PAPER 2 DIAGRAMS

COMPUTER SCIENCE

Topic: Paper 2 Diagrams

INSTRUCTIONS

- Carry out every instruction in each task.
- Answer **all** questions.
- Use a black or dark blue pen.
- You may use an HB pencil for any diagram, graphs or rough working.
- **Calculator Not Allowed**.
- Show your workings if relevant.

INFORMATION

- The number of marks for each question or part question is shown in brackets [ ].

bishaltwr@gmail.com

2 An algorithm is described as follows:

1. Input an integer value.
2. Jump to step 6 if the value is less than zero.
3. Call the function `IsPrime()` using the integer value as a parameter.
4. Keep a count of the number of times function `IsPrime()` returns `TRUE`.
5. Repeat from step 1.
6. Output the value of the count with a suitable message.

Draw a program flowchart to represent the algorithm.

[4]

3 **(a)** The module headers for five modules in a program are defined in pseudocode as follows:

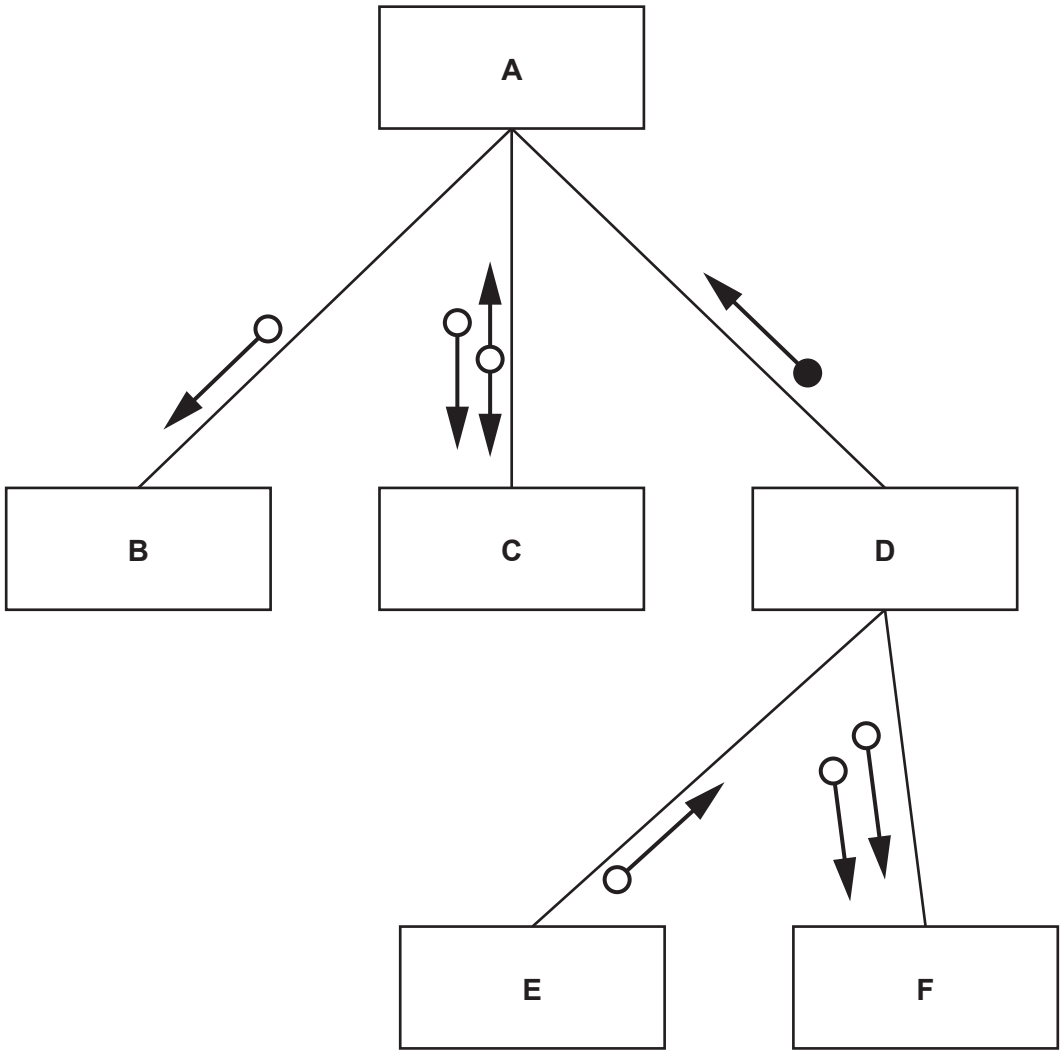
Pseudocode module header
`FUNCTION Mod_V(S2 : INTEGER) RETURNS BOOLEAN`
`PROCEDURE Mod_W(P4 : INTEGER)`
`PROCEDURE Mod_X(T4 : INTEGER, BYREF P3 : REAL)`
`PROCEDURE Mod_Y(W3 : REAL, Z8 : INTEGER)`
`FUNCTION Mod_Z(F3 : REAL) RETURNS INTEGER`

An additional module `Head()` repeatedly calls three of the modules in sequence.

A structure chart has been partially completed.

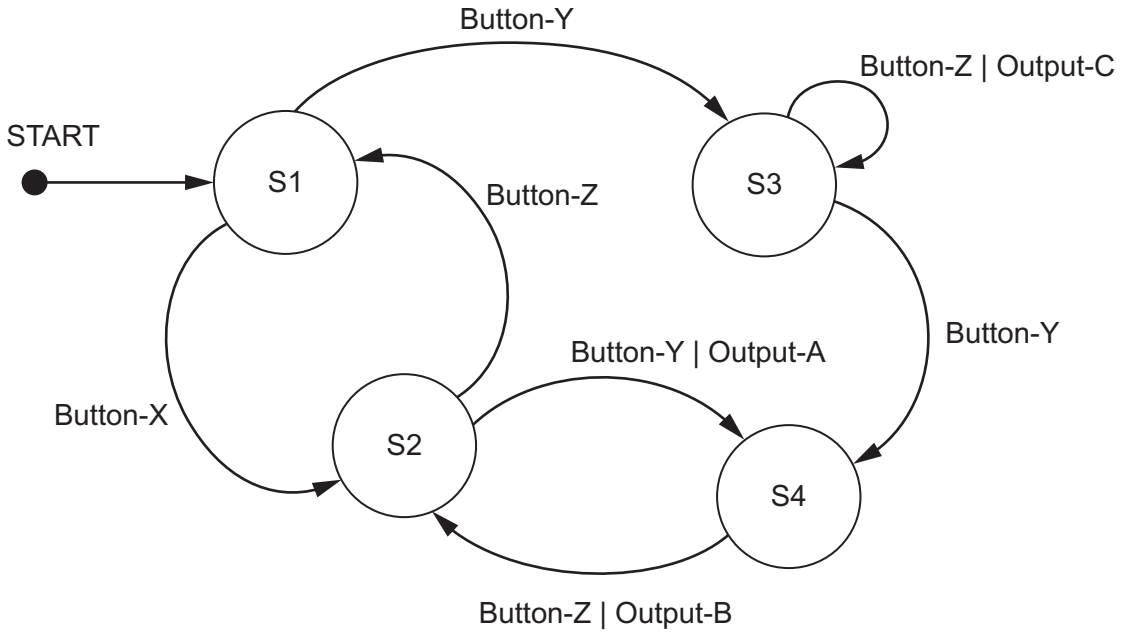
(i) Complete the structure chart to include the information given about the six modules.

Do **not** label the parameters and do **not** write the module names.



[3]

2 Examine the following state-transition diagram.



(a) Complete the table with reference to the diagram.

	Answer
The number of different inputs	
The number of different outputs	
The single input value that could result in S4	

[3]

(b) The initial state is S1.

Complete the table to show the inputs, outputs and next states.

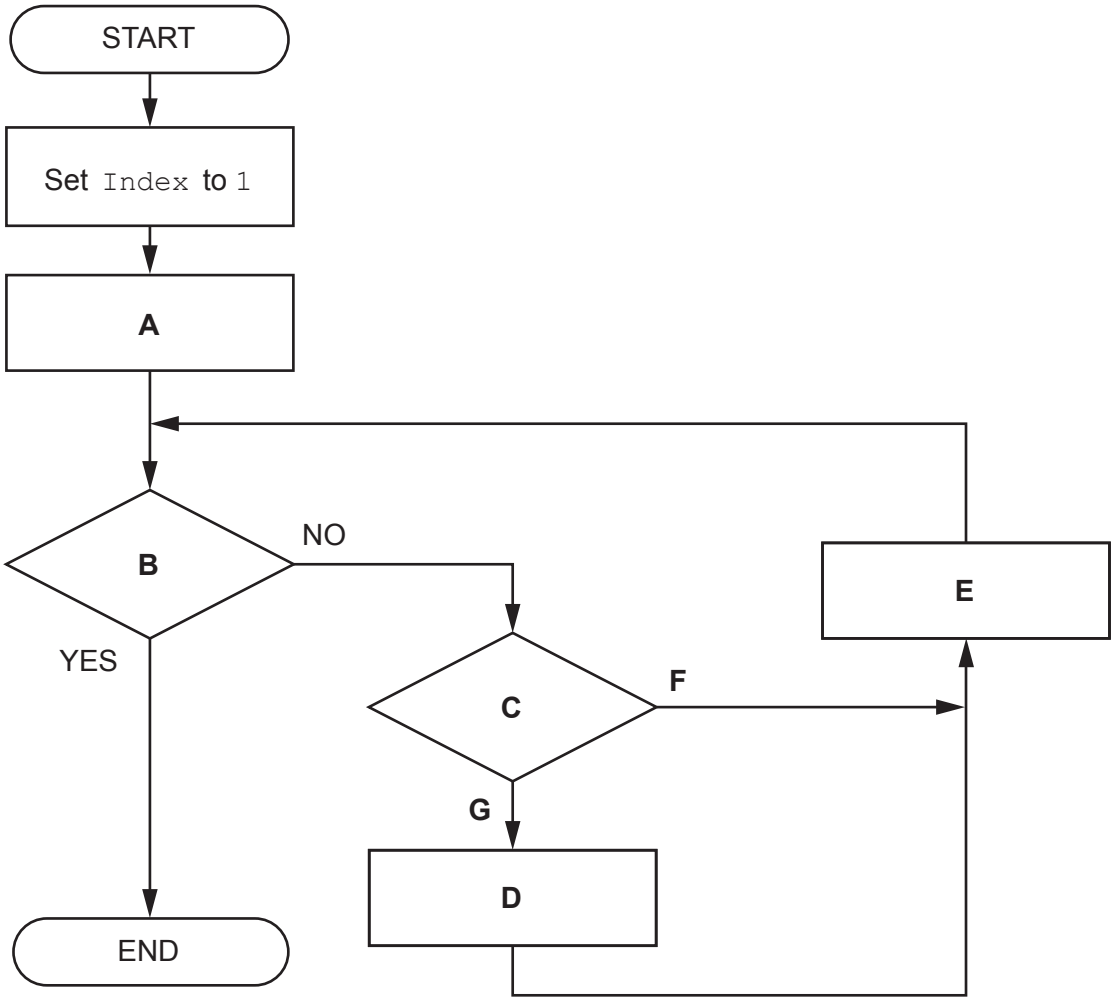
Input	**Output**	**Next state**
Button-Y		
	none	
Button-Z		S2
	none	

[4]

(b) Strings may consist of several words separated by spaces.

For example, the string "never odd or even" becomes a palindrome if the spaces are removed.

The program flowchart represents an algorithm to produce a string `OutString` by removing all spaces from a string `InString`.



Complete the table by writing the text that should replace each of the labels **B**, **C**, **D**, **F** and **G**.

Note: the text may be written as a pseudocode statement.

Label	Text
A	Set `OutString` to `""`
B	
C	
D	
E	Set `Index` to `Index + 1`
F	
G	

[4]

4 A program controls the heating system in a sports hall.

Part of the program involves reading a value from a sensor. The sensor produces a numeric value that represents the temperature. The value is an integer, which should be in the range 0 to 40 inclusive.

A program function has been written to validate the values from the sensor.

(a) A test plan is needed to test the function.

Complete the table. The first line has been completed for you.

You can assume that the sensor will generate only integer data values.

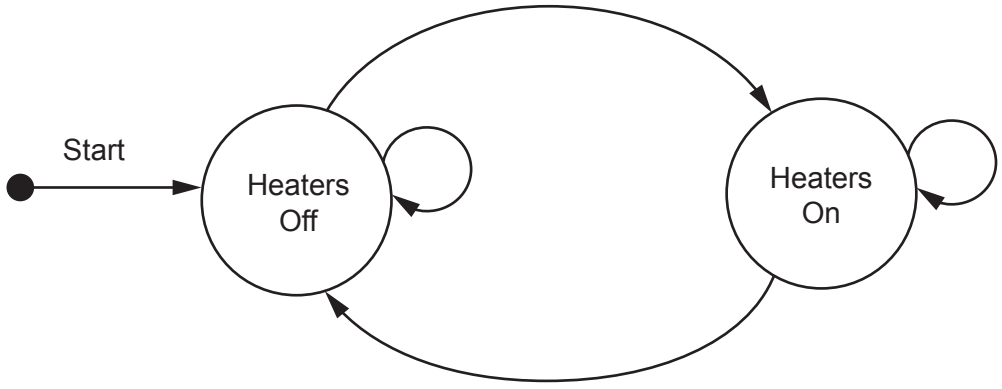
Test	Test data value	Explanation	Expected outcome
1	23	Normal data	Data is accepted
2			
3			
4			
5			

[4]

(b) A program module controls the heaters. This module operates as follows:

- If the temperature is below 10, switch the heaters on.
- If the temperature is above 20, switch the heaters off.

Complete the following state-transition diagram for the heating system:



[3]

2 **(a)** An algorithm will:

1. input an integer value
2. jump to step 6 if the value is zero
3. sum and count the positive values
4. sum and count the negative values
5. repeat from step 1
6. output the two sum values and the two count values.

Draw a program flowchart on the following page to represent the algorithm.

Note that variable declarations are not required in program flowcharts.

[5]

4 A program controls the heating system in a sports hall.

Part of the program involves reading a value from a sensor. The sensor produces a numeric value that represents the temperature. The value is an integer, which should be in the range 0 to 40 inclusive.

A program function has been written to validate the values from the sensor.

(a) A test plan is needed to test the function.

Complete the table. The first line has been completed for you.

You can assume that the sensor will generate only integer data values.

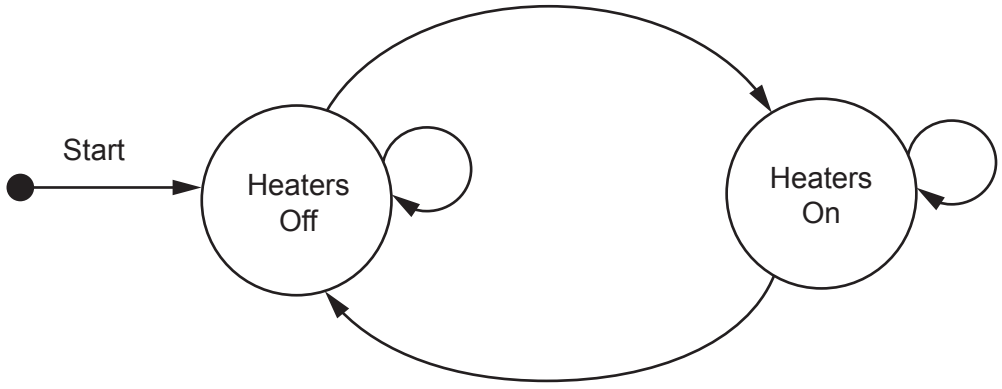
Test	Test data value	Explanation	Expected outcome
1	23	Normal data	Data is accepted
2			
3			
4			
5			

[4]

(b) A program module controls the heaters. This module operates as follows:

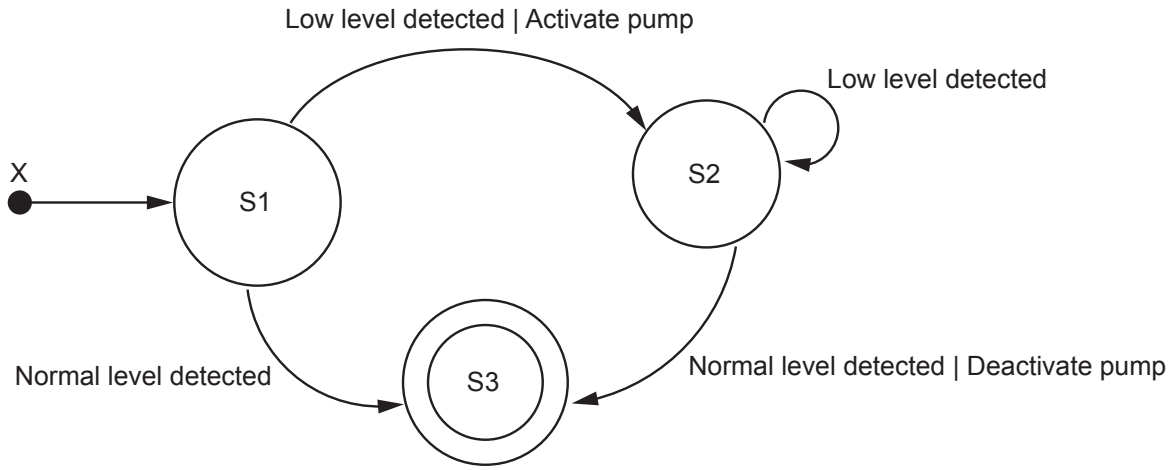
- If the temperature is below 10, switch the heaters on.
- If the temperature is above 20, switch the heaters off.

Complete the following state-transition diagram for the heating system:



[3]

2 **(a)** Examine the following state-transition diagram.



(i) Complete the table with reference to the diagram.

	Answer
The number of transitions that result in a different state	
The number of transitions with associated outputs	
The label that should replace 'X'	
The final or halting state	

[4]

(ii) The current state is S1. The following inputs occur.

1. Low level detected
2. Low level detected
3. Low level detected
4. Low level detected

Give the number of outputs and the current state.

Number of outputs .....

Current state .....

[2]

2 (a) Four program modules form part of a program for a library.

A description of the relationship between the modules is summarised as follows:

Module name	Description
`UpdateLoan()`	• Calls either `LoanExtend()` or `LoanReturn()`
`LoanExtend()`	• Called with parameters `LoanID` and `BookID` • Calls `CheckReserve()` to see whether the book has been reserved for another library user • Returns `TRUE` if the loan has been extended, otherwise returns `FALSE`
`CheckReserve()`	• Called with `BookID` • Returns `TRUE` if the book has been reserved, otherwise returns `FALSE`
`LoanReturn()`	• Called with parameters `LoanID` and `BookID` • Returns a `REAL` (which is the value of the fine to be paid in the case of an overdue loan)

Draw a structure chart to show the relationship between the four modules and the parameters passed between them.

[5]

(c) A program will:

- input 50 unique integer values
- output the largest value
- output the average of the values **excluding** the largest value.

Draw a program flowchart to represent the algorithm.

Variable declarations are **not** required.

It is not necessary to check that each input value is unique.

[6]

3 Four program modules are defined as follows:

Pseudocode module header
`PROCEDURE Sub1_A(XT : INTEGER, PB : STRING)`
`FUNCTION Sub1_B(RA : INTEGER) RETURNS BOOLEAN`
`PROCEDURE Sub1_C(SB : INTEGER, BYREF SA : STRING)`
`PROCEDURE Section_1()`

(a) A structure chart will be produced as part of the development process.

Describe the purpose of a structure chart.

.....

.....

.....

..... [2]

(b) Module `Section_1()` calls one of the other three modules. The module called will be selected when the program runs.

Draw the structure chart.

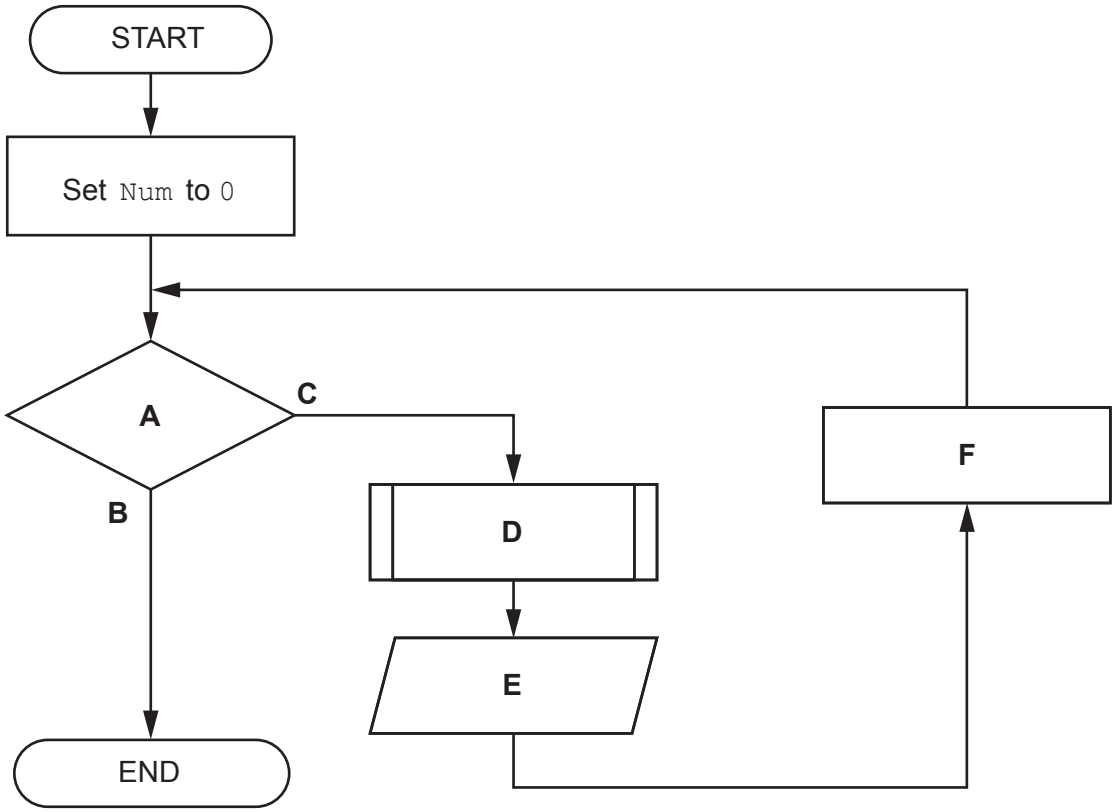
[5]

(b) A procedure `FirstTen()` will output the factorial of the numbers from 0 to 9. The procedure will use the function from **part (a)**.

The required output is:

```
Factorial of 0 is 1
Factorial of 1 is 1
Factorial of 2 is 2
   ⌒
Factorial of 9 is 362880
```

The program flowchart represents an algorithm for `FirstTen()`.



Complete the table by writing the text that should replace each label **A** to **F**.

Label	Text
A	
B	
C	
D	
E	
F	

[4]

(ii) Explain why it may be better to store the names of the students in a file rather than in an array.

.....

.....

..... [1]

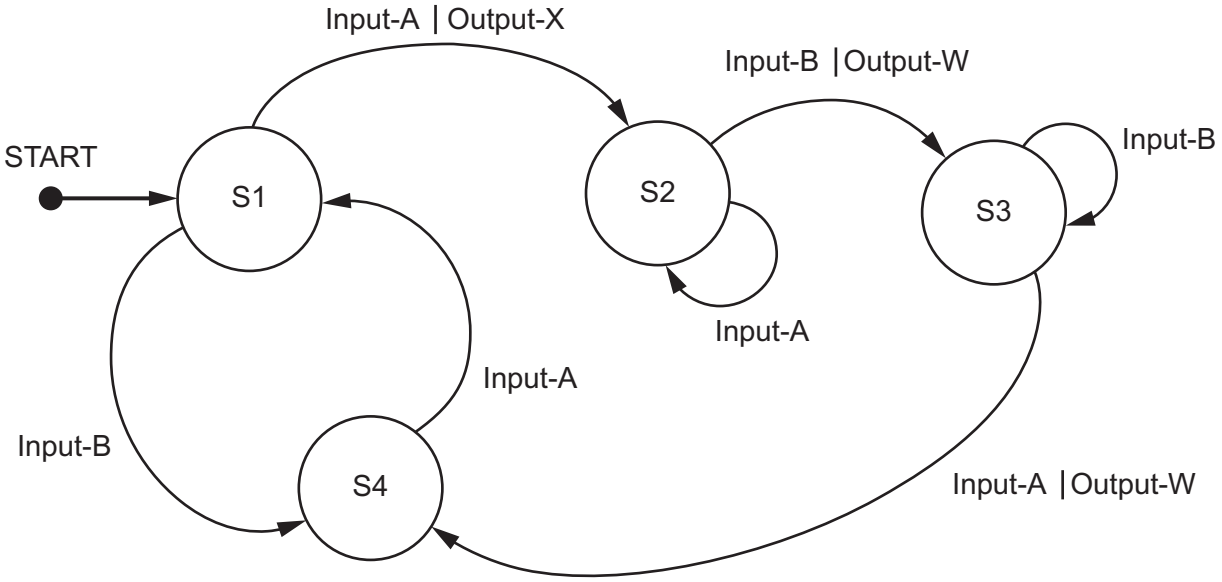
(iii) Explain why `WRITE` mode cannot be used in the answer to **part 2(a)(i)**.

.....

.....

..... [1]

(b) Examine the following state-transition diagram.



Complete the table to show the inputs, outputs and next states.

Input	Output	Next state
		S1
Input-A		
		S2
	Output-W	
	Output-W	

[4]

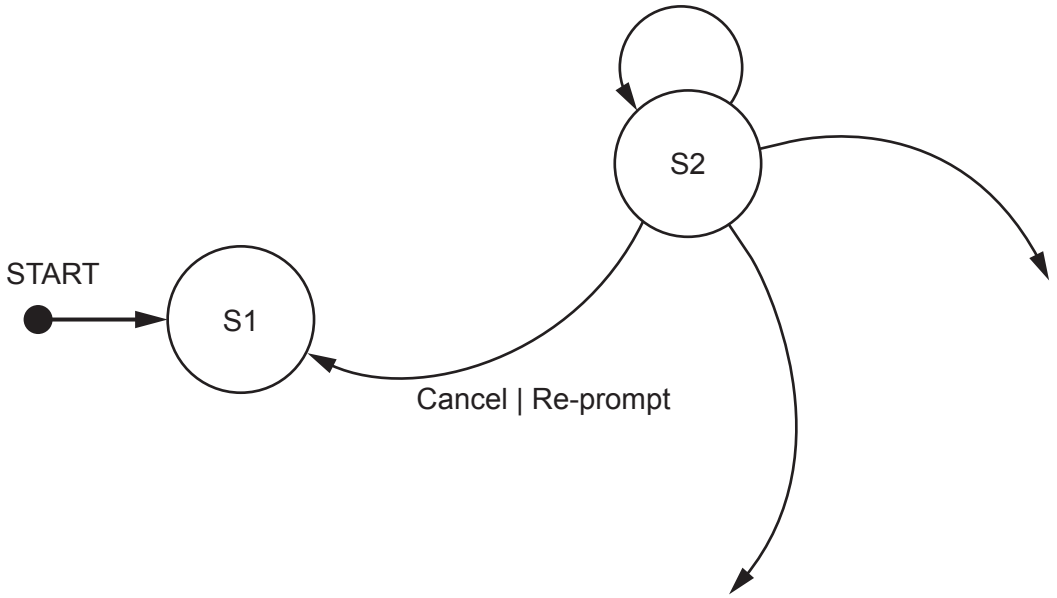
(b) The program includes a module to validate a Personal Identification Number (PIN). This is used when customers pay for goods using a bank card.

A state-transition diagram has been produced for this module.

The table show the inputs, outputs and states for this part of the program:

Current state	Input	Output	Next state
S1	Input PIN		S2
S2	Re-input PIN	Display error	S2
S2	Cancel	Re-prompt	S1
S2	Valid PIN	Enable payment	S4
S2	Too many tries	Block Account	S3

Complete the state-transition diagram to represent the information given in the table.



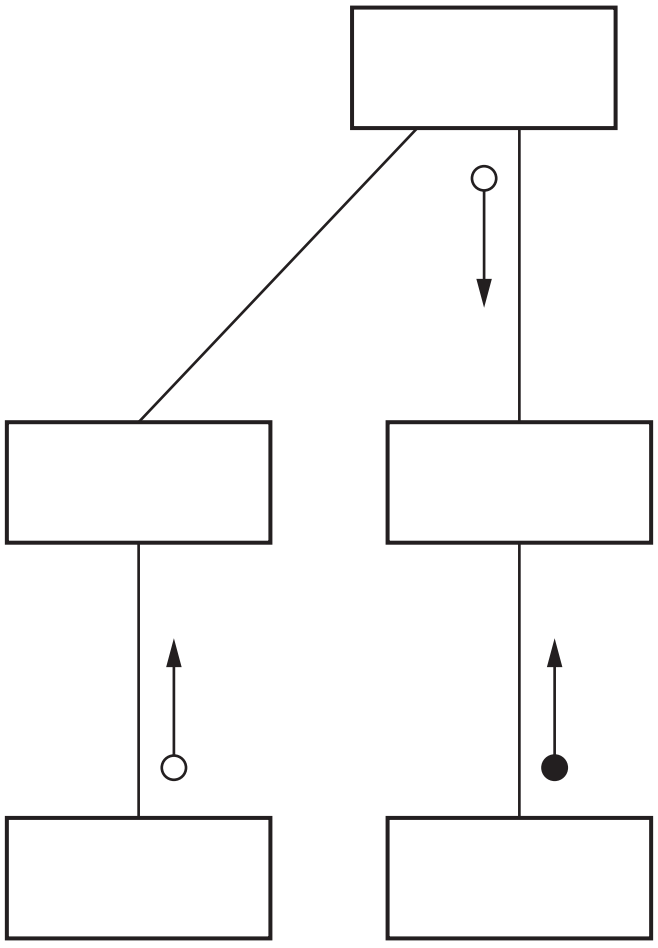
[4]

(b) Part of the library program contains program modules with headers as follows:

Pseudocode module header
`PROCEDURE Module-X()`
`PROCEDURE Module-Y(BYREF RA : INTEGER, SA : REAL)`
`PROCEDURE Overlay()`
`FUNCTION Replace(RA : INTEGER, RB : BOOLEAN) RETURNS BOOLEAN`
`FUNCTION Reset(TA : STRING) RETURNS INTEGER`

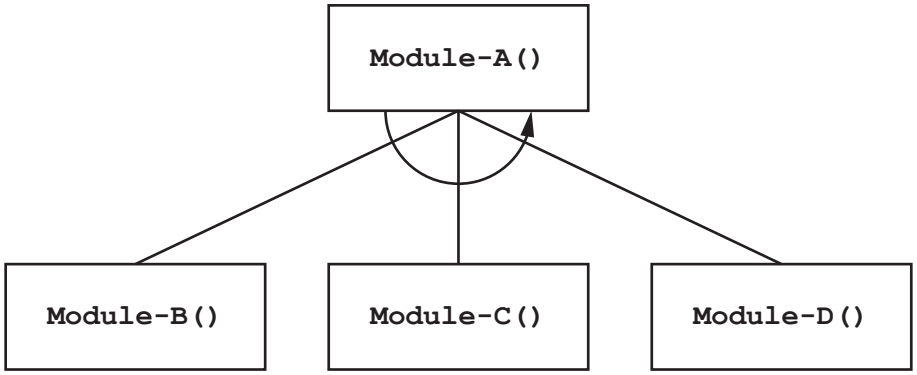
`Module-X()` and `Module-Y()` are both called from module `Overlay()`.

Complete the structure chart.



[3]

(b) The program designer produces a structure chart for the new module. Part of the structure chart is shown:



(i) Explain the relationship between the four modules shown.

.....

.....

.....

..... [2]

(ii) Two new modules are added: `Module-X()` and `Module-Y()`.

- `Module-X()` has no parameters.
- `Module-Y()` will take a string and a real number as parameters and return a Boolean value.
- `Module-D()` will call either `Module-X()` or `Module-Y()`.

Draw **only** the part of the structure chart that represents the relationship between `Module-X()`, `Module-Y()` and `Module-D()`.

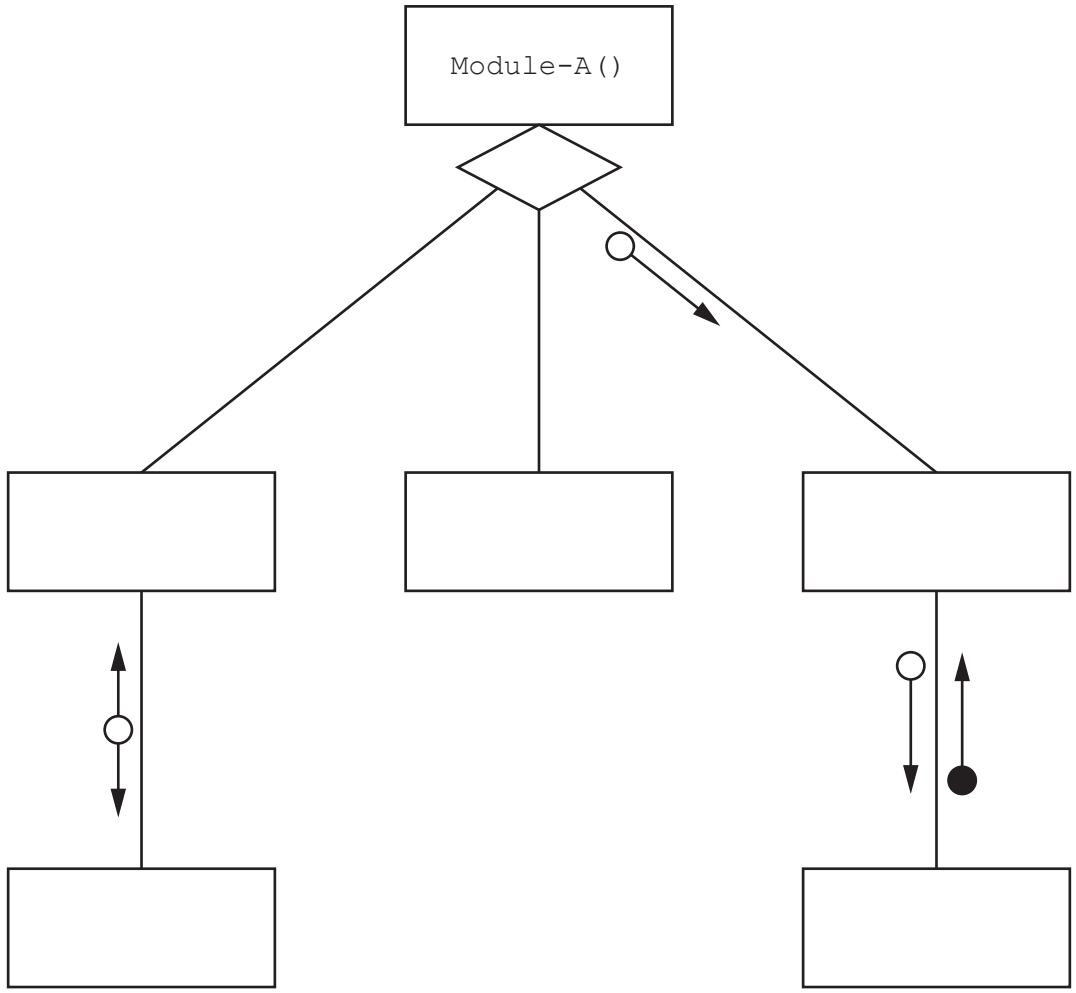
[3]

7 A program contains six modules:

Pseudocode module header
`PROCEDURE Module-A()`
`PROCEDURE Module-X(T1 : INTEGER, S2 : REAL)`
`PROCEDURE Reset(BYREF Code : INTEGER)`
`FUNCTION Restore(OldCode : INTEGER) RETURNS BOOLEAN`
`FUNCTION Module-Y(RA : INTEGER, RB : BOOLEAN) RETURNS BOOLEAN`
`FUNCTION Module-Z(SA : INTEGER) RETURNS INTEGER`

`Module-X()` calls `Reset()`
`Module-Y()` calls `Restore()`

(a) Complete the structure chart for these modules.



[4]

(b) Explain the meaning of the diamond symbol as used in the diagram in part **(a)**.

.....

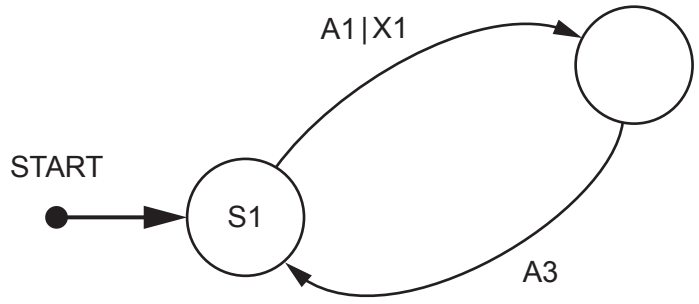
..... [2]

7 An algorithm is represented by a state-transition diagram.

The table shows the inputs, outputs and states for the algorithm:

Current state	Input	Output	Next state
S1	A1	X1	S2
S2	A3	none	S1
S2	A2	X4	S5
S5	A1	X1	S5
S5	A4	X2	S2
S5	A3	none	S3
S1	A9	X9	S3
S3	A9	X9	S4

Complete the state-transition diagram to represent the information given in the table.



[5]

3 A record structure is declared to hold data relating to components being produced in a factory:

```
TYPE Component
   DECLARE Item_ID : STRING
   DECLARE Reject : BOOLEAN
   DECLARE Weight : REAL
ENDTYPE
```

The factory normally produces a batch (or set) of 1000 components at a time. A global array is declared to store 1000 records for a batch:

```
DECLARE Batch : ARRAY [1:1000] OF Component
```

Two global variables contain the minimum and maximum acceptable weight for each component. The values represent an inclusive range and are declared as:

```
DECLARE Min, Max : REAL
```

(a) (i) A program uses a variable `ThisIndex` as the array index to access a record.

Write a pseudocode clause to check whether or **not** the weight of an individual component is within the acceptable range.

.....

.....

..... [3]

(ii) When batches of less than 1000 components are processed, it is necessary to indicate that certain elements in the array are unused.

Suggest how an unused array element could be indicated.

.....

..... [1]

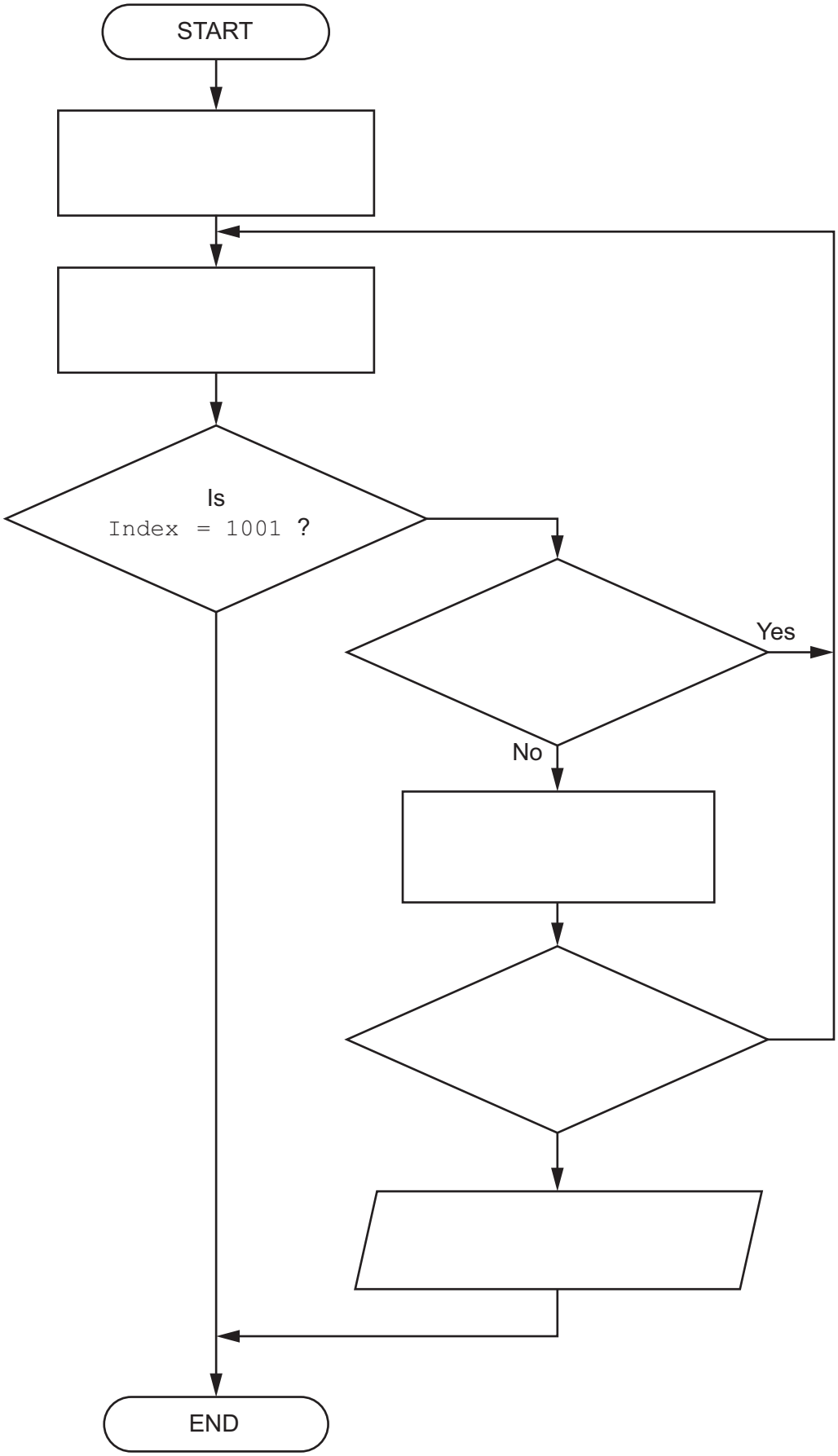
(b) A module `InRange()` will:

- be called with an integer parameter representing an index value of a record in the `Batch` array
- check if the weight of the indexed component is within the acceptable range
- return `TRUE` if the weight is in the range and `FALSE` if it is **not**.

A module `BatchCheck()` will:

- iterate through a batch of 1000 component records
- call module `InRange()` to check each individual component record
- keep a count of the number of components that fail
- output a suitable warning message and immediately stop if the number of failed components exceeds 5.

Complete the program flowchart to represent the algorithm for module `BatchCheck()`.



[5]

7 Seven program modules form part of a program. A description of the relationship between the modules is summarised below. Any return values are stated in the description.

Module name	Description
Mod-A	calls Mod-B followed by Mod-C
Mod-B	• called with parameters Par1 and Par2 • calls either Mod-D or Mod-E, determined when the program runs • returns a Boolean value
Mod-C	• called with parameters Par1 and Par3 • Par3 is passed by reference • repeatedly calls Mod-F followed by Mod-G
Mod-D	called with parameter Par2
Mod-E	• called with parameter Par3 • returns an integer value
Mod-F	called with parameter Par3
Mod-G	• called with parameter Par3 • Par3 is passed by reference

Parameters in the table are as follows:
- Par1 and Par3 are of type string.
- Par2 is of type integer.

(a) (i) Identify the modules that would be implemented as functions.

..... [1]

(ii) Modules Mod-F and Mod-G are both called with Par3 as a parameter.
In the case of Mod-F, the parameter is passed by value.
In the case of Mod-G, the parameter is passed by reference.

Explain the effect of the **two** different ways of passing the parameter Par3.

.....

.....

.....

..... [2]

(b) Draw a structure chart to show the relationship between the seven modules and the parameters passed between them.

[6]

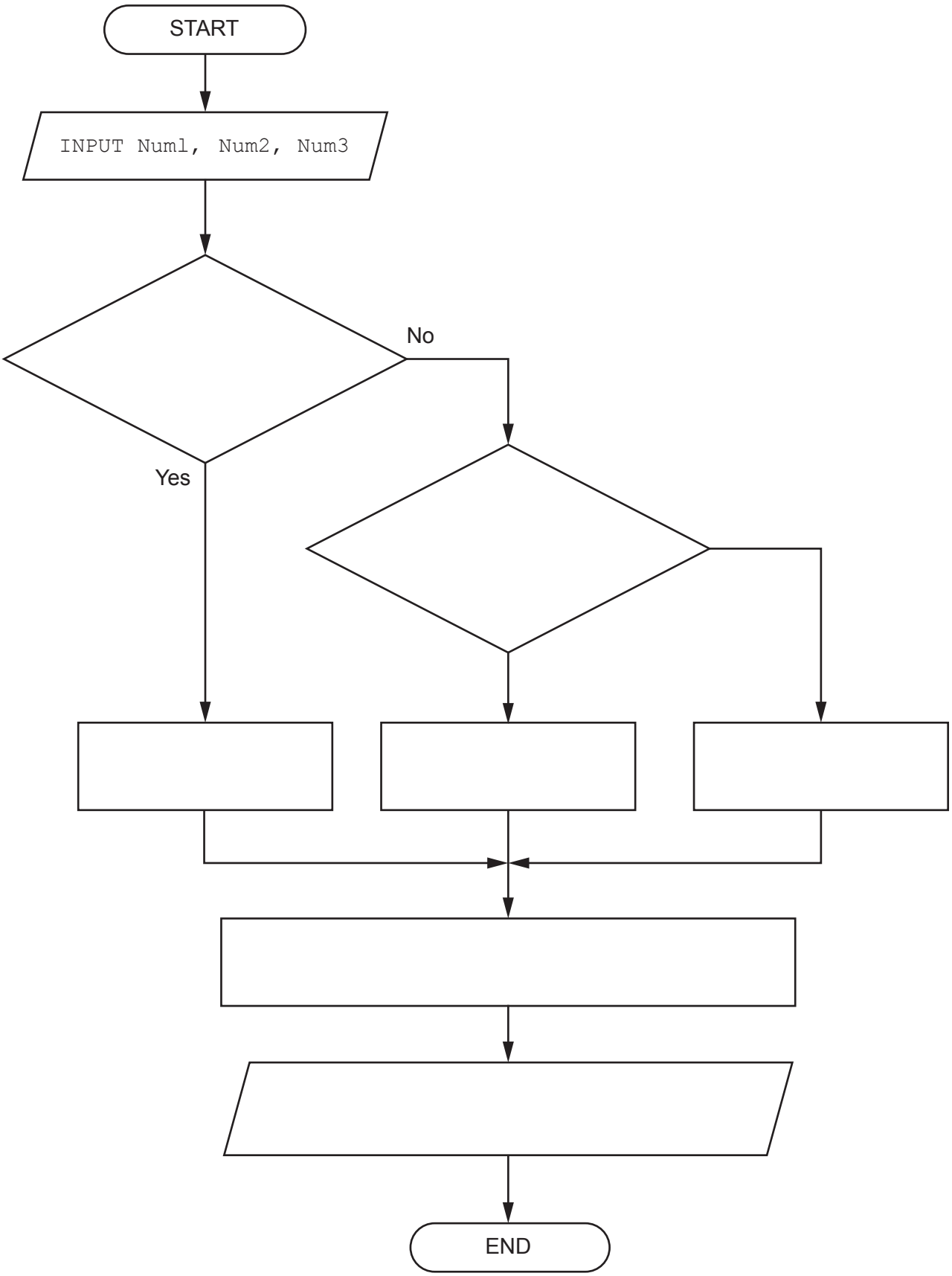
2 A program is being developed.

(a) An algorithm for part of the program will:

- input three numeric values and assign them to identifiers `Num1`, `Num2` and `Num3`
- assign the largest value to variable `Ans`
- output a message giving the largest value and the average of the three numeric values.

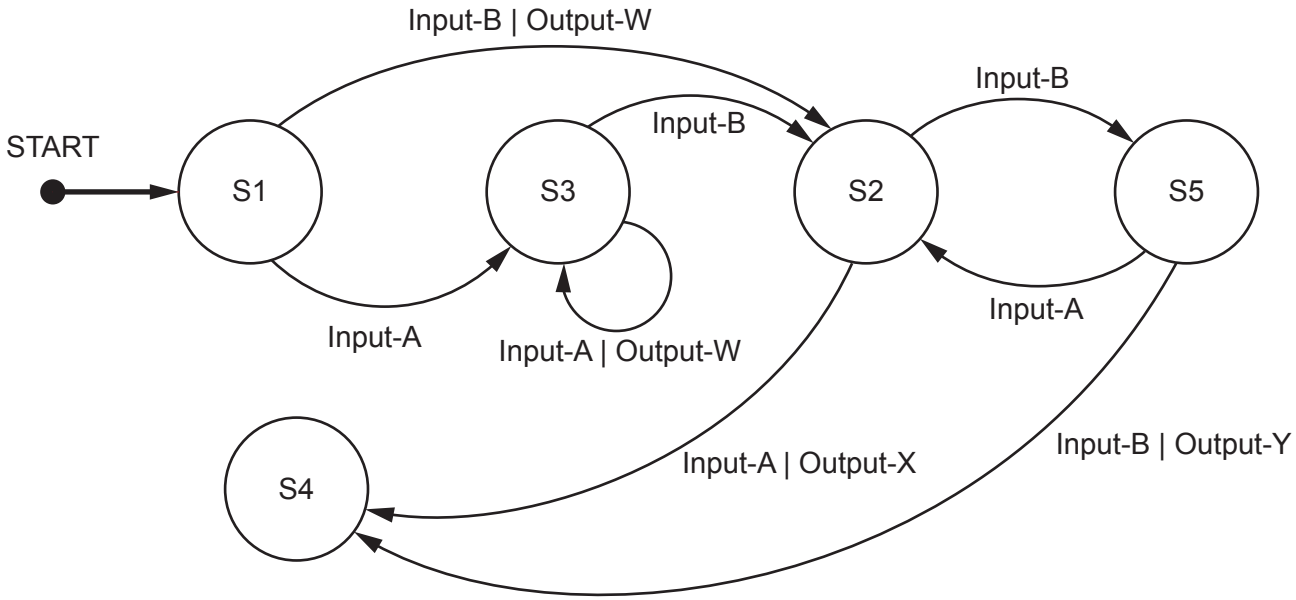
Assume the values are all different and are input in no particular order.

Complete the program flowchart on page 5 to represent the algorithm.



[5]

(b) A different part of the program is represented by the following state-transition diagram.



(i) Complete the table to show the inputs, outputs and next states.

Assume that the current state for each row is given by the 'Next state' on the previous row. For example, the first Input-A is made when in state S1.

If there is no output for a given transition, then the output cell should contain 'none'.

The first two rows have been completed.

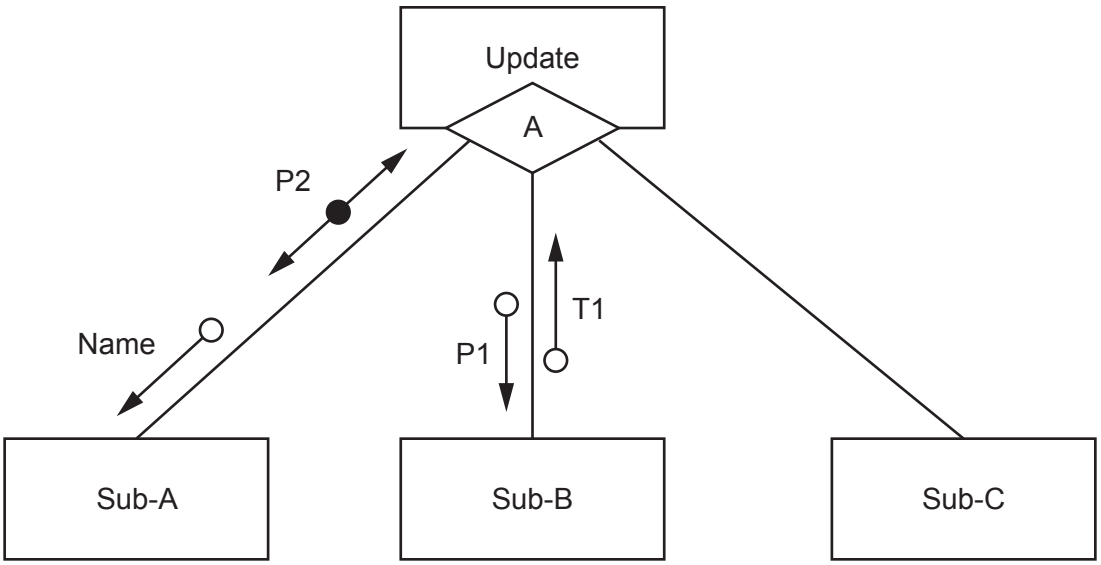
Input	Output	Next state
		S1
Input-A	none	S3
	Output-W	
	none	
Input-B		
Input-A		
		S4

[5]

(ii) Identify the input sequence that will cause the minimum number of state changes in the transition from S1 to S4.

..... [1]

(b) The structure chart illustrates part of the membership program:



Data item notes:

- Name contains the name of a club member
- P1 and T1 are of type real.

(i) Explain the meaning of the diamond symbol (labelled with the letter A) in the chart.

.....

.....

.....

..... [2]

(ii) Write the pseudocode module headers for Sub-A and Sub-B.

Sub-A

.....

.....

Sub-B

.....

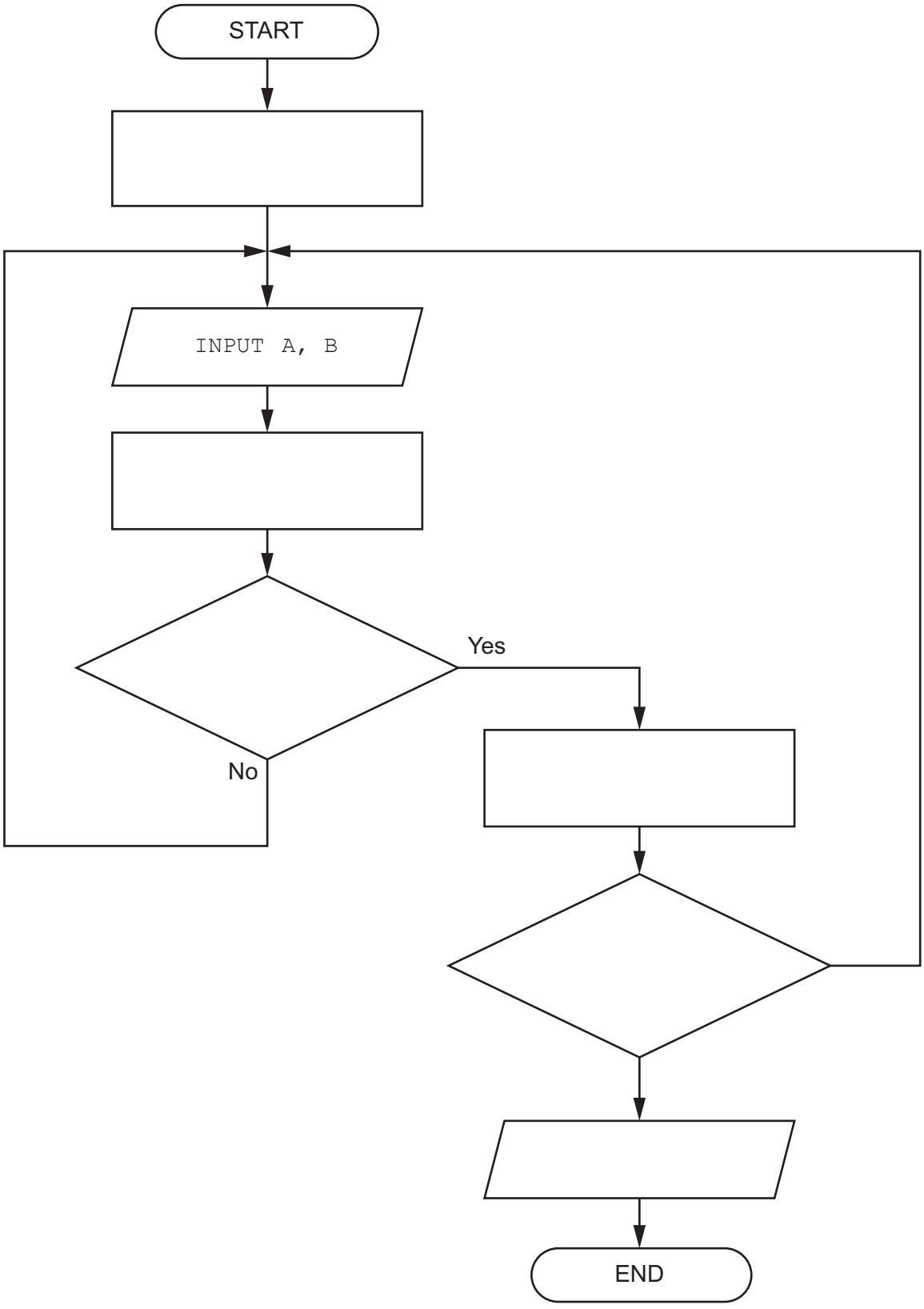
.....

[4]

2 An algorithm has three steps. It will:

1. repeatedly input a pair of numeric values `A` and `B`
2. count the number of pairs that are input until `A` has been greater than `B` 10 times
3. output the number of pairs that were input.

(a) Complete the program flowchart.



[5]

(d) Part of the coffee shop program contains three program modules as follows:

- Module `Init()` has no parameters and returns a Boolean.
- Module `Reset()` takes a string as a parameter and returns an Integer.
- Module `Check()` repeatedly calls `Init()` followed by `Reset()`.

Draw a structure chart to represent the relationship between the **three** modules, including all parameters and return values.

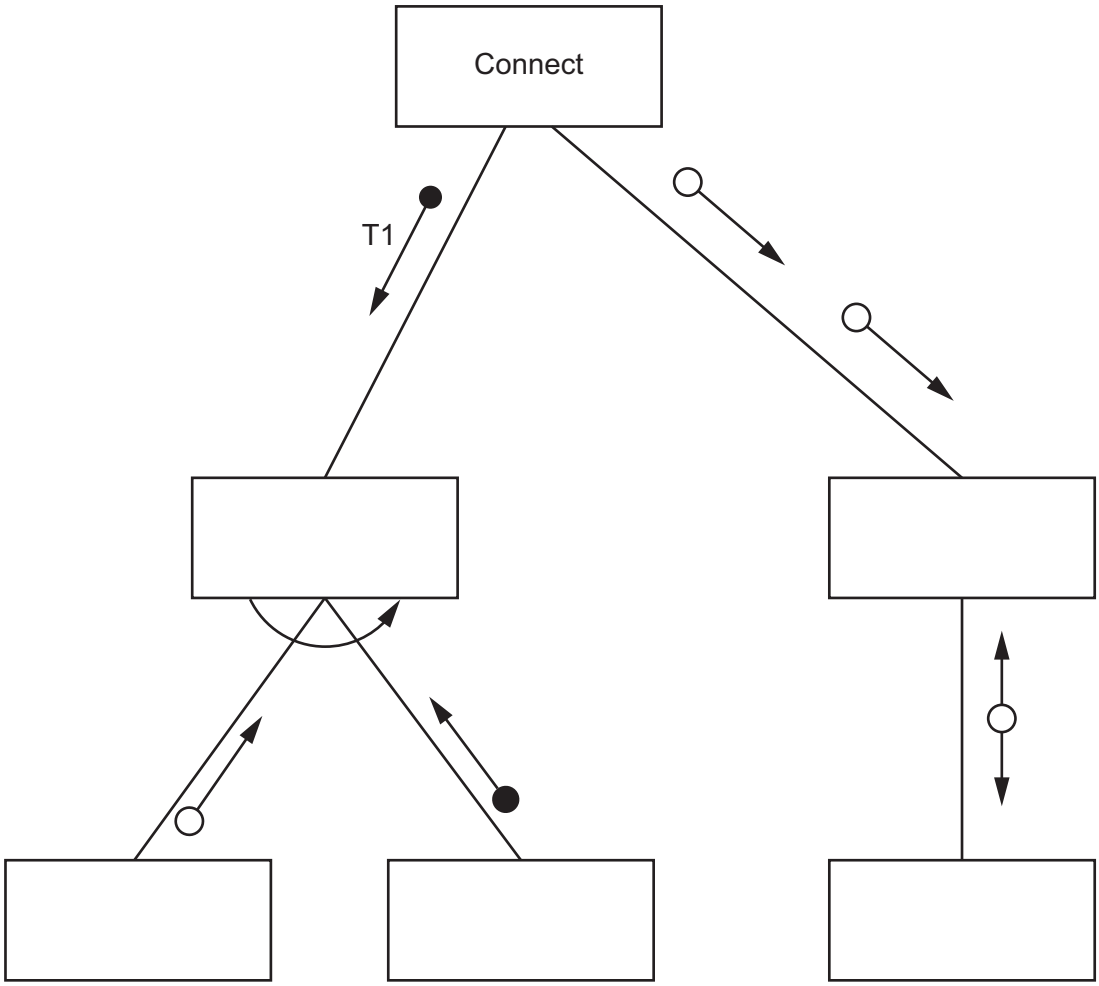
[3]

7 A program contains six modules with headers as follows:

Pseudocode module header
`PROCEDURE Connect()`
`FUNCTION Activate(H1 : STRING, Code : INTEGER) RETURNS BOOLEAN`
`FUNCTION Sync(T1 : BOOLEAN, S2 : REAL) RETURNS INTEGER`
`PROCEDURE Initialise(BYREF ID : INTEGER, BYVAL CC : INTEGER)`
`FUNCTION Reset(RA : STRING) RETURNS INTEGER`
`FUNCTION Enable(SA : INTEGER) RETURNS BOOLEAN`

Module `Connect()` will call either `Activate()` or `Sync()`. This is decided at run-time.

(a) Complete the structure chart for these modules.



[5]

(b) Explain the meaning of the curved arrow symbol used in the diagram in part **(a)**.

.....

..... [2]