UNIVERSITY OF CAMBRIDGE INTERNATIONAL EXAMINATIONS
General Certificate of Education Advanced Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER					

9693515187

COMPUTING **9691/31**

Paper 3 **October/November 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 **16** printed pages.

IB13 11_9691_31/3RP

UNIVERSITY *of* CAMBRIDGE
International Examinations

[Turn over

1 **(a)** Convert the following infix expressions into reverse Polish notation:

(i) `(a + b) / 7`

....

.... [1]

(ii) `2 / (3 * Z + 5)`

....

.... [2]

(b) What is the value of this reverse Polish expression:

`x y + p q - /`

for `x = 3`, `y = 9`, `p = 5` and `q = 1`?

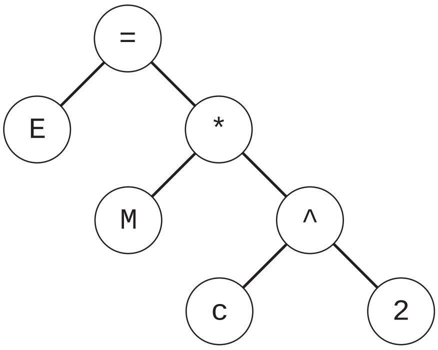
Show your working.

....

....

.... [2]

(c) A binary tree can be used to represent an expression or a statement.



The diagram shows the binary tree for the infix statement:

`E = M * c ^ 2`

(i) Explain how the infix form for this statement is produced using a tree traversal.

....

.... [1]

(ii) What is the reverse Polish notation for this statement?

....

.... [1]

(iii) Explain how the reverse Polish notation for the statement is produced using a tree traversal.

....

.... [1]

2 Cross country runners take part in races.

- A runner must be registered with one club only and club names are unique.
- A club has runners; each runner has a unique national MemberID.
- Each race is organised by a club and the Club Secretary records which runners are entered for each race.
- Runners may enter any race.
- There is only one race on any one day.

At present each club records the data for the competition races it organises. The data is stored in flat files.

(a) Describe **three** advantages that a relational database would have over the use of flat files.

1 ....

....

2 ....

....

3 ....

.... [3]

(b) **(i)** What is the relationship between runner and race?

.... [1]

(ii) What is the relationship between club and race?

.... [1]

(c) A database solution is to be developed.
Two of the tables are `RUNNER` and `RACE`.

(i) Draw an entity-relationship (E-R) diagram showing a database design which can be produced so that the runner and race data are fully normalised.

[2]

(ii) Explain how the relationships are implemented.

....

....

....

.... [2]

(d) The following table design is suggested for `RUNNER`.

`RUNNER(MemberID, RunnerName, RunnerDOB, ClubName, ClubAddress)`

This is poorly designed.

(i) Is this table in First Normal Form (1NF)?
Explain.

....

.... [1]

(ii) Is this table in Second Normal form (2NF)?
Explain.

....

.... [1]

(iii) The table is not in Third Normal Form (3NF).
Explain.

....

.... [1]

(iv) Using only the attributes given in the `RUNNER` table above, produce a new design which is fully normalised.

The table descriptions should be expressed as:

`TableName(Attribute1, Attribute2, Attribute3, ...)`

....

....

....

.... [2]

(e) Explain why all tables in the final design should be fully normalised.

....

....

....

.... [2]

(f) The table to store the race data has the following design:

`RACE(RaceDate, RaceStartTime, StartVenue, Distance, OrganisingClubName)`

Write a Data Manipulation Language (DML) query to report all races after the 1st January 2013 which are less than 10 km. Display the race date and organising club name only.
Use the keywords `SELECT`, `FROM`, `WHERE`.

....

....

....

....

....

.... [3]

3 **(a)** Most modern computers are designed using Von Neumann architecture.

Explain what is meant by Von Neumann architecture.

....

....

....

.... [2]

(b) **(i)** Convert the hexadecimal number 7A to denary.

.... [1]

(ii) Convert the binary number 0101 1100 to hexadecimal.

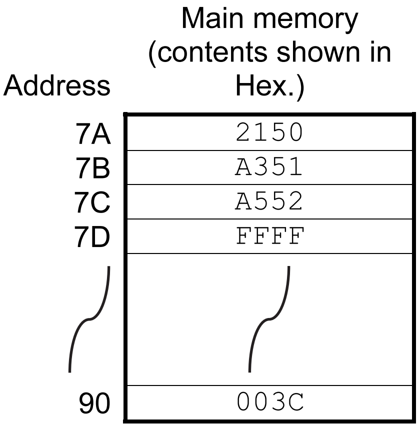
.... [1]

(iii) Why do computer scientists often write binary numbers in hexadecimal?

....

.... [1]

(c) The diagram shows a program loaded into main memory starting at memory address 7A Hex.



Address	Main memory (contents shown in Hex.)
7A	2150
7B	A351
7C	A552
7D	FFFF
90	003C

(i) How many bits are used for each main memory location?

.... [1]

The trace table below is used to show how the contents of the special-purpose registers change as the program is executed.
The steps in the fetch stage of the fetch-execute cycle are shown in the first column using register transfer notation. (For example, `MAR ← [PC]` means the content of the Program Counter is copied to the Memory Address Register.)

(ii) Complete the trace table for the fetching of the **first program instruction (2150)**:

- Show the changing contents of the registers
- Put a tick in the Address bus/Data bus column to show when the signals on that bus change.

Fetch stage	**Special purpose registers (Contents shown in Hex.)**				**Buses**	
	PC	MAR	MDR	CIR	Address bus	Data bus
	7A					
`MAR ← [PC]`						
`PC ← [PC] + 1`						
`MDR ← [[MAR]]`						
`CIR ← [MDR]`						

[5]

(d) The following table shows some of a processor's instruction set in assembly language.

Instruction		Explanation
Op Code	Operand	
`LDD`	`<address>`	Direct addressing. Load the contents of the given address to `ACC`
`LDI`	`<address>`	Indirect addressing. At the given address is the address to be used. Load the contents of this second address to `ACC`
`LIX`	`<address>`	Load the contents of the address to the Index register (`IX`)
`LDX`	`<address>`	Indexed addressing. Form the address as `<address>` + the contents of `IX`. Copy the contents of this address to `ACC`

The following program is to be executed. Shown are:

- the first four instructions only of this program
- the memory locations which are accessed by this program.

Address	Main memory
100	`LIX 200`
101	`LDD 201`
102	`LDI 201`
103	`LDX 201`
200	`3`
201	`216`
202	`99`
203	`217`
204	`63`
216	`96`
217	`97`

Complete the trace table below for the first **four** program instructions.
Show each change in the contents of the registers.

Instruction	Register	
	Accumulator (`ACC`)	Index Register (`IX`)
`LIX 200`		
`LDD 201`		
`LDI 201`		
`LDX 201`		

[4]

4 Object-oriented programming is one programming paradigm.

(a) Explain the difference between a class and an object.

....

....

....

....

....

.... [3]

(b) The following scenario is to be implemented with object-oriented programming.

A library has resources (`RESOURCE`) available for lending out to borrowers. Resources include books (`BOOK`), and recordings (`RECORDING`). Recordings are available for either films (`FILM`) or music (`MUSIC`) CDs.

Data stored will include:

- library ID for every item
- author for books
- release date for music CDs and films
- title for every available item
- number of tracks for CDs
- running time for films
- whether or not on loan

Complete the class diagram showing the classes and properties only for the data given above.

RESOURCE
LibraryID: INTEGER

[8]

(c) Explain what is meant by encapsulation.

....

....

....

.... [2]

5 **(a)** Describe the operation of a stack data structure.

....

.... [1]

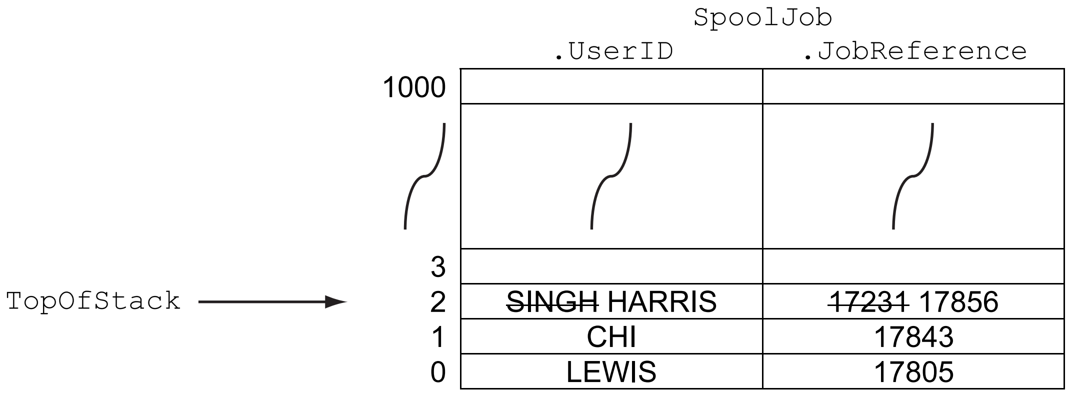
A stack is to be implemented to manage the spooled print jobs sent to a network printer. A job reference and the user ID of the network account are recorded for each print job.

The stack is implemented using the following user-defined data type and variables.

```
TYPE Stack
   JobReference : STRING
   UserID       : STRING
ENDTYPE
```

Identifier	Data Type	Description
`SpoolJob`	`ARRAY[1000] OF Stack`	Stores the job reference and user ID for each print job
`TopOfStack`	`INTEGER`	Stores the index position of the print job currently at the top of the stack
`NewReferenceNo`	`STRING`	Stores the job reference of the new print job added to `SpoolJob`
`NewUserID`	`STRING`	Stores the user ID of the new print job added to `SpoolJob`

(b) The diagram shows the state of `SpoolJob` and `TopOfStack` after three print jobs were received from users LEWIS, CHI and SINGH (in that order), a print job was sent to the printer, then a new print job received from user HARRIS.



	SpoolJob	
	.UserID	.JobReference
1000		
3		
2	~~SINGH~~ HARRIS	~~17231~~ 17856
1	CHI	17843
0	LEWIS	17805

(i) What is the value of:

`SpoolJob[2].UserID`? ....

`SpoolJob[TopOfStack - 1].JobReference`? .... [2]

(ii) Spooling a new print job is to be implemented with a procedure `PushJob`.

Shown below is the incomplete pseudocode for the `PushJob` procedure.

Using the variables and user-defined type given, fill in the missing pseudocode.

```
PROCEDURE PushJob

    IF .............................................................

       THEN

          OUTPUT "Stack is already FULL"

       ELSE

          INPUT NewUserID

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

          TopOfStack ← .............................................

          SpoolJob[TopOfStack].JobReference ← NewReferenceNo

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

    ENDIF

ENDPROCEDURE
```

[4]

(c) Processing a print job is to be implemented with a `PopJob` procedure.

Complete the pseudocode for this `PopJob` procedure.

```
PROCEDURE PopJob

    IF TopOfStack = ................................................

       THEN

          OUTPUT " .................................................. "

       ELSE

          PROCESS SpoolJob[TopOfStack]

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

    ENDIF

ENDPROCEDURE
```

[3]

(d) Explain why the choice of a stack data structure for this application is a poor choice. Suggest an alternative data structure.

....

....

....

....

....

.... [3]

6 **(a)** A PC operating system uses a file allocation table (FAT) to manage its hard disk secondary storage.

(i) Describe what is meant by a FAT.

....

....

....

.... [2]

(ii) Explain how the contents of the FAT change when a file is deleted from the hard disk.

....

....

....

.... [2]

(b) (i) The processor receives an interrupt. This triggers the following sequence of steps.

1. Save the contents of the Program Counter on the ....
2. Also save ....
3. Load and run the appropriate ....
4. Restore what was saved at step 2
5. Restore the ....
6. Continue execution of the interrupted process

Complete the statements above. [4]

(ii) Interrupts can be allocated priorities.

While execution is occurring at **step 3**, a higher priority interrupt is received. Explain what additional steps must now be added to the sequence in **(b)(i)**. State where in the sequence these additions occur.

....

....

....

....

....

.... [3]

7 Encryption of data is widely used in computing.

(a) One application is the sending of payment data using a debit/credit card for an online purchase.

State **two** other applications where encryption is used.
Describe the reason for encrypting the data for each application.

Application 1 ....

Reason ....

....

Application 2 ....

Reason ....

.... [4]

(b) Explain the terms encryption algorithm and encryption key.

Encryption algorithm ....

....

Encryption key ....

.... [2]

(c) Asymmetric encryption uses both a public key and a private key.

Explain how they work together to encrypt and decrypt a message.

....

....

....

....

....

.... [3]

(d) Authorisation and authentication are processes designed to protect the computer system and data.

Give **one** technique used for each.

Authorisation ....

....

Authentication ....

.... [2]