Cambridge International **A Level**

Cambridge International Examinations
Cambridge International Advanced Level

CANDIDATE NAME

CENTRE NUMBER

CANDIDATE NUMBER

6062941421

COMPUTING **9691/32**

Paper 3 **October/November 2015**

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 an HB pencil for any diagrams, graphs or rough working.
Do not use staples, paper clips, glue or correction fluid.
DO **NOT** WRITE IN ANY BARCODES.

Answer **all** questions.
No marks will be awarded for using brand names of 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 **17** printed pages and **3** blank pages.

BLANK PAGE

1 Members of a youth group take part in different activities. Activities include music tuition courses for various instruments.

A music course has one session each week and lasts for a number of weeks. Each course studies one particular instrument.

Each member has recorded:

- member ID
- other data recorded

Each course has recorded:

- course ID
- instrument
- start date
- number of weeks duration
- course tutor

Over a period of time members may enrol for many courses.

Each course is attended by a number of members.

The data are to be stored in a relational database.

A table description can be expressed as:

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

The primary key is indicated by underlining one or more attributes.

(a) Complete the initial database design by:
- adding two attributes to the `MEMBER` table
- adding appropriate attributes to the `COURSE` table

`MEMBER(MemberID, .....)`

`COURSE(.....)`

[2]

(b) A third table `COURSE-MEMBER` will record which members have enrolled for which courses.

Show the attributes for the table. Show the primary key.

You should not create a course member ID.

`COURSE-MEMBER( .....)` [2]

(c) The database will also store data about tutors and the instrument(s) they can teach.

The following single table, `TUTOR`, is suggested:

Table: `TUTOR`

TutorID	TutorName	TutorTelNo	Instrument
43	Wilber	567541	saxophone
			piano
			violin
92	Said	887654	saxophone
			drums
49	Tasha	897543	guitar

State why the table is not in First Normal Form (1NF).

.....

.....

.....[1]

(d) (i) A revised design is:

```
TUTOR(TutorID, TutorName, PhoneNo)

TUTOR-INSTRUMENT(TutorID, Instrument)
```

Underline the primary key of each table above. [2]

(ii) Name the type of relationship between the `TUTOR-INSTRUMENT` and `TUTOR` tables.

.....[1]

(iii) Write a data manipulation language (DML) query to find the `TutorID` of all saxophone tutors.

Use the keywords `SELECT`, `FROM` and `WHERE`.

.....

.....

.....[3]

(iv) Study this DML command.

```
UPDATE TUTOR-INSTRUMENT
SET Instrument = 'piano'
WHERE TutorID = 49 AND Instrument = 'guitar'
```

Write TRUE or FALSE for each of the six statements:

	TRUE / FALSE
Creates a new record in the `TUTOR-INSTRUMENT` table	
Amends an existing record in the `TUTOR-INSTRUMENT` table	
Assigns the `Instrument` attribute the value 'guitar'	
Assigns the `Instrument` attribute the value 'Piano'	
Make a change to the existing records for all tutors	
Changes one record in the `TUTOR` table	

[3]

(e) (i) A new record needs to be added to the `TUTOR-INSTRUMENT` table. The tutor with TutorID 57 can now teach the flute.

Complete the following DML command.

`INSERT INTO` .....

(.....)

..... (.....)

[2]

(ii) Referential integrity requires that the data is consistent in two related tables.

There is a relationship between the `TUTOR` and `TUTOR-INSTRUMENT` tables.

Using these two tables, explain how data could fail a referential integrity check.

.....

.....

.....

.....[2]

2 **(a)** A program is loaded into main memory starting at memory address 88 in denary.

The diagram shows the memory location contents in Hexadecimal (Hex).

Address	Main memory (hex)
88	7E
89	4A
90	A7
91	FF
. . . .	
256	17
257	8C
258	1F
259	33

(i) Convert the memory address 90 into binary.

90 = ...[1]

(ii) Convert the memory address 90 into hexadecimal.

90 = ...[1]

(iii) State the number of bits used for the contents of each main memory location.

.....[1]

(b) The following table is to show the steps in the fetch stage of the fetch-execute cycle.

Complete the table.

Register transfer notation	Description
.....	The contents of the Program Counter are copied to the Memory Address Register.
PC ← [PC] + 1	....
.....	The Memory Address Register contains an address. Copy the contents of this address to the Memory Data Register.
CIR ← [MDR]	....

[4]

(c) (i) One of the buses found in a typical microprocessor architecture is the control bus.

Describe its purpose.

.....

.....

.....

Give **one** example of a control signal used.

.....[2]

(ii) Name and describe **two** other buses used in a typical microprocessor architecture.

1 .....

.....

.....

.....

2 .....

.....

.....

.....[4]

3 (a) The following table shows some assembly language instructions from the computer's instruction set.

Instruction		Explanation
Op code	**Operand**	
LDD	<address>	Direct addressing. Load the contents of the given address to the Accumulator (ACC)
LDI	<address>	Indirect addressing. The address to be used is at the given address. 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 from <address> + the contents of IX. Copy the contents of this calculated address to ACC

The test program shown in memory locations 700 onwards is to be executed.

Shown are:

- the first four instructions only of this program
- other memory locations which contain data accessed by the program

Complete the trace table below for the first four program instructions.

Instruction	Register	
	ACC	**Index Register (IX)**
LIX 902		
LDD 901		
LDI 901		
LDX 901		

Address	Main memory
700	LIX 902
701	LDD 901
702	LDI 901
703	LDX 901
. . . .	
900	0
901	917
902	2
903	25
904	28
. . . .	
916	96
917	13

[4]

(b) A program written in assembly language needs translation before it can be executed.

A programmer creates, translates and executes an assembly language program. **Five** of the six statements below are to be used to complete a description of this process:

- `Amend PROG.ASM using the text editor`
- `IF errors reported`
- `Produce the PROG.EXE executable file`
- `PROG.ASM is input to the assembler software`
- `Translate PROG.ASM using the interpreter`
- `UNTIL no errors reported`

Complete the pseudocode description below.

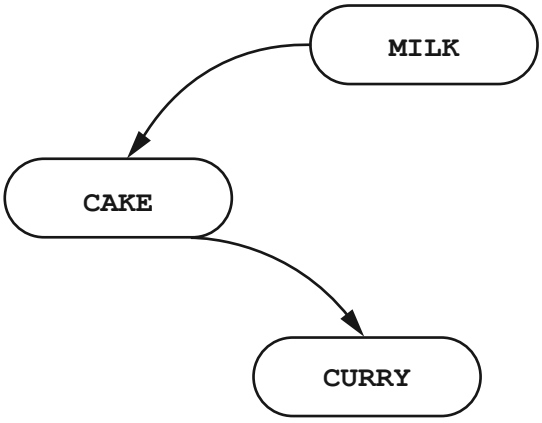
```
Use text editor to create program PROG.ASM

REPEAT

  .....

  .....

   THEN

     .....

  ENDIF

.....

.....

Execute PROG.EXE
```

[5]

4 **(a)** A dataset of food names are to be organised as an ordered binary tree.

The food names below join the binary tree in the order shown:

```
MILK, CAKE, CURRY
```



Four more food names join the binary tree in the order shown:

```
SCONES, RICE, BREAD, COUSCOUS
```

(i) Draw these values on the binary tree above. [4]

(ii) On the binary tree above, write on the tree as follows:

- label the root
- draw a line around the right subtree [2]

(iii) State the number of leaf nodes for the completed tree in **part (a)(i)**.

.....[1]

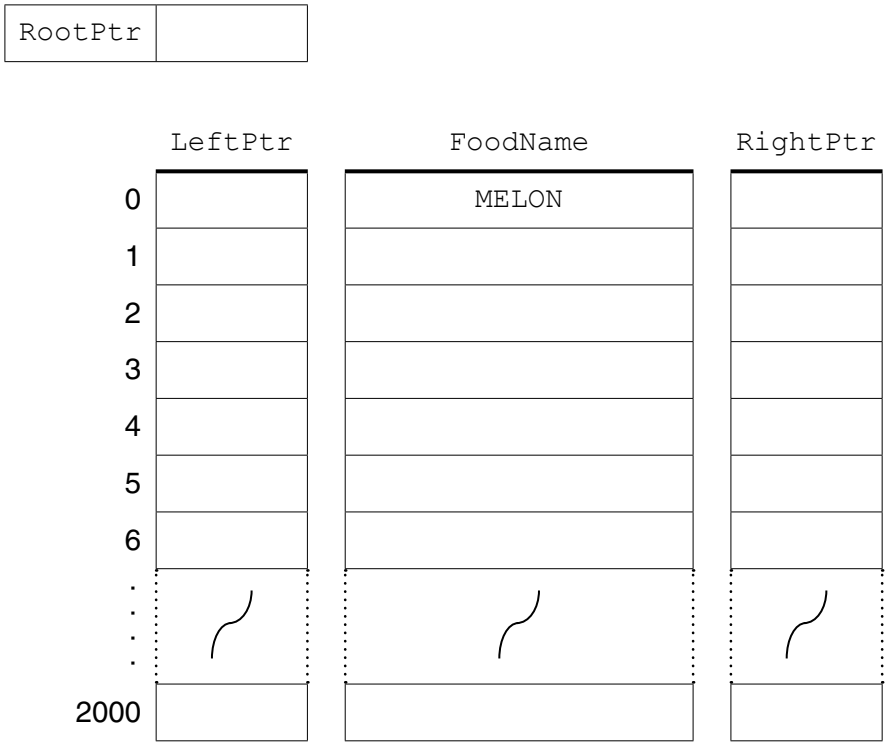
(b) The binary tree is implemented in a high-level language using a number of data structures and one variable:

Variable	Data type	Description
RootPtr	INTEGER	The array subscript of the root of the tree
FoodName	ARRAY[0 : 2000] OF STRING	Array of food names
RightPtr	ARRAY[0 : 2000] OF INTEGER	Array of right pointer values
LeftPtr	ARRAY[0 : 2000] OF INTEGER	Array of left pointer values

A new dataset of food names is used as test data:

```
MELON, BEETROOT, TURNIP, APPLE, PARSNIP, SWEDE, QUINCE
```

Complete the diagram below showing the contents of the arrays and the root pointer variable.



[4]

(c) An algorithm is designed in pseudocode to search the binary tree for a particular food name.

The algorithm uses the additional variables below:

Variable	Data type	Description
SearchFood	STRING	Food name to search for
Current	INTEGER	The array subscript for the item currently considered
IsFound	BOOLEAN	Flags to TRUE when SearchFood is found

Complete the algorithm below:

```
//binary tree search
INPUT SearchFood
IsFound ← .....
Current ← .....
REPEAT
  IF Food[Current] = .....
    THEN
      //found
      OUTPUT "Found"
      .....
    ELSE
      IF SearchFood < Food[Current]
        THEN
          // move left
          Current ← LeftPtr[Current]
        ELSE
          .....
      ENDIF
  ENDIF
UNTIL IsFound = TRUE OR .....

IF IsFound = FALSE
  THEN
    OUTPUT SearchFood "Not Found"
ENDIF
```

[6]

(d) (i) The dataset used in **part (b)** was:
`MELON, BEETROOT, TURNIP, APPLE, PARSNIP, SWEDE, QUINCE`

Draw the binary tree.

[2]

A well-balanced tree has approximately the same number of nodes in the left and right subtrees.

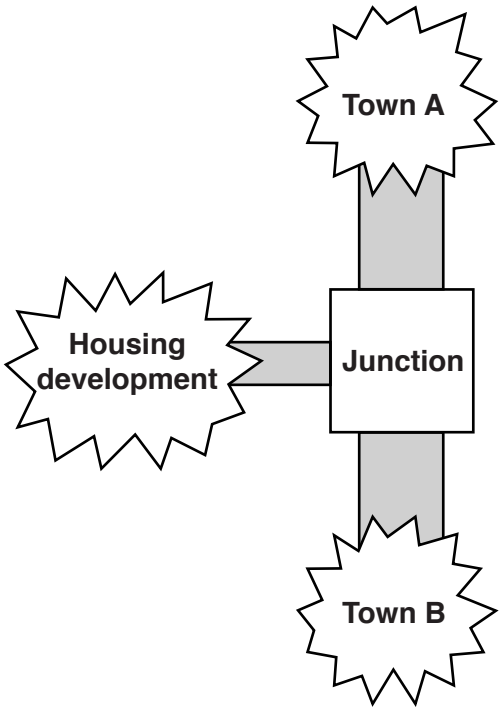
(ii) State whether or not this is a well-balanced tree.

.....

.....[1]

5 A busy main road connects Town A and Town B. A large new housing development is being built near the main road. A new road from the housing development will connect to the main road. Traffic control signals are planned for the junction.

It is proposed to simulate the traffic flows to assist the planning of the traffic control system.



(a) (i) Explain what is meant by a simulation.

.....

.....

.....

.....[2]

(ii) Describe why a computer is suited to carry out a simulation.

.....

.....[1]

(b) Data will need to be collected before a simulation can be carried out.

Name **two** datasets which will be collected.

1 .....

.....

2 .....

.....[2]

(c) The number of traffic lanes approaching the junction could be an input variable for the simulation.

(i) Name **two** other input variables which will be used.

1 .....

.....

2 .....

.....
[2]

(ii) State an output variable which will provide data for the planning process for the traffic control system.

.....

.....[1]

6 **(a)** A user-defined function `CountBig` is defined, using pseudocode, as follows:

```
CountBig(ThisArray : ARRAY OF INTEGER, ThisValue : INTEGER,
                                          ThisFlag : CHAR) : INTEGER
```

The function checks each element in the array `ThisArray`. It counts the number of elements which are greater than (or greater than or equal to) `ThisValue`.

The function returns the count value.

`ThisFlag` has two permitted values:
- Y – indicates values equal to `ThisValue` are to be included in the count
- N – indicates values equal to `ThisValue` will not be included in the count

An error is generated if the function is not correctly formed.

The function is used with the three arrays shown below:

	Subscript/Index									
Identifier	1	2	3	4	5	6	7	8	9	10
`ScoreA`	16	19	23	28	28	36				
`ScoreB`	71	67	51	82	77	12	80	16		
`ScoreC`	19	19	19	19	19	19	19	19	19	

State the value returned by the following function calls:

(i) `CountBig(ScoreA, 23, 'Y')`

.....[1]

(ii) `CountBig(ScoreB, 71, 'N')`

.....[1]

(iii) `CountBig('Y', ScoreC, 18)`

.....[1]

(iv) `CountBig(ScoreC, 19, 'N')`

.....[1]

(v) `CountBig("ScoreC", 19, 'N')`

.....[1]

(b) The function header used in **part (a)** was:

```
FUNCTION CountBig(ThisArray : ARRAY OF INTEGER,
                  ThisValue : INTEGER,
                  ThisFlag : CHAR) : INTEGER
```

(i) Use the following letters to label your function header above.

A Function parameter(s)

B Where the return data type is stated [2]

(ii) Write a single statement which calls the function `CountBig` to count the number of times value 83 or larger is found in the array `Rejects`.

The function returns the value to the variable `RejectCount` in the calling statement.

.....[2]

(c) A programmer writes pseudocode for a new function `StringFound`. The function is to report whether or not a string value `ThisValue` is found in the string array `ThisArray`. The function returns a Boolean value.

Write the function header for function `StringFound`.

.....

.....

.....[4]

7 A company is considering a new local area network (LAN) for the order processing operation of its business. It intends to use the existing four computers to form a wired bus topology LAN.

(a) Define what is meant by a Local Area Network.

.....

.....[2]

(b) The following are extracts from conversations between the Order Processing Manager and the IT Manager.

(i) *"The computers may need to be fitted with additional hardware"*

State what hardware this is.

.....[1]

(ii) *"It is important that only the order processing staff have access to the LAN and access to the Internet is restricted".*

State what authentication technique will be used for this.

.....

Name the hardware needed to restrict access to the Internet.

.....[2]

(c) *"We will need to purchase a licence for a network operating system".*

Describe **three** tasks performed by the network operating system software.

1 .....

.....

2 .....

.....

3 .....

.....

[3]

(d) Name the hardware device which will be used on the LAN for the storage of all user data and programs.

.....[1]

BLANK PAGE

BLANK PAGE