UNIVERSITY OF CAMBRIDGE INTERNATIONAL EXAMINATIONS
General Certificate of Education Advanced Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER					

7503710109

COMPUTING **9691/33**

Paper 3 **May/June 2012**

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 **16** printed pages.

IB12 06_9691_33/3RP

UNIVERSITY of CAMBRIDGE
International Examinations

1 A database is designed to store data about all aircraft owned by an airline and the flight schedules.
The following table `AircraftSchedule` was a first attempt at part of the database design.

Table: `AircraftSchedule`

AircraftID	Type	YearBought	FlightCode	Departure	Arrival
1	747	1998	2032	Delhi	Singapore
			1187	Singapore	Melbourne
			1326	Melbourne	Tokyo
			1556	Tokyo	Delhi
2	747-400	2007	1426	Bristol	Amsterdam
			1427	Amsterdam	Bristol
			5564	Bristol	Rome
			7865	Rome	Istanbul
3	747-400	2007	1090	London	New York
			1165	New York	Boston

(a) (i) Explain why the table is not in First Normal Form (1NF).

....

.... [1]

(ii) Explain your answer in terms of the data above.

....

.... [1]

(b) The design is changed to the following:

```
Aircraft (AircraftID, Type, YearBought)
Schedules (FlightCode, Departure, Arrival)
```

Using the data given in the original table:

(i) Show what data is now stored in the table `Aircraft`.

Table: `Aircraft`

AircraftID	Type	YearBought

[1]

(ii) How many records are now stored in table `Schedules`?

.... [1]

(c) **(i)** Explain what is meant by a primary key.

....

....

....

.... [2]

(ii) What is the primary key of table `Aircraft`?

.... [1]

(d) **(i)** Explain what is meant by a foreign key.

....

....

....

.... [2]

(ii) State what foreign key needs to be added to the `Schedules` table design.

.... [1]

(e) An additional table `Airport` is designed as shown:

`Airport (`<u>`AirportName`</u>`, Country, NoOfRunways)`

Explain why this table is in Third Normal Form (3NF).

....

....

....

.... [2]

(f) The normalisation process is designed to eliminate data inconsistency.

Explain what is meant by data inconsistency.

....

.... [1]

For Examiner's Use

2 A binary pattern can be used to represent a variety of different data used in a computer system.

(a) The pattern could represent an ASCII character code.
The table shows part of the ASCII code table.

ASCII Code Table

Character	Decimal	Character	Decimal	Character	Decimal
<Space>	32	I	73	R	82
A	65	J	74	S	83
B	66	K	75	T	84
C	67	L	76	U	85
D	68	M	77	V	86
E	69	N	78	W	87
F	70	O	79	X	88
G	71	P	80	Y	89
H	72	Q	81	Z	90

Consider the binary pattern: **0100 1110**.

(i) What character is represented by this binary pattern?

.... [1]

(ii) What is the hexadecimal for this binary pattern?

.... [1]

(b) (i) A computer system needs to be able to store positive and negative integers.
Two possible representations are:

- sign and magnitude
- two's complement.

Describe **two** advantages of using two's complement.

....

....

....

.... [2]

(ii) The integers -13 and +59 are to be added using two's complement addition.
Show your working.

-13									
+59									+

[3]

For Examiner's Use

(c) Real numbers are to be stored in floating point representation with:

- 8 bits for the mantissa, followed by
- 4 bits for the exponent
- two's complement used for both the mantissa and the exponent

(i) Consider the binary pattern:

1	**0**	**1**	**0**	**1**	**0**	**0**	**0**

0	**1**	**1**	**1**

What number is this in denary? Show your working.

....

....

....

....

....

.... [3]

(ii) Explain how you can recognise that the above pattern is normalised.

....

.... [1]

(iii) Show the binary pattern for the smallest negative number (negative sign and large magnitude) which can be stored using a normalised 12-bit floating point representation.

Mantissa:

Exponent:

Work out its denary value.

Denary: .... [3]

3 A linked list is to be implemented with the data structures described in the variable table. The countries are to be organised in alphabetical order.

Identifier	Data Type	Description
`Country`	`ARRAY[250] OF STRING`	Stores the country names
`Pointer`	`ARRAY[250] OF INTEGER`	Array index which points to the next country in the linked list
`HeadPointer`	`INTEGER`	Array index pointing to the first country in the linked list

`HeadPointer` □

	`Country`		`Pointer`
1	SWEDEN	1	
2	DENMARK	2	
3	INDIA	3	
4	COLUMBIA	4	
5	BANGLADESH	5	4
6	NEPAL	6	
7	MAURITIUS	7	
250		250	

(a) Complete the above diagram showing all the pointer values for this linked list. [4]

For Examiner's Use

(b) The following pseudocode uses the linked list to output all country names which are alphabetically before a requested country.
For example, the user inputs NEPAL – the pseudocode outputs all the values which are alphabetically before NEPAL.

Fill in the gaps in the pseudocode.

```
INPUT RequestedValue
IF ..........................................
    THEN
        //special case – the list is empty ...
        OUTPUT “Linked list is empty”
    ELSE

        ..........................................
        Current ← HeadPointer
        REPEAT
            IF Country[Current] < RequestedValue
                THEN
                    OUTPUT Country[Current]
                    Current ← ..........................................
                ELSE
                    NoMoreValues ← TRUE
            ENDIF
        UNTIL NoMoreValues = TRUE
```

[3]

(c) An algorithm is to be designed which inputs a requested country and outputs all the values in the linked list after this country.

Describe how, using the pointers, this algorithm works.

....

....

....

....

....

....

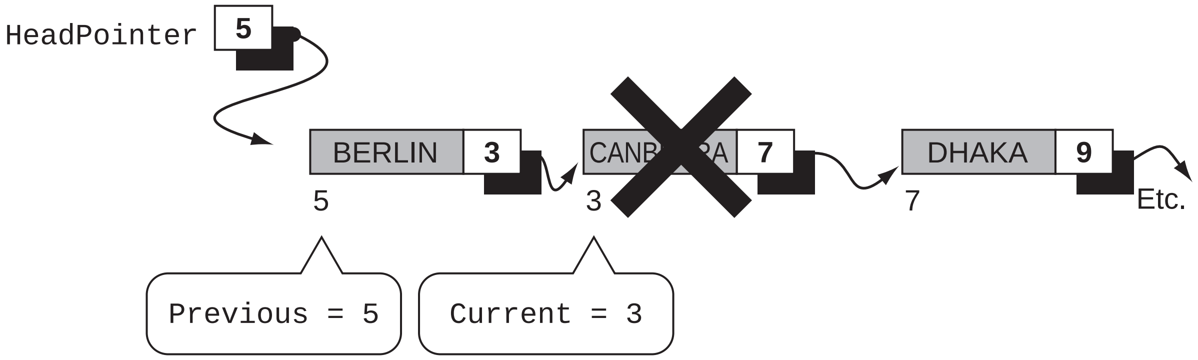
....

.... [4]

(d) A linked list is maintained for capital cities using arrays `Capital` and `Pointer`.
An algorithm is required to delete a value from the linked list.
The algorithm will use the following variables:

Identifier	Data Type	Description
`Current`	`INTEGER`	Array index for the current capital
`Previous`	`INTEGER`	Array index for the previous capital

The following diagram shows the first three capitals in the linked list.
We are about to delete CANBERRA.
The list has been searched from the `HeadPointer` position until the capital to be deleted, CANBERRA, is found.



Describe the steps in the algorithm to delete CANBERRA from the linked list.
(Do not attempt to write the complete algorithm.)

....

....

....

....

....

....

....

.... [4]

4 Expressions can be written in either infix or reverse Polish notation.

(a) Evaluate this reverse Polish expression:

`9 6 – 5 *`

.... [1]

(b) Write the following infix expressions in reverse Polish.

(i) `(c + 5)/(b – c)`

....

.... [1]

(ii) `3 * 9 – 6 / 2`

....

.... [2]

(c) Describe **one** benefit of storing an expression in reverse Polish.

....

.... [1]

(d) An expression in reverse Polish can be evaluated on a computer system using a stack.

(i) Describe the operation of a stack.

....

.... [1]

(ii) A stack is to be implemented as an array with an integer variable to point to the 'top of stack' index position.

State whether this is a static data structure or a dynamic data structure and explain why.

....

....

....

.... [2]

For Examiner's Use

(iii) The reverse Polish expression `4 7 * 2 + 5 /` is to be evaluated using a stack. The first available location on the stack is 1.

Show how the contents of the stack change as this expression is evaluated.

5							
4							
3							
2							
1							

[4]

5 **(a)** Define what is meant by the term computer simulation.

....

....

....

.... [2]

(b) Give **two** reasons why a computer system is particularly suited to carrying out a simulation.

1 ....

....

2 ....

.... [2]

(c) A new road has been built which crosses an existing road at right angles. A new set of traffic control lights is to be installed to control the traffic flows on the existing and new road.

Identify **three** variables which need to be controlled by the software simulation of the operation of the traffic lights.

1 ....

2 ....

3 .... [3]

(d) The values input to the simulation will affect the outputs produced.

Give **one** example for this traffic control light scenario of a change to an input which will directly affect the output.

Input change ....

....

Effect on the output ....

.... [2]

6 The table shows the assembly language instructions for a processor which has one general purpose register – the Accumulator.

Instruction		Explanation
Op Code	Operand	
`LDD`	`<address>`	Load using direct addressing
`STO`	`<address>`	Store the contents of the Accumulator at the given address
`LDI`	`<address>`	Load using indirect addressing
`LDX`	`<address>`	Load using indexed addressing
`INC`		Add 1 to the contents of the Accumulator
`END`		End the program and return to the operating system

(a) Write on the diagram to explain the assembly language instruction shown below. Show the contents of the Accumulator after the execution of the instruction.

```
LDD 66
```

Accumulator

□

	Main memory
60	0110 0000
61	0100 0000
62	1111 1110
63	1111 0000
64	0101 1101
65	0001 0001
66	1010 1000
67	1100 0001
200	1001 1111

[2]

(b) Write on the diagram to explain the assembly language instruction shown. Show the contents of the Accumulator after the execution of the instruction.

```
LDI 61
```

Accumulator

□

	Main memory
60	0110 0000
61	0100 0000
62	1111 1110
63	1111 0000
64	0101 1101
65	0001 0001
66	1010 1000
67	1100 0001
200	1001 1111

[3]

For Examiner's Use

(c) Trace this assembly language program using the given trace table. The first instruction of the program is loaded into main memory at address 200.

```
200     LDD  208
201     INC
202     STO  208
203     LDD  207
204     INC
205     STO  207
206     END
207      16
208     150
```

Accumulator

Memory Address	
207	208
16	150

[4]

(d) Explain the relationship between assembly language instructions and machine code instructions.

....

.... [1]

7 **(a)** Explain what is meant by an interrupt.

....

....

....

.... [2]

(b) An operating system uses interrupts which have priorities.

Describe the sequence of steps which would be carried out by the interrupt handler software when an interrupt is received and serviced.

....

....

....

....

....

....

....

....

....

....

....

.... [6]

(c) Modern personal computer operating systems support multi-tasking.
One of the modules of such an operating system will be for memory management.

Describe **two** different strategies which could be used to manage the available main memory.

1 ....

....

....

....

....

....

2 ....

....

....

....

....

.... [6]

(d) Once a process finishes and memory becomes available, the scheduler will decide which process/job is to be loaded next.

State **three** attributes of a process which are used to assess which job will be the next to be loaded into main memory.

1 ....

....

2 ....

....

3 ....

.... [3]