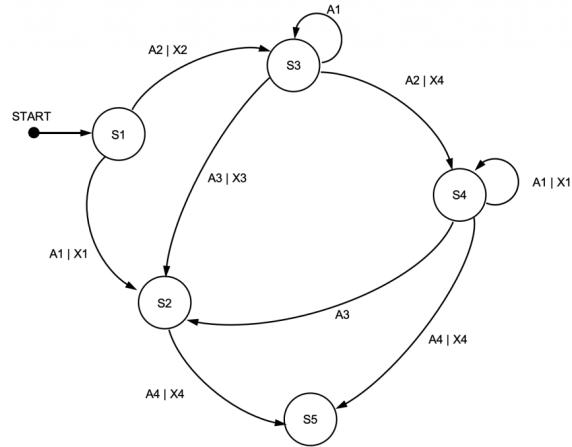


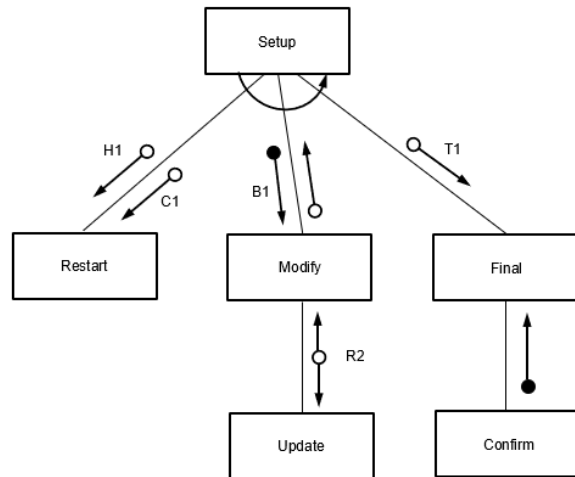
7(a)



One mark for each:

- 1 Label A1 | X1 added on line from S1 to state labelled S2
- 2 States S3, S4 and S5 labelled
- 3 Line from S2 to S5 with event label A4|X4
- 4 Lines from S3 with correct event labels including 'loop' label
- 5 Lines from S4 with correct event labels including 'loop'

7(b)



One mark per point:

- 1 All boxes correctly labelled
- 2 Iteration arrow
- 3 Parameter to Restart
- 4 Return value from Modify **and** parameter to Final
- 5 BYREF parameter to Update

7(a)

Example explanation:

The arrow means that Convert **repeatedly** calls Analyse then Update **and** then Sort (in that order)10

MP1 Reference to 'repetition'...**MP2** Naming **all four modules** correctly **and** the sequence is stated or implied

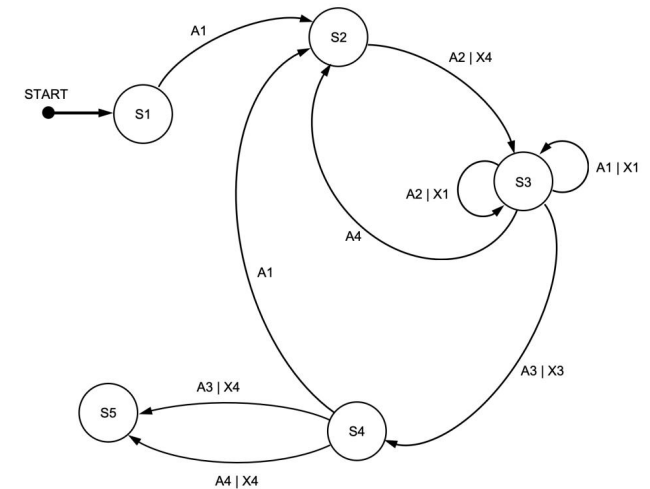
7(b)

MP1 PROCEDURE Analyse (Count : INTEGER)FUNCTION Update (Key : STRING) RETURNS INTEGER**MP2****MP3****MP4** PROCEDURE Sort (BYREF P2 : BOOLEAN)

Mark as follows:

One mark per underlined part

7(a)

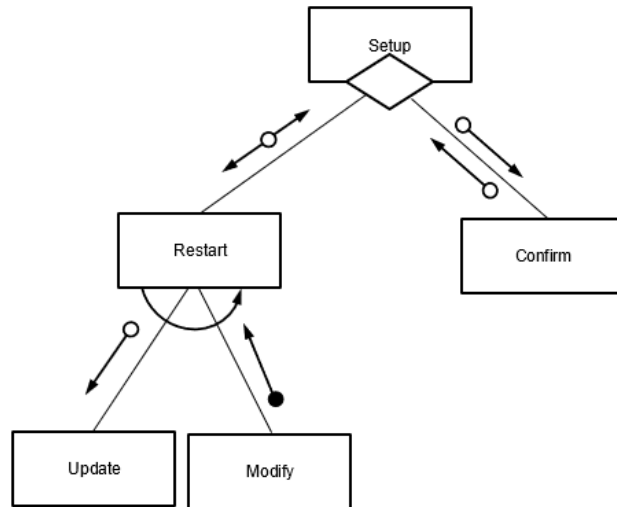


One mark for each:

- 1 S2 to S5 labelled
- 2 Label A1 added to event from S1 to S2
- 3 Event lines from S2 to S3 **and** S3 to S2 correctly labelled
- 4 Events lines from S3 to S3 **and** event line from S3 to S4 all labelled correctly
- 5 Event lines from S4 to S5 correctly **and** event line from S4 to S2 labelled correctly

Max 4 for any additional events/lines

7(b)



One mark for each:

- 1 All **boxes** correctly labelled and connected in correct hierarchy
- 2 BYREF parameter to Restart, parameter to Confirm and return
- 3 Diamond and loop symbol
- 4 Parameter to Update **and** return from Modify

6(a)

MP1
MP2
Max 1

A function returns a (single) **value** and a procedure does not
A procedure can pass parameters by ByRef

6(b)

Sub_Part1A:

```
PROCEDURE Sub_Part1A(R: INTEGER)
```

1 Mark

Sub_Part2A:

```
FUNCTION Sub_Part2A(T : REAL) RETURNS BOOLEAN
```

2 Marks

Sub_Part3A:

```
PROCEDURE Sub_Part3A(V: INTEGER, W: BOOLEAN, BYREF U :  
STRING)
```

2 Marks

6(c)

Symbol	Explanation
	The module Main calls either Sub_A or Sub_B (which one is called is determined at run time)
	The module Sub_A will repeatedly call Sub_Part1A followed by Sub_Part2A then by Sub_Part3A

MP1 Shape1: reference to **selection****MP2** Shape2: reference to **repetition // iteration****MP3** **Descption (x2)** including correct use of **all module names** in **both** parts

7(a)	One mark per point: 1 All module identified in correct hierarchy 2 Parameters to both Sync and Activate and return values 3 Parameters to both Reset and Enable 4 Parameters to Initialise 5 Decision diamond
7(b)	Explanation: Means that Sync repeatedly calls Reset and (then) Enable One mark for each point: MP1: reference to iteration MP2: naming all three modules correctly including correct sequence of calls

7(a)	• Customer ID – to reference the other customer details • Email address – to send the email • Name – to personalise the email • Date of last visit – to select which customers should receive an email • Unique voucher code (or method of code generation) – to include in the email Mark as follows: One mark per item and justification Max 3 marks
7(b)	Abstraction
7(c)	MP1 Data structures // data dictionary // identifier table(s) // validation rules MP2 Data-flow diagram // state-transition diagram MP3 User interface // Format for the email MP4 Testing method / Test plan / Test data / Trace tables MP5 Choice of email protocol to be used // Programming language to be used // Development environment MP6 Use of library routines // program to send the email Max 3 marks
7(d)	MP1 Three boxes correctly labelled and correct hierarchy MP2 Parameter and return values MP3 Iteration arrow

7(a)	One mark per answer: MP1 Analysis MP2 Named document, examples include: • A <u>requirement specification</u> • Definition of System Objectives • List of problems with existing system • Survey results • Feasibility study • Interview notes Max 2 marks
------	---

7(b)	One mark for name and use Mark as follows: Examples include: Sub-module: <code>ScanCard()</code> / <code>GetID()</code> Use: Read the barcode from the loyalty card / Get the customer ID from the barcode Sub-module: <code>GetPoints()</code> Use: Get the number of points the customer has Sub-module: <code>ExchangePoints()</code> / <code>UpdateCard()</code> Use: Reduce the number of loyalty points Sub-module: <code>GetDiscount()</code> / <code>CalculateDiscount()</code> Use: Calculate the discount Sub-module: <code>CalculatePoints()</code> Use: Calculate points (following a purchase) Sub-module: <code>UpdatePoints()</code> Use: Update total points a customer has (following a purchase) Sub-module: <code>ShowDiscount()</code> Use: Display the discount Max 4 marks
------	--