design which can be produced so that the doctor and patient data are fully normalised.

Explain how the relationships are implemented.

....

....

....

.... [4]

(b) **(i)** What is the relationship between WARD and PATIENT?

.... [1]

(ii) Show this relationship with an E-R diagram.

[1]

(c) Two of the incomplete table designs are:

```
PATIENT(PatientID, PatientName, Address, NextOfKin)
WARD(WardName, NurseInCharge, NumberOfBeds,
                              NumberOfOccupiedBeds, Specialism)
```

Explain how the relationship between PATIENT and WARD is implemented.

....

....

....

.... [2]

(d) The following Data Manipulation Language query is run.

```
SELECT WardName
FROM WARD
WHERE NumberOfOccupiedBeds < NumberOfBeds
```

What useful information is produced for the Hospital Administrator?

....

....

....

.... [2]

2 **(a)** Explain the use of Backus-Naur Form (BNF) in computer science.

....

....

....

.... [2]

(b) A set of BNF rules are defined as follows:

```
1. <BinaryDigit>  ::= 0 | 1
2. <Parentheses>  ::= ‟
3. <Binary>       ::= <BinaryDigit> | <BinaryDigit> <Binary>
4. <BinaryString> ::= <Parentheses> <Binary> <Parentheses>
```

(i) A BNF rule can be recursive.

Explain what is meant by recursive.

....

.... [1]

(ii) State the rule above which is recursive.

Rule number ....is recursive. [1]

(iii) For each expression state whether it represents a valid or invalid binary string. List the rule number(s) in the order you have applied them to arrive at your decision.

Expression	**Valid/Invalid**	**Rules used**
0		
‟1″		
‟001″		

[7]

(c) The rules used in **(b)** are to be extended to allow a binary string to start w
character.
For example "$010" is a valid binary string.

Rewrite the set of rules to allow this additional format.

....

....

....

....

....

.... [2]

3 The table shows the assembly language instructions for a processor which has one g purpose register – the Accumulator (ACC), and an index register (IX).

Instruction		**Explanation**
Op Code	**Operand**	
`LDD`	`<address>`	Direct addressing. Load the contents of the given address to ACC
`STO`	`<address>`	Store the contents of ACC at the given address
`LDI`	`<address>`	Indirect addressing. At the given address is the address to be used. Load the contents of this second address to ACC
`LDX`	`<address>`	Indexed addressing. Form the address as `<address>` + the contents of IX. Copy the contents of this address to ACC
`INC`	`<register>`	Add 1 to the contents of the register (ACC or IX)
`ADD`	`<address>`	Add the contents of the given address to the contents of ACC
`OUT`		Output the contents of ACC (as a denary number) to the monitor
`IN`		Input a denary number from the keyboard and store in ACC
`END`		End the program and return to the operating system

The diagram shows a program loaded in main memory starting at location 100.

Two of the op-codes have been partially blanked out.

Locations 200 onwards contain data which is used by the program.

(a) The instruction at address 100 is fetched. Shown are the contents of the registers after execution.

ACC	IX
88	2

100	`LD■ 202`
101	`INC ACC`
102	`INC ACC`
103	`LD■ 203`
104	`INC ACC`
105	`LDI 203`
106	`INC ACC`
107	`END`
200	38
201	205
202	88
203	200
204	48
205	126

Which mode of addressing was used by this load instruction at address 100?

.... [1]

(b) The instruction at address 103 is fetched. Shown are the contents of the registers after execution.

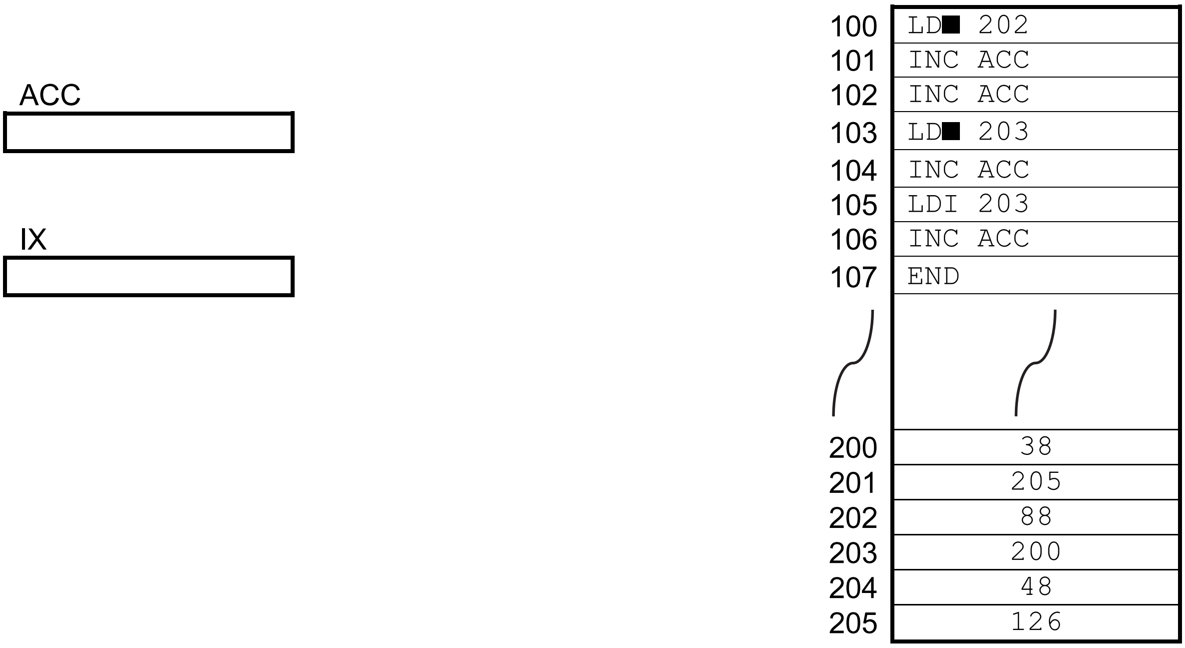
ACC	IX
126	2

Which mode of addressing was used by this load instruction at address 103?

.... [1]

(c) The instruction at address 105 is fetched and executed.

Draw on the diagram to explain how this instruction is executed and show the conten of ACC after execution.



100	LD■ 202
101	INC ACC
102	INC ACC
103	LD■ 203
104	INC ACC
105	LDI 203
106	INC ACC
107	END
200	38
201	205
202	88
203	200
204	48
205	126

[2]

(d) Trace this assembly language program using the trace table below.
The user inputs 19 followed by 37.

```
100     IN
101    INC  ACC
102    STO  109
103     IN
104    INC  ACC
105    ADD  109
106    STO  110
107    OUT
108    END
```

ACC	Memory location		Output
	109	110	

[5]

(e) In **(d)** the program was shown in assembly language. In practice this must be machine code in order to execute the program.

Explain how the assembler software translates a program from assembly language into machine code.

....

....

....

....

....

.... [3]

4 A binary tree is implemented with three 1-dimensional arrays.

Identifier	Data Type	Description
`Data`	`ARRAY[100] OF STRING`	Stores the data values
`LeftP`	`ARRAY[100] OF INTEGER`	Stores the left index pointer
`RightP`	`ARRAY[100] OF INTEGER`	Stores the right index pointer
`Root`	`INTEGER`	Stores the index position of the root value

(a) An array is a static data structure.

(i) Explain the difference between a static and a dynamic data structure.

....

....

....

.... [2]

(ii) What benefit would be gained from using a dynamic data structure to implement a binary tree?

....

.... [1]

The initially empty tree has the following items added in this order:

`MELON, PEAR, BANANA, ORANGE`

(b) Draw the binary tree after these four items have been added.

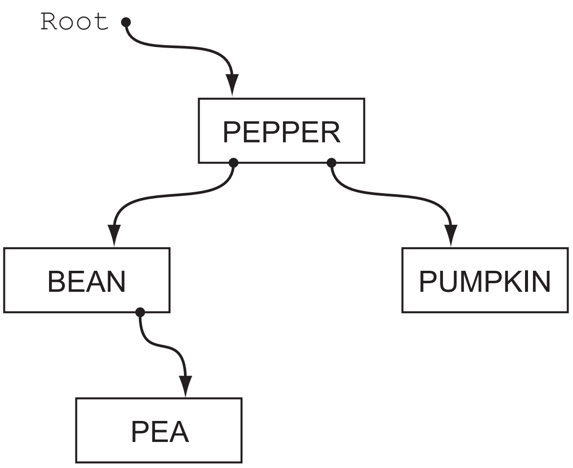
[3]

(c) The following algorithm traverses the tree shown and outputs the nodes in orde
'in-order traversal'.

```
PROCEDURE InOrder(Root)
  IF LeftP[Root] <> 0
    THEN
      // move left
      InOrder(LeftP[Root])
  ENDIF

  OUTPUT Data[Root]

  IF RightP[Root] <> 0
    THEN
      // move right
      InOrder(RightP[Root])
  ENDIF
ENDPROCEDURE
```

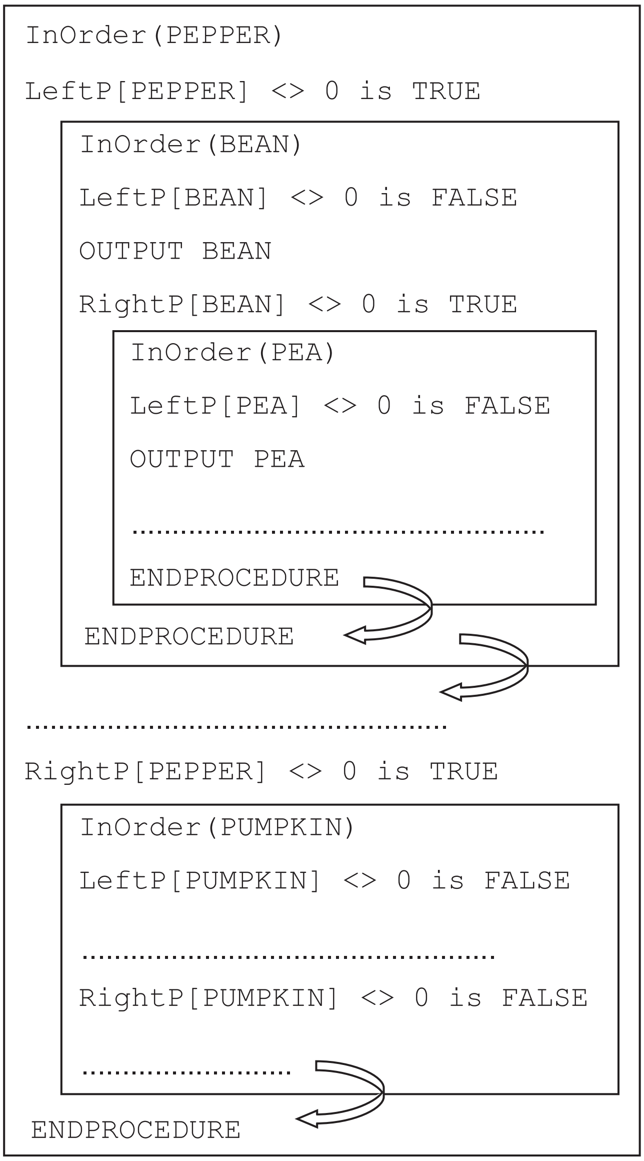


(i) Copy a line from procedure `InOrder` that makes the procedure recursive.

.... [1]

(ii) The diagram shows a trace of the execution of this algorithm for the give data.

Fill in the missing lines of code.



```
InOrder(PEPPER)
LeftP[PEPPER] <> 0 is TRUE
    InOrder(BEAN)
    LeftP[BEAN] <> 0 is FALSE
    OUTPUT BEAN
    RightP[BEAN] <> 0 is TRUE
        InOrder(PEA)
        LeftP[PEA] <> 0 is FALSE
        OUTPUT PEA

        ..................................................
        ENDPROCEDURE
    ENDPROCEDURE

..................................................
RightP[PEPPER] <> 0 is TRUE
    InOrder(PUMPKIN)
    LeftP[PUMPKIN] <> 0 is FALSE

    ..................................................
    RightP[PUMPKIN] <> 0 is FALSE

    .........................
ENDPROCEDURE
```

[4]

(iii) What do the arrows in the diagram represent?

....

.... [1]

For
iner's

5 The following are the first few lines of a source code program written in high-level la
XYZ which is about to be translated by the language compiler.

```
// program written 12 June 2013
Declare IsFound : Boolean;
Declare NoOfChildren : Integer;
Declare Count : Integer;
Constant TaxRate = 15;

// start of main program
For Count = 1 To 50
…
…
…
```

(a) During the lexical analysis stage the compiler will use a keyword table and a symbol table.

(i) Describe what information is contained in these tables.

Keyword table ....

....

....

Symbol table ....

....

.... [2]

(ii) Explain how the table contents are used to translate the source code.

....

....

....

.... [2]

(iii) Describe **one** other task done at the lexical analysis stage which does not involve the use of the keyword table or symbol table.

....

.... [1]

(b) The final stage of compilation is code optimisation.

(i) Explain what is meant by code optimisation.

....

....

....

.... [2]

(ii) Give **one** practical example of code which would benefit from optimising.

....

.... [1]

6 A multiprogramming, multi-user operating system organises the available memory into two fixed sized partitions.

- Partition 1 – size 30 Gb – is used only for batch processing
- Partition 2 – size 40 Gb – is used only for interactive processing at a terminal

A program, once loaded, occupies the same memory locations until its execution is complete.

(a) Explain what is meant by:

(i) Batch processing

....

....

....

....

....

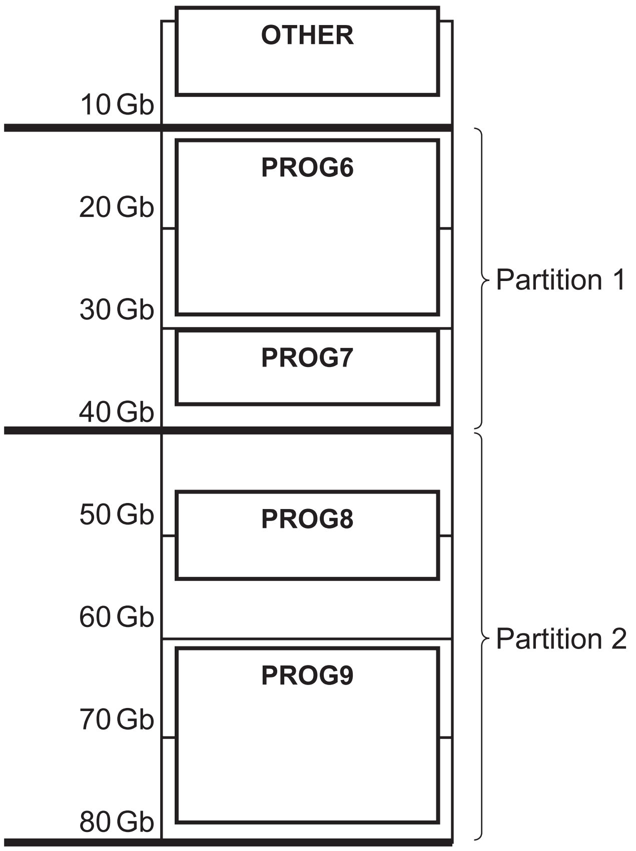
.... [3]

(ii) Interactive processing

....

.... [1]

(b) The diagram shows the current contents of main memory with a list of pro
waiting to be scheduled.



Currently waiting to be loaded are:

- PROG16 (batch processing requiring 25 Gb)
- PROG17 (interactive processing requiring 8 Gb)
- PROG44 (interactive processing requiring 9 Gb)
- PROG45 (interactive processing requiring 8 Gb)
- PROG23 (batch processing requiring 15 Gb)

(i) If **PROG6** completes execution, which programs (if any) can be loaded next?

.... [1]

(ii) If **PROG8** completes execution, which programs (if any) can be loaded next?

.... [1]

(iii) Describe **two** variations on this use of partitions which may make more efficient use of memory.

1 ....

....

2 ....

.... [2]

For
iner's

(c) 10 Gb of the main memory is labelled OTHER and will not be used for the execu application programs.

Name **two** possible items of software this memory will be used for.

1 ....

2 .... [2]

(d) Any program loaded is always in one of three possible states. One is the ‘running’ state meaning the job currently has use of the processor.

Name and describe the **two** other states.

1 ....

....

2 ....

.... [4]

7 **(a)** A user-defined function `CountCharacter` is defined, using pseudocode, as follo

```
FUNCTION CountCharacter(ThisArray : CHAR, UBound : INTEGER,
                              ThisCharacter : CHAR) RETURNS INTEGER
```

The function checks each element in the array `ThisArray` with upper bound `UBound`.
The function returns the number of times `ThisCharacter` appears in `ThisArray`.
If the function is incorrectly formed it will give a ‘COMPILE ERROR’.

The function is used with the three arrays shown below:

	Subscript/Index									
Identifier	1	2	3	4	5	6	7	8	9	10
`City1`	L	O	N	D	O	N				
`City2`	N	E	W	<Space>	Y	O	R	K		
`City3`	S	T	O	C	K	H	O	L	M	

What is returned by the following function calls?

(i) `CountCharacter(City1, 6, ‘O’)`

.... [1]

(ii) `CountCharacter(City2, ‘R’, 8)`

.... [1]

(iii) `CountCharacter(City3, 9, ‘t’)`

.... [1]

(iv) `CountCharacter(City2, ’W’)`

.... [1]

For iner's

(b) A programmer writes pseudocode to report whether or not a string value `ThisVa` found in the string array `ThisArray`. This is designed as a function `StringFo` which returns a Boolean value.

(i) Show the function header for function `StringFound`.

....

.... [3]

(ii) Use the following numbers to label your function header above.

1. Function parameter(s)
2. Where the return data type is stated [2]

(iii) Write a single statement which calls the function `StringFound` to search for the value "LISBON" in the array `CapitalCities` (upper bound 300). The function returns the value to variable `CityWasFound` in the calling statement.

....

.... [2]

8 (a) A robot is a moveable, mechanical device that can sense its surroundings controlled by a computer program.

(i) Give an example of a use of a robot.

....

.... [1]

(ii) Explain how your example fits with the given description of a robot.

....

....

....

.... [2]

(b) Name **two** hardware components that would be found in the example given in **(a)(i)**. Explain how each hardware component is used in the operation of the robot.

1 ....

....

....

2 ....

....

.... [4]

(c) What type of operating system would be used for any robot?

.... [1]