4.2 Regular languages

Name: ___________________________

Class: ___________________________

Date: ___________________________

Time: **206 minutes**

Marks: **162 marks**

Comments:

Q1.

Postcodes are used to aid the sorting of mail and help to ensure that mail being sent arrives at the correct destination as quickly as possible.

The format of a UK postcode (ignoring any spaces) is shown in **Figure 1**.

Figure 1

- 1 or 2 letters
- followed by:
  - 1 numeric digit or
  - 2 numeric digits or
  - 1 numeric digit then 1 letter
- followed by 1 numeric digit
- followed by 2 letters

When a post box is emptied in the town of Ipswich the mail in the post box is taken to a central sorting office. Each item is looked at and placed in one of three vans depending upon the postcode written on the envelope.

Postcodes that begin with IP1, IP2, IP3 or IP4 followed by one numeric digit and two letters, eg IP2 8QY, are for mail being sent to an address in the town of Ipswich and go in Van A. Other postcodes that begin with IP, eg IP5 3QW, are for areas not in the town but near to Ipswich and go in Van B. Postcodes that start with anything other than IP, eg CO3 5FN, are not for the Ipswich area and go in Van C. IP postcodes do not use the full range of formats available for UK postcodes.

A finite state machine (FSM) could be used to sort mail using postcodes. **Figure 2** shows a state transition diagram for an FSM used at the Ipswich sorting office.

In **Figure 2**, if a transition is not defined from a state for a particular input symbol then the FSM will stop processing the input and it will be rejected.

(a) If the FSM in **Figure 2** reaches state S12 what does it mean?

____

____

(1)

(b) If the FSM in **Figure 2** finishes at state S11 what does it mean?

____

____

(1)

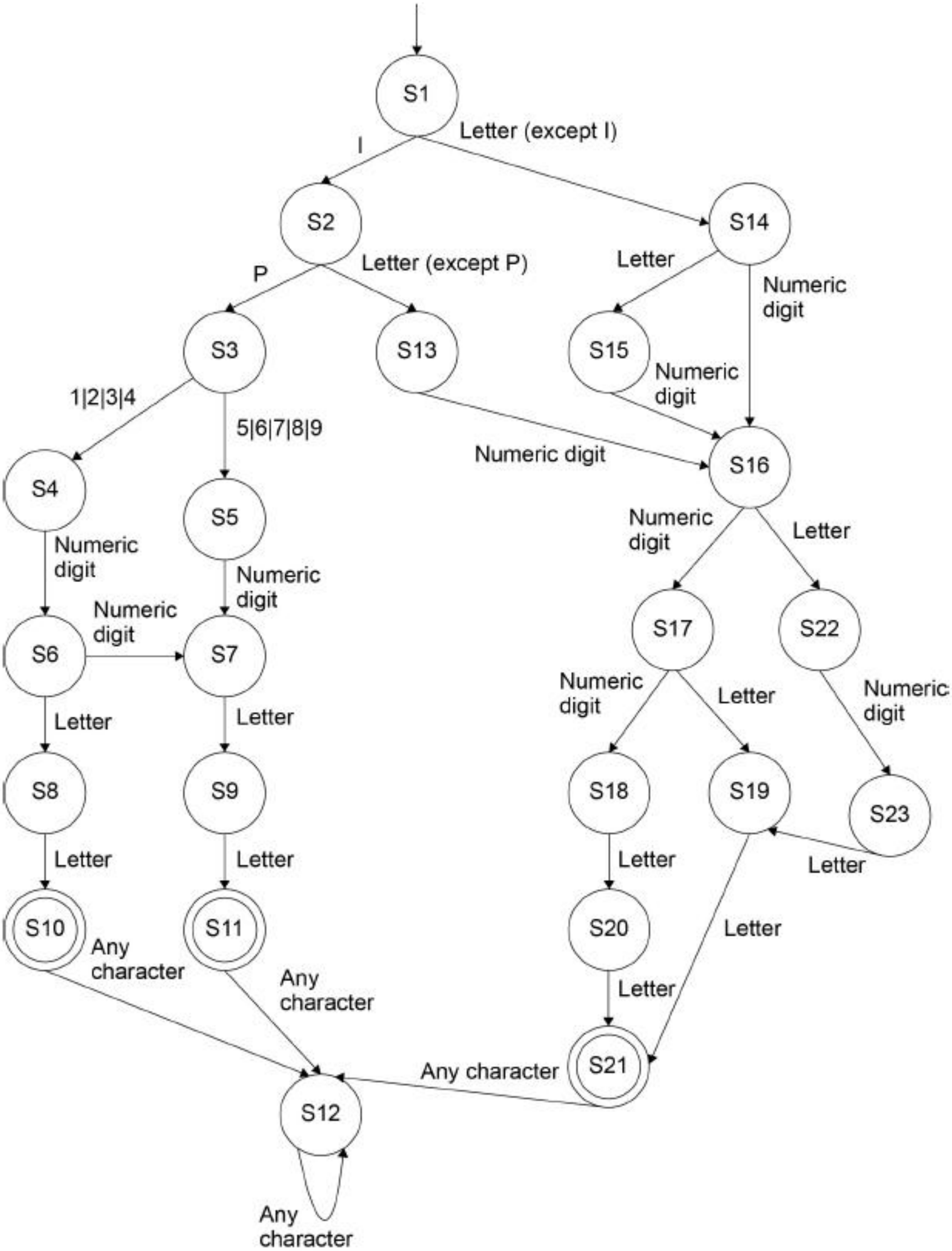
(c) Assuming that the FSM in **Figure 2** can be used to recognise any valid IP postcode, state **one** format used for UK postcodes that IP postcodes do **not** use.

____

____

(1)

Figure 2



(d) The language recognised by an FSM can also be represented by a regular expression. When writing regular expressions `\d` is used to represent any numeric digit and `\a` is used to represent any alphabetic character.

For example, the regular expression `\d \d \a \d` describes the language of all strings that contain two numeric digits followed by one letter and then one numeric digit.

Write a regular expression that represents a valid UK postcode as described in **Figure 1**. In your answer you should only use the | metacharacter once.

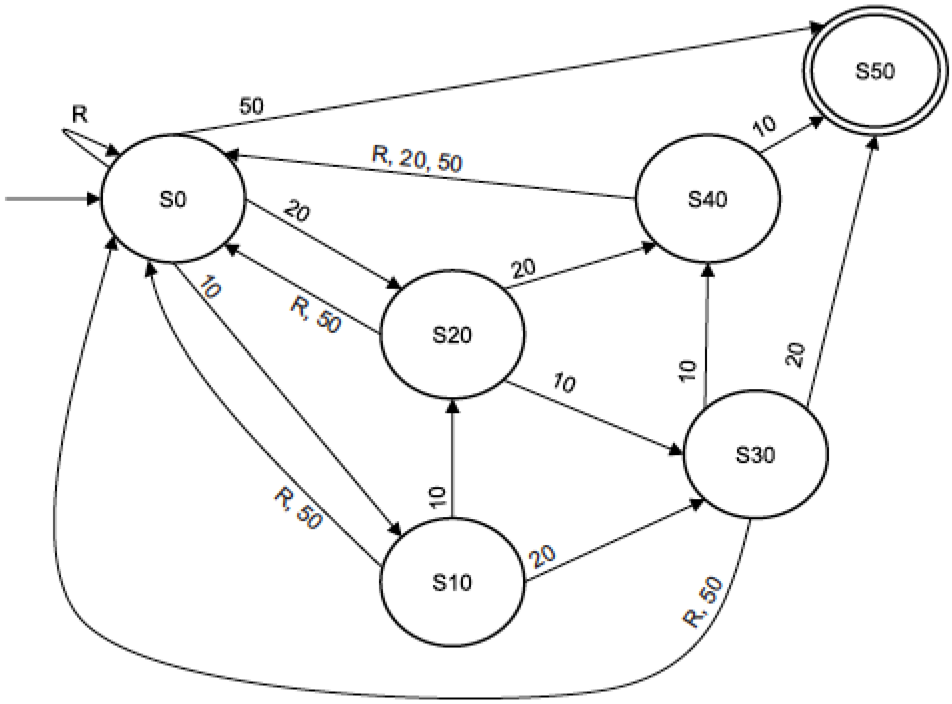
(4)

(Total 7 marks)

Q2.

The figure below shows the state transition diagram of a finite state machine (FSM) used to control a vending machine.

The vending machine dispenses a drink when a customer has inserted exactly 50 pence. A transaction is cancelled and coins returned to the customer if more than 50 pence is inserted or the reject button (R) is pressed. The vending machine accepts 10, 20 and 50 pence coins. Only one type of drink is available.



(a) An FSM can be represented as a state transition diagram or as a state transition table.
The table is an incomplete state transition table for part of the figure.
Complete the table by filling in the unshaded cells.

Original State	Input	New State
S30	10	S40
S30		
S30		
S30		

(2)

(b) The vending machine is to be adapted so that it also accepts 5p coins. What is the minimum number of states that will need to be added to the FSM shown in the figure so that 5p coins are also accepted?

(1)
(Total 3 marks)

Q3.

A bar code scanner is connected to a computerised point of sale system (till). When a product is sold, the bar code that is printed on the product is scanned by the scanner and transmitted to the point of sale system.

This transmission uses asynchronous serial communication and odd parity.

Figure 1 shows the ASCII code for the character "9", which has been read from the bar code, being transmitted to the point of sale system.

(a) Write the missing values of the stop bit, parity bit and start bit on **Figure 1**.

Figure 1

		0	1	1	1	0	0	1	
Stop Bit	Parity Bit	ASCII Code							Start Bit

Direction of data transmission →

(2)

(b) Explain what asynchronous data transmission is and why stop and start bits are required when asynchronous data transmission is used.

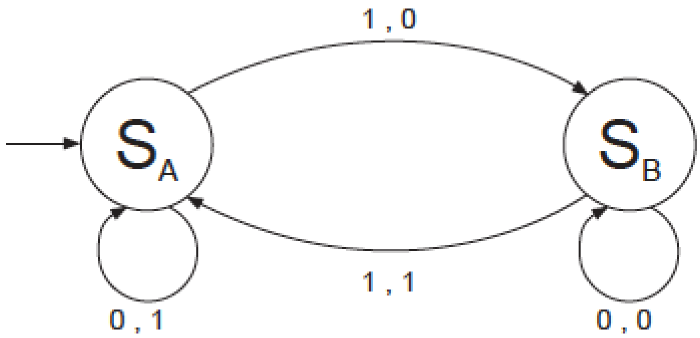
(3)

As part of the process of preparing the data for transmission, the 7-bit ASCII code (0111001) is processed by a Mealy machine (a type of Finite State Machine with output).

The ASCII code is processed from left to right, i.e. the leftmost 0 is the first digit to be processed.

Figure 2 shows a diagram of the Mealy machine. Each transition is labelled with the input symbol that will trigger the transition, followed by a comma, followed by the output that will be produced.

Figure 2



(c) What output is generated by the Mealy machine in **Figure 2** for the input `0111001`?

(1)

(d) The last digit output by the Mealy machine is used in the transmission.

Explain what this last digit represents.

(1)

(e) Serial communication has been chosen instead of parallel communication even though the scanner and point of sale system are located next to each other.

State **two** reasons why this choice is appropriate.

Reason 1

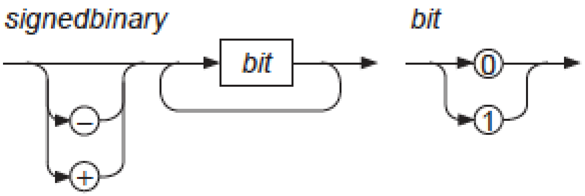
Reason 2 ___

(2)
(Total 9 marks)

Q4.

The table shows three definitions of a language for signed binary numbers, each written using a different standard notation.

All three definitions are supposed to be of the same language for signed binary numbers, but one of them contains an error which means that it defines a different language.

<table>
<tr><td>1</td><td><signedbinary> ::= + <binary> | - <binary> | <binary>
<binary> ::= <bit> | <bit> <binary>
<bit> ::= 0 | 1</td></tr>
<tr><td>2</td><td></td></tr>
<tr><td>3</td><td>$(+\mid-)(0\mid1)^{+}$</td></tr>
</table>

(a) What is the name of the standard notation used in definition **2** in the table?

___ **(1)**

(b) State the number of the definition (**1 to 3**) in the table that does not define the same language as the other two definitions.

☐

(1)

(c) Explain how the language defined by the definition that you have identified in part **(b)** would differ from the language defined by the other two definitions.

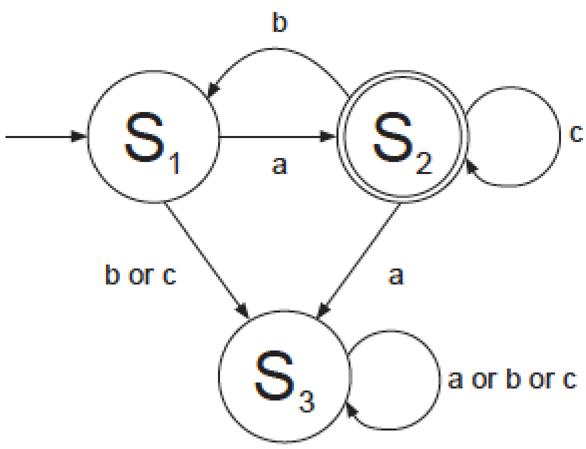
___

___

___

(1)

The diagram shows a finite state automaton that recognises a language.



(d) Write a regular expression that would recognise the same language as the finite state automaton in the diagram.

____

(1)

(Total 4 marks)

Q5.

Figure 1 contains the pseudo-code for a program to output a sequence according to the ‘Fizz Buzz’ counting game.

Figure 1

```
OUTPUT "How far to count?"
INPUT HowFar
WHILE HowFar < 1
  OUTPUT "Not a valid number, please try again."
  INPUT HowFar
ENDWHILE
FOR MyLoop ← 1 TO HowFar
  IF MyLoop MOD 3 = 0 AND MyLoop MOD 5 = 0
  THEN
      OUTPUT "FizzBuzz"
    ELSE
      IF MyLoop MOD 3 = 0
        THEN
          OUTPUT "Fizz"
        ELSE
          IF MyLoop MOD 5 = 0
            THEN
              OUTPUT "Buzz"
            ELSE
              OUTPUT MyLoop
          ENDIF
      ENDIF
  ENDIF
ENDFOR
```


What you need to do:

Write a program that implements the pseudo-code as shown in **Figure 1**.

Test the program by showing the result of entering a value of `18` when prompted by the program.

Test the program by showing the result of entering a value of `-1` when prompted by the program.

Evidence that you need to provide

(a) Your PROGRAM SOURCE CODE for the pseudo-code in **Figure 1**.

(8)

(b) SCREEN CAPTURE(S) for the tests conducted when a value of `18` is entered by the user and when a value of `-1` is entered by the user.

(1)

The main part of the program uses a `FOR` repetition structure.

(c) Explain why a `FOR` repetition structure was chosen instead of a `WHILE` repetition structure.

____

____

(1)

(d) Even though a check has been performed to make sure that the variable `HowFar` is greater than `1` there could be inputs that might cause the program to terminate unexpectedly (crash).

Provide an example of an input that might cause the program to terminate and describe a method that could be used to prevent this.

____

____

____

____

____

____

(3)

(e) Programs written in a high level language are easier to understand and maintain than programs written in a low level language.

The use of meaningful identifier names is one way in which high level languages can be made easier to understand.

State **three** other features of high level languages that can make high level language programs easier to understand.

____

____

____

____

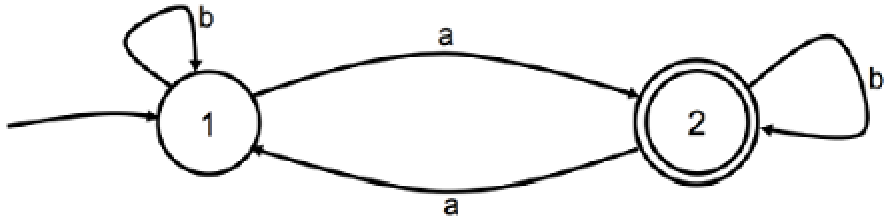
____

____

(3)

(f) The finite state machine (FSM) shown in **Figure 2** recognises a language with an alphabet of `a` and `b`.

Figure 2



Input strings of `a` and `aabba` would be accepted by this FSM.

In the table below indicate whether each input string would be accepted or not accepted by the FSM in **Figure 2**.

If an input string would be accepted write YES.
If an input string would **not** be accepted write NO.

Input string	Accepted by FSM?
aaab	
abbab	
bbbbba	

(2)

(g) In words, describe the language (set of strings) that would be accepted by this FSM shown in **Figure 2**.

____

____

____

____

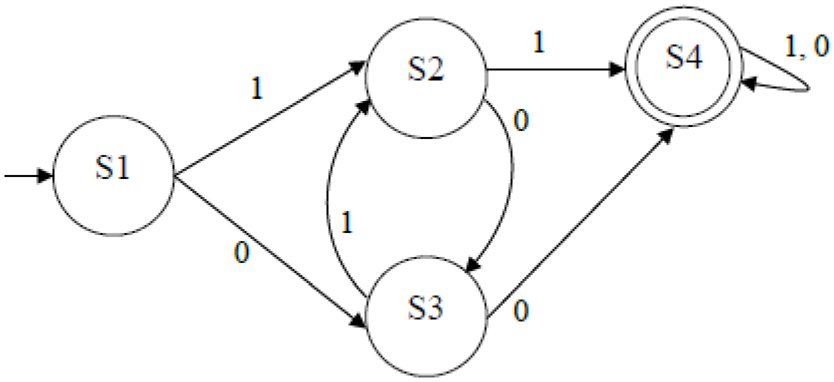
(2)

(Total 20 marks)

Q6.

A finite state machine (FSM) can be used to define a language: a string is allowed in a language if it is accepted by the FSM that represents the rules of the language.
Figure 1 shows the state transition diagram for an FSM.

Figure 1



An FSM can be represented as a state transition diagram or as a state transition table. The table below is an incomplete state transition table for **Figure 1** .

(a) Complete the table.

Original state	Input	New state
S3		
S3		

(1)

(b) Any language that can be defined using an FSM can also be defined using a regular expression.

The FSM in **Figure 1** defines the language that allows all strings containing at least, either two consecutive 1s or two consecutive 0s.

The strings 0110, 00 and 01011 are all accepted by the FSM and so are valid strings in the language.

The strings 1010 and 01 are not accepted by the FSM and so are not valid strings in the language.

Write a regular expression that is equivalent to the FSM shown in **Figure 1**.

____

____

____

____

____

____

(3)

(c) Backus-Naur Form (BNF) can be used to define the rules of a language.

Figure 2 shows an attempt to write a set of BNF production rules to define a language of full names.

Figure 2

Note: underscores (_) have been used to denote spaces.
Note: rule numbers have been included but are not part of the BNF rules.

Rule number

```
1    <fullname> ::= <title>_<name>_<endtitle> |
                    <name> |
                    <title>_<name> |
                    <name>_<endtitle>

2    <title> ::= MRS | MS | MISS | MR | DR | SIR

3    <endtitle> ::= ESQUIRE | OBE | CBE

4    <name> ::= <word> |
                <name>_<word>

5    <word> ::= <char><word>

6    <char> ::= A | B | C | D | E | F | G | H | I |
                J | K | L | M | N | O | P | Q | R |
                S | T | U | V | W | X | Y | Z
```

BNF can be used to define languages that are not possible to define using regular expressions. The language defined in **Figure 2** could not have been defined using regular expressions.

Complete the table below by writing either a **'Y'** for **Yes** or **'N'** for **No** in each row.

Rule number (given in Figure 2)	Could be defined using a regular expression
1	
2	
3	
4	
5	
6	

(1)

(d) There is an error in rule 5 in **Figure 2** which means that no names are defined by the language.

Explain what is wrong with the production rule and rewrite the production rule so that the language does define some names – the names 'BEN D JONES', 'JO GOLOMBEK' and 'ALULIM' should all be defined.

____

(2)
(Total 7 marks)

Q7.

(a) Represent the **denary** number 55 using **8-bit unsigned binary**.

(1)

(b) Represent the **denary** number 55 using **hexadecimal**.

(1)

(c) Why are bit patterns often displayed using hexadecimal instead of binary?

(1)

(d) Represent the denary number -59 as an **8-bit two's complement binary integer.**

(2)

(e) Represent the denary number 5.625 as an **unsigned binary fixed point number** with three bits before and five bits after the binary point.

(2)

(f) The ASCII system uses 7 bits to represent a character. The ASCII code in denary for the numeric character '0' is 48; other numeric characters follow on from this in sequence.

The numeric character '0' is represented using 7 bits as `0110000`.

Using 7 bits, express the ASCII code for the character '6' in binary.

____

____

(1)

(g) How many different character codes can be represented using 7-bit ASCII?

____

____

(1)

(h) Examples of logical bitwise operators include `AND`, `OR`, `NOT` and `XOR`.

Describe how **one** of these logical bitwise operators can be used to convert the 7-bit ASCII code for a numeric character into a 7-bit pure binary representation of the number, (eg `0110001` for the numeric character '1' would be converted to `0000001`).

____

____

____

____

(2)

(i) Characters are transmitted using an 8-bit code that includes the 7-bit ASCII code and a single parity bit in the most significant bit. A parity bit is added for error checking during data transmission.

Using even parity, what 8-bit code is sent for the numeric character '0'?

____

____

(1)

(j) Hamming code is an alternative to the use of a single parity bit. Hamming code uses multiple parity bits - this allows it to correct some errors that can occur during transmission.

The parity bits are located in the power of two bit positions (1, 2, 4, 8, etc.). The other bit positions are used for the data bits.

Describe how the receiver can detect and correct a single-bit error using Hamming code.

____

____

____

(4)

(k) **Figure 1** shows the bit pattern **received** in a communication that is using **even parity** Hamming code. The data bits received are `1101000`.

Figure 1

Bit position	11	10	9	8	7	6	5	4	3	2	1
Bit	1	1	0	0	1	0	0	0	0	0	1

There has been a single-bit error in the data transmission.

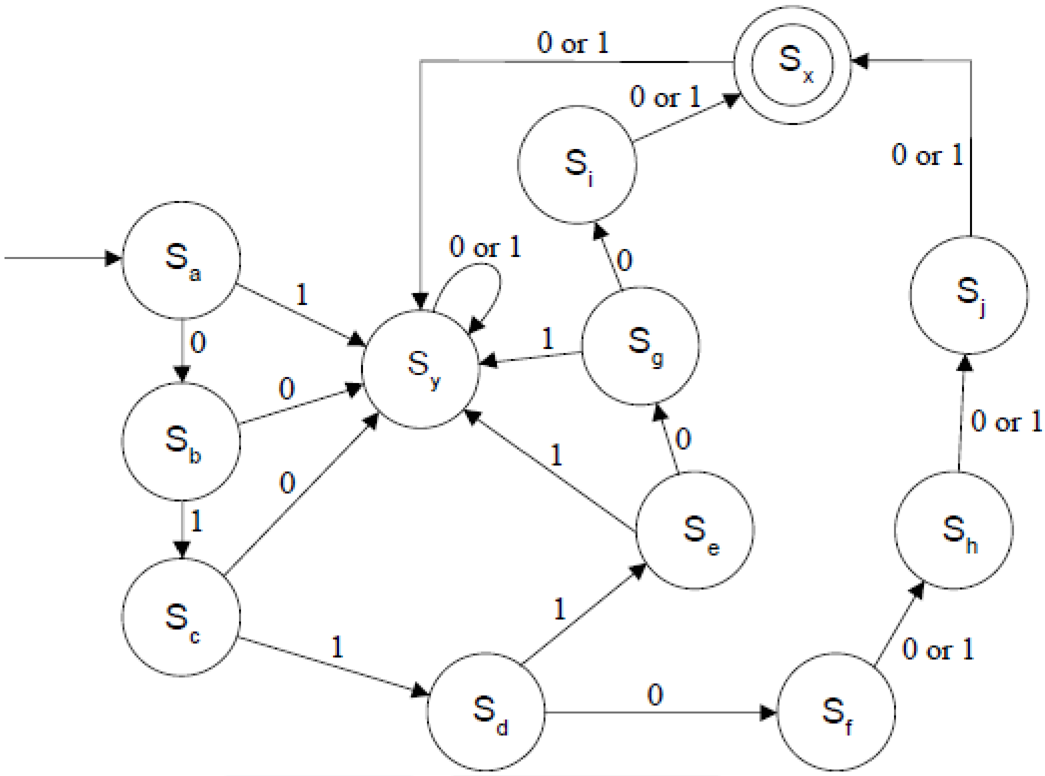
Which bit position from the bit pattern in **Figure 1** contains an error?

(1)

Figure 2 shows a Finite State Machine (FSM) represented as a state transition diagram.

The machine takes a bit pattern as an input. The bit pattern is considered to be valid if the machine ends up in the accepting state S_x. The bit patterns `0111000` and `0110001` are valid; the bit patterns `010011` and `01110111` are not valid. Bit patterns are processed starting with the left-most bit.

Figure 2



(l) Is the bit pattern `0111010` a valid input for the FSM shown in **Figure 2**?

___

___

(1)

(m) A finite state machine can also be represented as a state transition table. The table below shows part of the state transition table that represents the finite state machine shown in **Figure 2**. The state transition table is only partially complete.

Initial State	Input	New State
S_g	1	
S_y		
S_y		

Complete the table by filling in the unshaded cells.

(2)

(n) What is the purpose of the FSM shown in **Figure 2**?

___

___

(1)

(o) The state S_i in the FSM shown in **Figure 2** is not necessary and is going to be removed from the FSM.

Describe the change that needs to be made so that state S_i can be removed without changing the functionality of the FSM.

____

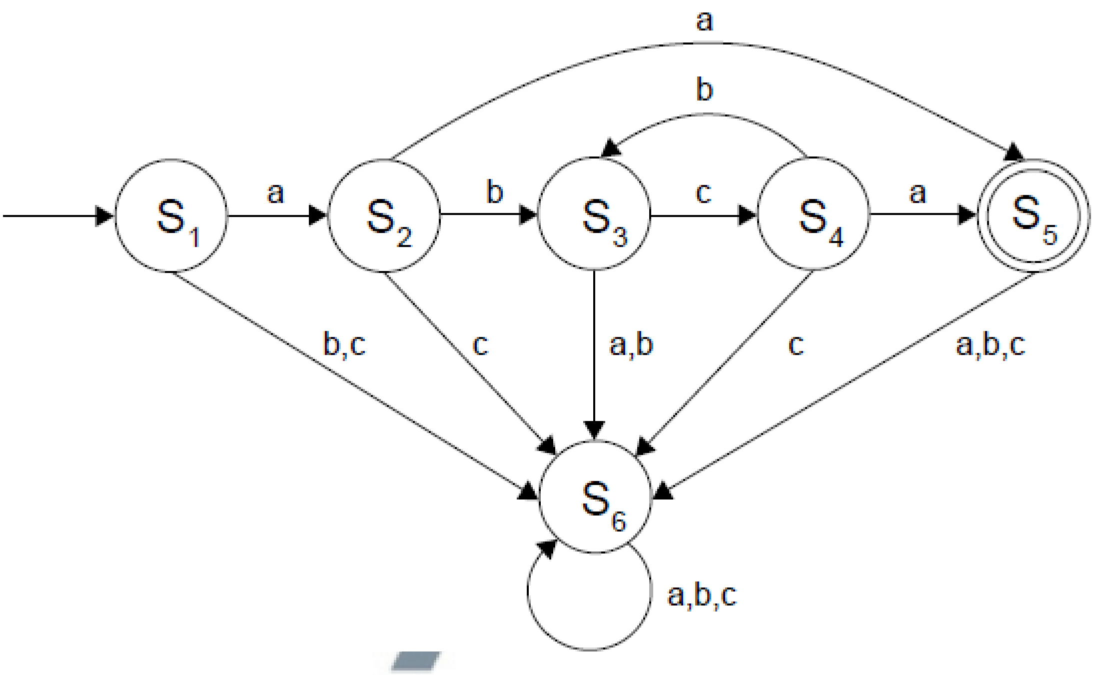
____

(1)

(Total 22 marks)

Q8.

The diagram below shows a Finite State Automaton (FSA). The FSA has input alphabet {a, b, c} and six states, S_1, S_2, S_3, S_4, S_5 and S_6.



(a) Complete the empty cells in the **part of the transition table shown** below for the FSA in the diagram.

Current State	S_1	S_1	S_1	S_2	S_2	S_2	S_3	S_3	S_3
Input Symbol	a	b	c	a	b	c			
Next State	S_2	S_6	S_6	S_5	S_3	S_6			

(1)

(b) Give the name of the state that the FSA will end up in when processing the string `abcb`.

____

(1)

(c) Describe the purpose of state S_6.

____

____

(1)

(d) The FSA in the diagram above accepts strings that consist of:

- a letter `a`
- followed by zero or more occurrences of the string `bc` and
- ending with a second letter `a`.

For any FSA, it is possible to write a regular expression that will match the same language (set of strings) as the FSA.

Write a regular expression that will match the same language that is accepted by the FSA above.

____

____

(2)

(e) The Turing Machine is a more powerful abstract model of computation than the FSA.

Explain why the Turing Machine model can be used to recognise a greater range of languages than an FSA could.

____

____

(1)

(Total 6 marks)

Q9.

Regular expressions can be used to search for strings.

(a) For each of the following regular expressions, describe the set of strings that they would match.

(i) `b*c`

____

____

(1)

(ii) `b?c`

____

____

(1)

(b) Write a regular expression that matches the letter `b`, followed by zero or more occurrences of the string `cd` followed by either a single letter `e` or the string `fg`.

____

____

(2)

(Total 4 marks)

Q10.

(a) What is the denary equivalent of the hexadecimal number A7?

You may use the space below for rough working. You may get some marks for your working, even if your answer is incorrect.

Answer ____________________

(2)

(b) Represent the denary value 7.625 as an **unsigned binary fixed point** number, with 4 bits before and 4 bits after the binary point.

Use the space below for rough working.

Answer ____________________

(2)

(c) Represent the denary value -18 as an **8-bit two's complement binary integer**.

Use the space below for rough working.

Answer ____________________

(2)

(d) What is the **largest positive denary value** that can be represented using **8-bit two's complement binary**?

Use the space below for rough working.

Answer ____________________

(1)

(e) Describe how **8-bit two's complement binary** can be used to subtract one number from another number. In your answer you must show how the calculation 23 – 48 would be completed using the method that you have described.

You may use the space below for rough working.

Answer ____________________

(4)

Figure 1 shows a state transition diagram for a finite state machine (FSM).

Table 1 shows the outputs produced by the finite state machine in **Figure 1** for some possible input strings. Some of the outputs are missing from the table below. Input strings are processed starting with the right-most bit.

Figure 1

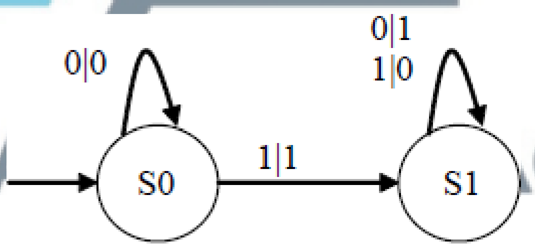


Table 1

Input string	Output string
00010011	11101101
00010010	**(a)**
00010100	11101100
00010101	**(b)**

(f) What output string should be in position **(a)** in the table?

____

____

(1)

(g) What output string should be in position **(b)** in the table?

____

____

(1)

(h) What is the purpose of the finite state machine shown in **Figure 1**?

____

____

(1)

(i) A finite state machine can be represented as a state transition diagram or as a state transition table. **Table 2** is an incomplete state transition table for **Figure 1**.

Complete the **unshaded** cells in the table below.

Table 2

Input	Original state	Output	New state
0	S0	0	S0
1		1	S1
0	S1		S1

(3)

(Total 17 marks)

Q11.

A particular Turing machine has states S_B, S_0, S_1, S_R and S_T. S_B is the start state and S_T is the stop state. The machine stores data on a single tape which is infinitely long in one direction. The machine's alphabet is 0, 1, #, x, y and □ where □ is the symbol used to indicate a blank cell on the tape.

The transition rules for this Turing machine can be expressed as a transition function δ. Rules are written in the form:

δ(Current State, Input Symbol) = (Next State, Output Symbol, Movement)

So, for example, the rule:

$\delta(S_B, 1) = (S_1, y, \longrightarrow)$

means:

IF the machine is currently in state S_B AND the input symbol read from the tape is 1

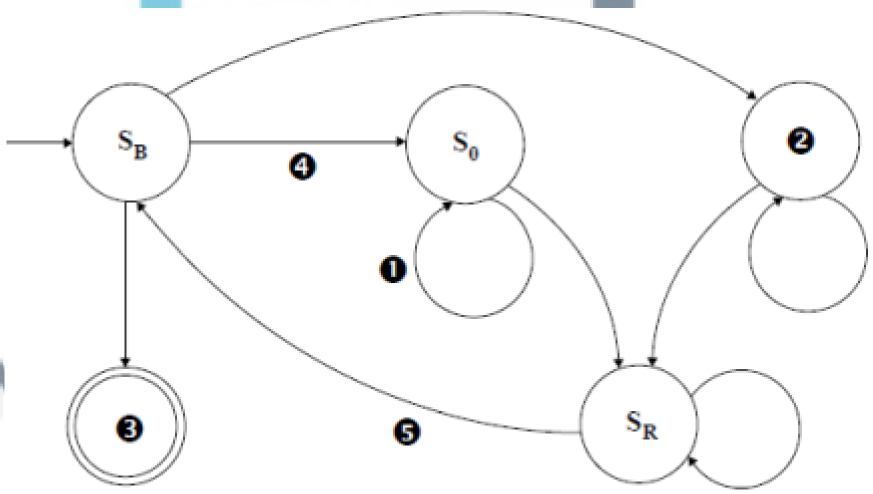
THEN the machine should change to state S_1, write a y to the tape and move the read/ write head one cell to the right

The machine's transition function, δ, is defined by:

$\delta(S_B, 0) = (S_0, x, \rightarrow)$	$\delta(S_1, 0) = (S_1, 0, \rightarrow)$
$\delta(S_B, 1) = (S_1, y, \rightarrow)$	$\delta(S_1, 1) = (S_1, 1, \rightarrow)$
$\delta(S_B, \#) = (S_T, \#, \rightarrow)$	$\delta(S_1, \#) = (S_1, \#, \rightarrow)$
	$\delta(S_1, \square) = (S_R, 1, \leftarrow)$
$\delta(S_0, 0) = (S_0, 0, \rightarrow)$	
$\delta(S_0, 1) = (S_0, 1, \rightarrow)$	$\delta(S_R, 0) = (S_R, 0, \leftarrow)$
$\delta(S_0, \#) = (S_0, \#, \rightarrow)$	$\delta(S_R, 1) = (S_R, 1, \leftarrow)$
$\delta(S_0, \square) = (S_R, 0, \leftarrow)$	$\delta(S_R, \#) = (S_R, \#, \leftarrow)$
	$\delta(S_R, x) = (S_B, 0, \rightarrow)$
	$\delta(S_R, y) = (S_B, 1, \rightarrow)$

Figure 1 shows an unlabelled finite state transition diagram for this machine. Some of the state transition arrows represent more than one of the machine's transition rules. For example, the arrow labeled ❶ represents the three rules: $\delta(S_0, 0) = (S_0, 0, \rightarrow)$, $\delta(S_0, 1) = (S_0, 1, \rightarrow)$ and $\delta(S_0, \#) = (S_0, \#, \rightarrow)$.

Figure 1



(a) (i) Which states are represented by the labels ❷ and ❸ in **Figure 1**?

❷ ______________________ ❸ ______________________

(1)

(ii) Which of the machine's transition rule(s) is / are represented by the arrow labelled ❹ in **Figure 1**?

____

____

(1)

(iii) Which of the machine's transition rule(s) is / are represented by the arrow labelled ❺ in **Figure 1**?

____

(1)

The machine's transition rule, δ, is repeated here so that you can answer part **(b)** without having to turn back in the question paper booklet.

$\delta(S_B, 0) = (S_0, x, \rightarrow)$	$\delta(S_1, 0) = (S_1, 0, \rightarrow)$
$\delta(S_B, 1) = (S_1, y, \rightarrow)$	$\delta(S_1, 1) = (S_1, 1, \rightarrow)$
$\delta(S_B, \#) = (S_T, \#, \rightarrow)$	$\delta(S_1, \#) = (S_1, \#, \rightarrow)$
	$\delta(S_1, \square) = (S_R, 1, \leftarrow)$
$\delta(S_0, 0) = (S_0, 0, \rightarrow)$	
$\delta(S_0, 1) = (S_0, 1, \rightarrow)$	$\delta(S_R, 0) = (S_R, 0, \leftarrow)$
$\delta(S_0, \#) = (S_0, \#, \rightarrow)$	$\delta(S_R, 1) = (S_R, 1, \leftarrow)$
$\delta(S_0, \square) = (S_R, 0, \leftarrow)$	$\delta(S_R, \#) = (S_R, \#, \leftarrow)$
	$\delta(S_R, x) = (S_B, 0, \rightarrow)$
	$\delta(S_R, y) = (S_B, 1, \rightarrow)$

(b) This Turing machine is carrying out a computation. The machine starts in state S_B with the string 01# on the tape. All other cells contain the blank symbol, □ (not shown).

Trace the computation of the Turing machine, using the transition function δ. Show the contents of the tape, the current position of the read / write head and the current state as the input symbols are processed. The first three steps and final state have been completed for you.

								State									State
1.	0	1	#				...	S_B	9.							...	
2.	x	1	#				...	S_0	10.							...	
3.	x	1	#				...	S_0	11.							...	
4.							...		12.							...	
5.							...		13.							...	
6.							...		14.							...	
7.							...		15.							...	
8.							...		16.							...	S_T

(6)

(i) Describe the purpose of the symbols x and y in this Turing machine's alphabet.

____

____

(1)

(ii) What does the Turing machine do?

____

____

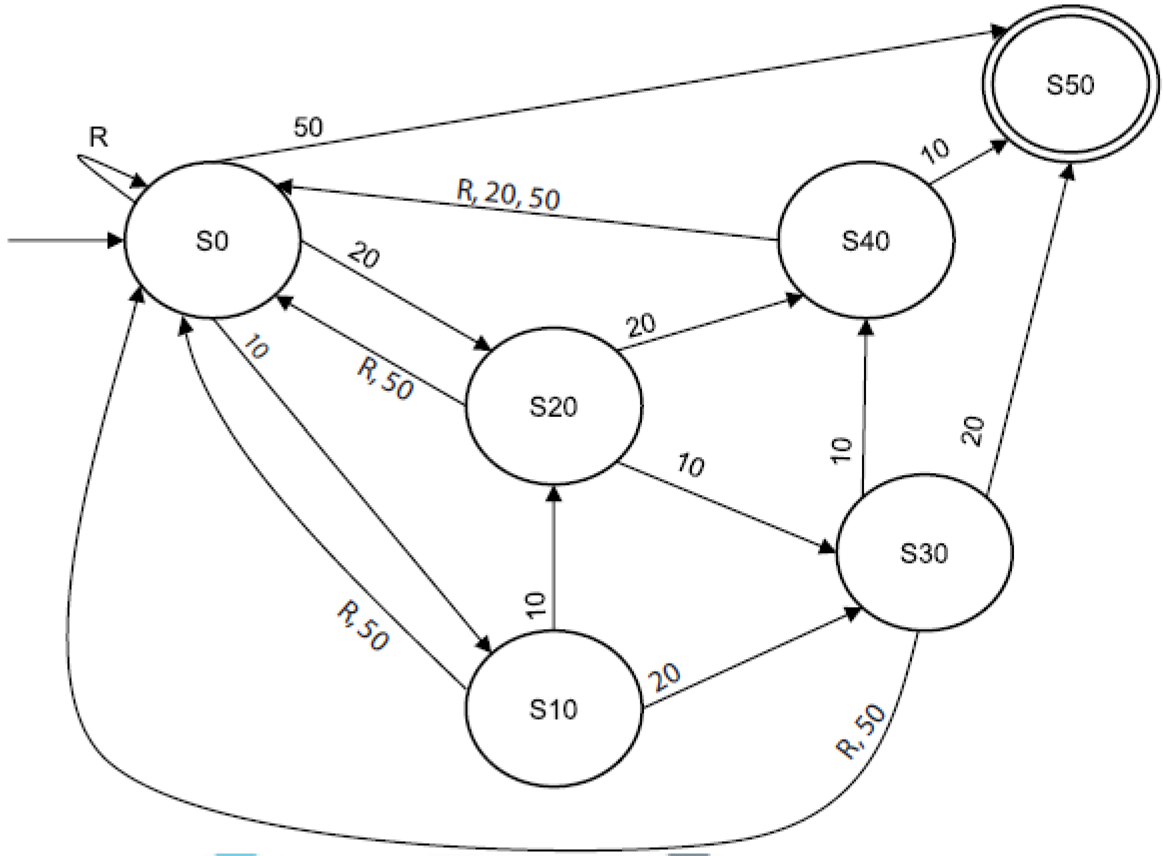
(1)
(Total 11 marks)

Q12.

The diagram below shows the state transition diagram of a finite state machine (FSM) used to control a vending machine.

The vending machine dispenses a drink when a customer has inserted exactly 50 pence. A transaction is cancelled and coins returned to the customer if more than 50 pence is

inserted or the reject button (R) is pressed. The vending machine accepts 10, 20 and 50 pence coins. Only one type of drink is available.
The only acceptable inputs for the FSM are 10, 20, 50 and R.



(a) An FSM can be represented as a state transition diagram or as a state transition table. The table below is an incomplete state transition table for part of the diagram above.

Complete the missing sections of the four rows of the table below.

Original state	**Input**	**New state**
S0	10	S10
S0		
S0		
S0		

(3)

(b) There are different ways that a customer can provide **exactly three** inputs that will result in the vending machine dispensing a drink. Three possible permutations are “20, 10, 20”, “10, R, 50” and “10, 50, 50”.

List **four** other possible permutations of **exactly three** inputs that will be accepted by the FSM shown in the diagram above.

___

(4)

(Total 7 marks)

Q13.

The diagram below shows some production rules that have been used to define the syntax of valid mathematical expressions in a particular programming language.

```
<expression> ::= <factor> | <factor> * <factor> | <factor> / <factor>
<factor> ::= <term> | <term> + <term> | <term> - <term>
<term> ::= - <expression> | <number>
<number> ::= <digit> | <digit> <number>
<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

(a) What notation method has been used in the diagram above?

(1)

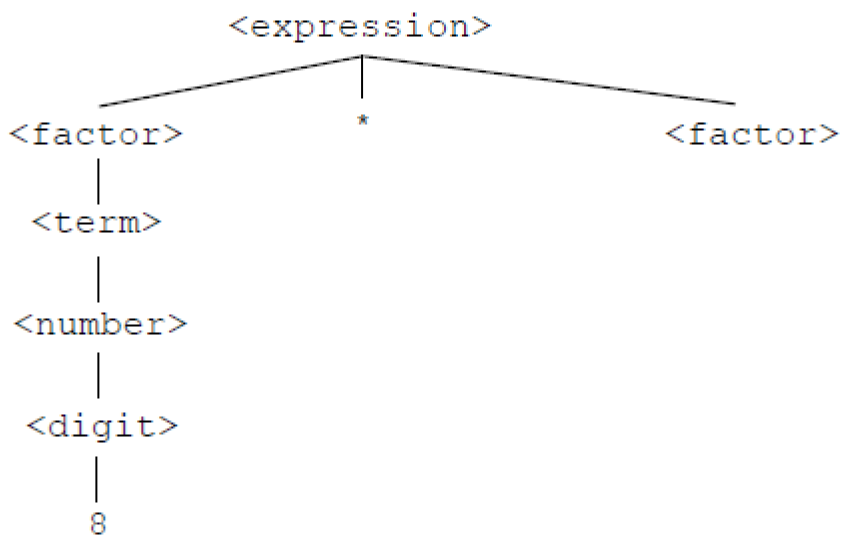
(b) Complete the table below by writing **Yes** or **No** in the empty column to indicate whether or not the strings are valid examples of the statement types from the diagram above.

Statement type	String	Valid (Yes / No)
<number>	129.376	
<factor>	23 + 17	

(2)

(c) A tree can be used to demonstrate that an `<expression>` is valid. This is known as a parse tree.

Complete the parse tree below to show that `8 * 4 + 21` is a valid `<expression>`.



(3)
(Total 6 marks)

Q14.

A particular long-distance data transmission system transmits data signals as electrical voltages using copper wire.

(a) What is the relationship between the bandwidth of the copper wire and the bit rate at which the data can be transmitted?

____

____

____

(1)

(b) The system is affected by latency.

What is *latency* in the context of data communications?

____

____

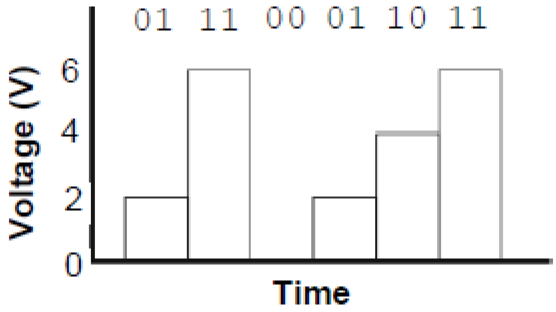
(1)

The system uses four different voltage levels so that two data bits can be transmitted with each signal change.

The table below shows the signal levels (in volts) that the system uses for particular binary patterns.

Binary pattern	Signal level (volts)
00	0
01	2
10	4
11	6

Using this system, the binary pattern 011100011011 would be transmitted as the voltage sequence 2,6,0,2,4,6 as shown in the graph below:



(c) What, **precisely**, is the relationship between the bit rate and the baud rate for this system?

____

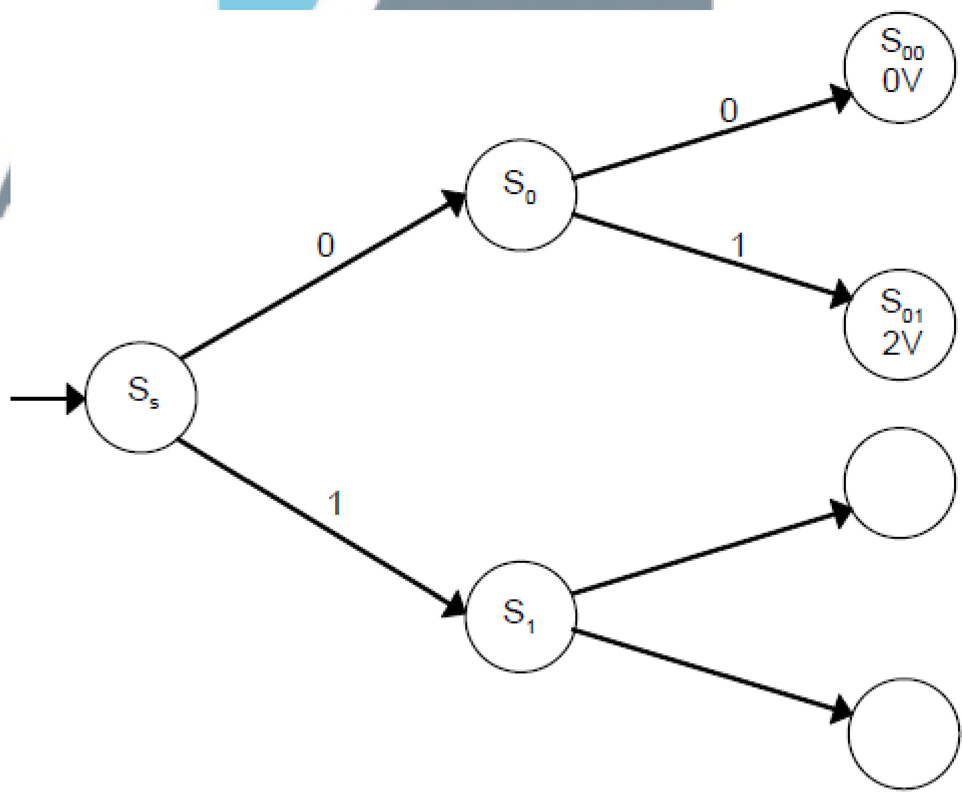
____

(1)

(d) A Moore machine is a type of finite state machine that produces output. The transitions are labelled with the inputs and each state is labelled with a name and the output that it produces; if a particular state has no output then it is labelled with just a name.

The diagram below shows an incomplete diagram of a Moore machine that will convert a two-bit binary code into the signal level (in volts) that is transmitted to represent it, as listed in the table above.

Complete the diagram below. Label all of the transitions and the states that are currently unlabelled. The machine should work for the four binary patterns 00, 01, 10 and 11.



(4)
(Total 7 marks)

Q15.

(a) Complete the missing parts of the question posed by the Halting problem in the diagram below.

Is it possible in general to ______________________________________ that can tell, given any program and its inputs and without ____________________________________, whether the given program with its given inputs will halt?

(2)

(b) What is the significance of the Halting problem?

____

____

(1)

(Total 3 marks)

Q16.

Regular expressions can be used to search for strings. For example, `de(f|g)*h+` matches any string that starts with `de` and is followed by zero or more instances of either `f` or `g` followed by one or more instances of `h`.

Write regular expressions that will match:

(a) any string that starts with a letter `a`, ends with a letter `c` and has one or more occurrences of the letter `b` in the middle of it, ie the expression should match the strings `abc`, `abbc`, `abbbc` and so on.

____

____

(1)

(b) any string that starts with either a `0` or a `1`, followed by zero or more occurrences of the digit `1` ie the expression should match the strings `0`, `1`, `01`, `11`, `011` and so on.

____

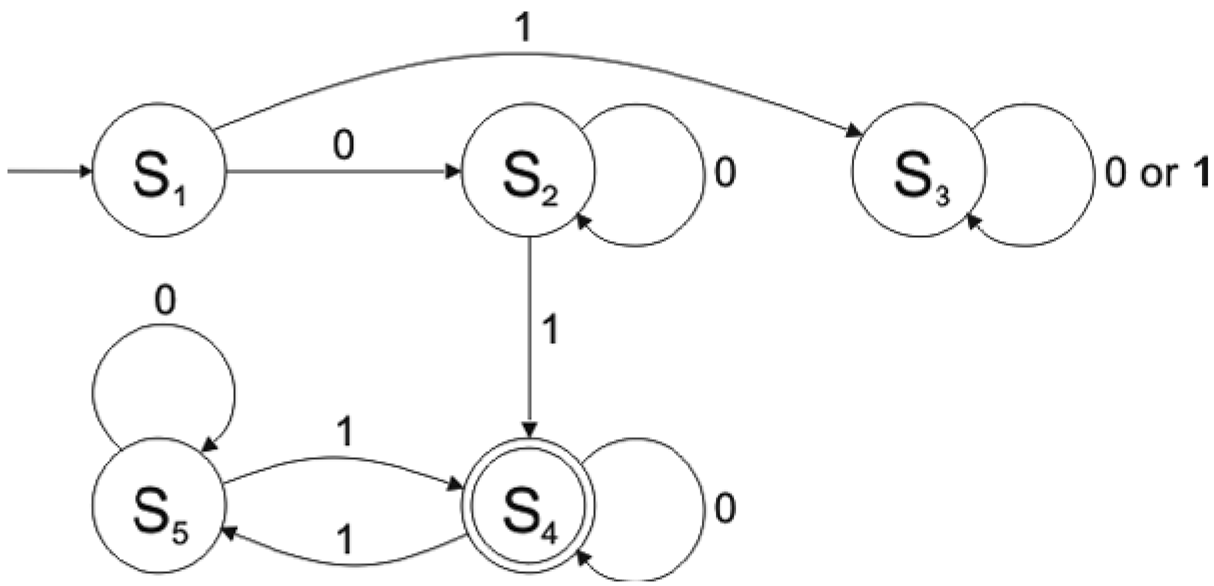
____

(1)

(Total 2 marks)

Q17.

The diagram below shows a Finite State Automaton (FSA). The FSA has input alphabet {0, 1} and five states, S_1, S_2, S_3, S_4 and S_5.



(a) Complete the transition table below for the FSA in the diagram above.

Current State	S_1	S_1	S_2	S_2	S_3	S_3	S_4	S_4	S_5	S_5
Input Symbol	0	1	0	1	0	1				
Next State	S_2	S_3	S_2	S_4	S_3	S_3				

(1)

(b) The state S_4 is a special state. This is indicated by the double circle in the diagram. What does the double circle signify?

____

____

(1)

(c) Write **Yes** or **No** in each row of the table below to indicate whether or not each of the four input strings would be accepted by the FSA in the diagram above.

Input String	**String Accepted? (Yes / No)**
101	
000	
010001101	
0100011011	

(2)

(d) Describe the language (set of strings) that the FSA will accept.

(2)
(Total 6 marks)

Q18.

Regular expressions can be used to search for strings.

(a) For each of the following regular expressions, describe the set of strings that they would find.

(i) `a+b`

(1)

(ii) `a?b`

(1)

(iii) `(ab)*`

(1)

(b) Write regular expressions that match:

(i) either `Clare` or `Claire`.

(1)

(ii) any non-empty string that:

- starts with `10`
- has zero or more occurrences of any combination of `0` or `1` in the middle
- ends with `01`

Example strings that the expression should match are `1001`, `100010101`, `101111010101001`.

(2)

Q19.

A particular Turing machine has states S_1, S_2 and S_3.
S_1 is the start state and S_3 is the stop state.
The machine uses one tape which is infinitely long in one direction to store data.
The machine's alphabet is 0, 1, o, e and □, where □ is the symbol used to indicate a blank cell on the tape.

The transition rules for this Turing machine can be expressed as a transition function δ.
Rules are written in the form:

δ(Current State, Input Symbol) = (Next State, Output Symbol, Movement)

So, for example, the rule:

$\delta\ (S_1, 0) = (S_1, 0, \rightarrow)$

means

IF	the machine is currently in state S_1 AND the input symbol read from the tape is 0
THEN	the machine should remain in state S_1, write a 0 to the tape and move the read/write head one cell to the right

The machine's transition function, δ, is defined by:

$\delta\ (S_1, 0) = (S_1, 0, \rightarrow)$

$\delta\ (S_1, 1) = (S_2, 1, \rightarrow)$

$\delta\ (S_1, \square) = (S_3, e, \rightarrow)$

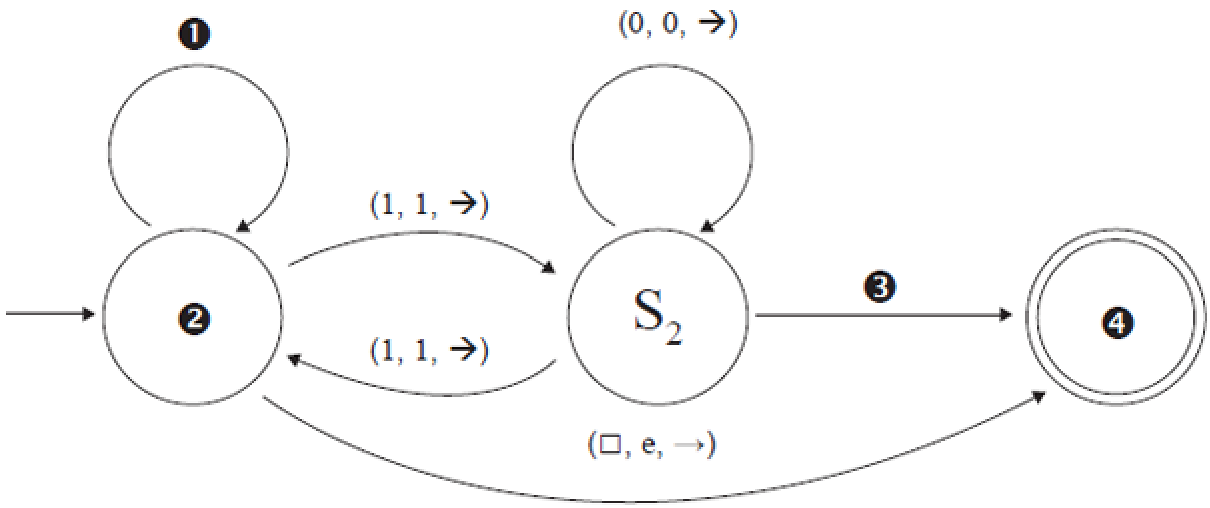
$\delta\ (S_2, 0) = (S_2, 0, \rightarrow)$

$\delta\ (S_2, 1) = (S_1, 1, \rightarrow)$

$\delta\ (S_2, \square) = (S_3, 0, \rightarrow)$

The diagram below shows a partially labelled finite state transition diagram for this machine.

Some labels are missing and have been replaced by numbers such as ❶. Each state transition arrow is labelled with the input symbol, the output symbol and the direction of movement, in that order. For example (□, e, →) means that if the input symbol is □, an e is written to the tape and the read/write head moves right one cell.



(a) Four labels are missing from the diagram above.

Write the missing labels in the table below.

Number	Correct Label
❶	
❷	
❸	
❹	

(2)

(b) The Turing machine is carrying out a computation using one tape which is infinitely long in one direction. The machine starts in state S_1 with the string `01100` on the tape.
All other cells contain the blank symbol, □. The read/write head is positioned at the leftmost zero, as indicated by the arrow.

			0	1	1	0	0		
			↑						

Current State: S_1

Trace the computation of the Turing machine, using the transition function δ. Show the contents of the tape, the current position of the read/write head and the current state as the input symbols are processed.

Current State:

Current State:

Current State: ☐

Current State: ☐

Current State: ☐

Current State: ☐

(4)

(c) What is the purpose of the algorithm represented by this Turing machine?

(1)

(d) Explain the importance of the theory of Turing machines to the subject of computation.

(2)

(Total 9 marks)

Q20.

(a) A **state transition diagram** models the operation of a hotel lift. A program is written to simulate the behaviour of the lift in a hotel.

Describe **three** states that should be present in this diagram.

____

____

____

(3)

(b) **Figure 1** shows a state transition diagram for a problem, which has two states SI and S2.

Figure 1

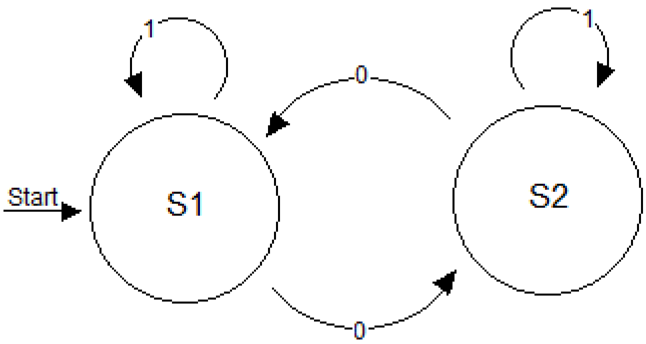


Table 1 is a state transition table for **Figure 1**. The **Next State** column is incomplete.

Table 1

Input	Current State	Next State
0	S1	S2
0	S2	(i)
1	S1	(ii)
1	S2	(iii)

(3)
(Total 6 marks)