**Q1.**

The finite state machine (FSM) represented as a state transition diagram recognises a language with an alphabet of 0, 1 and x.



Input strings of 0x and 1x are accepted by the FSM.

(a)     In the table indicate whether each input string is accepted or not accepted by the FSM in the diagram.

If an input string is accepted write YES.

If an input string is **not** accepted write NO.

Complete the table by filling in the unshaded cells.

|  |  |
| --- | --- |
| **Input string** | **Accepted by FSM?** |
| 111011x |   |
| 1110x |   |
| 111001x |   |

**(2)**

(b)     In words, describe the language (set of strings) that are accepted by the FSM in the diagram.

**(3)**

**(Total 5 marks)**

**Q2.**

A parking meter has an Add hours button (+), an Accept button, a coin slot, a payment card reader, a Cancel button and a number keypad.

The system operates in a specific sequence:

•   the system is initially in Idle Mode

•   when the user presses the + button the system goes into Select Hours Mode with the parking time set to 1 hour and the payment owed set to £1.00

○  each time the user presses the + button again, the number of hours’ parking time increases by 1 and the payment owed increases by £0.50

○  when the user presses the Accept button the system goes into Payment Due Mode and the user is able to make payments using cash or a payment card

○  the user can cancel the operation by pressing the Cancel button

•   using cash:

○  each time the user inserts a coin (except the final coin), the value of it is deducted from the payment owed

○  when the final coin that completes the payment is inserted, the system goes into Paid Mode

•   using a payment card:

○  when the user inserts a payment card into the card reader, the meter goes into a mode that allows the user to enter their PIN

○  the user then enters their PIN on the keypad

○  if the PIN is correct, the system goes into Paid Mode; otherwise the system goes into Idle Mode

•   the system remains in Paid Mode until the time paid for has elapsed.

The figure below shows a partially completed state transition diagram that represents the operation of the parking meter. Four of the states are labelled **(W)** to **(Z)** and events are labelled **(A)** to **(I)**.



Complete the table below by filling in the unshaded cells with the correct labels from the diagram above. You should write:

•   which labels **(A)** to **(I)** represent which event(s)

•   which labels **(W)** to **(Z)** represent which state.

Some of the cells in the table may need to be assigned more than one label.

Each label **must** only be used once.

|  |  |
| --- | --- |
| **Event / State** | **Label(s): (A) to (I), (W) to (Z)** |
| Card Payment Mode |   |
| Enter correct PIN |   |
| Enter incorrect PIN |   |
| Insert a coin (except final coin) |   |
| Insert final coin |   |
| Insert payment card |   |
| Paid Mode |   |
| Payment Due Mode |   |
| Press Accept |   |
| Press Cancel |   |
| Press + button |   |
| Select Hours Mode |   |

**(Total 6 marks)**

**Q3.**

A sentinel value is a special value **after** the end of a series of data values. It is a terminator for the series of data values but is not treated as part of the series. A sentinel value is used when you do not know how many data values are in the series.

The algorithm, represented using pseudo-code below, is an attempt at a method to add numbers that are input as a series terminated by the sentinel value -1

X ← 0

Result ← 0

WHILE X ≠ -1

  INPUT X

  Result ← Result + X

ENDWHILE

OUTPUT Result

(a)  Complete the table below by hand-tracing the algorithm shown above. You may not need to use all the rows in the table.

The first row of the table has already been completed for you.

The sequence of numbers for input is: 4, 6, 3, 2, -1

|  |  |  |
| --- | --- | --- |
| **X** | **Result** | **Output** |
| 0 | 0 | - |
|   |   |   |
|   |   |   |
|   |   |   |
|   |   |   |
|   |   |   |
|   |   |   |
|   |   |   |
|   |   |   |
|   |   |   |
|   |   |   |

**(3)**

(b)  Comment on the result of the trace and describe how the algorithm should be modified.

**(2)**

**(Total 5 marks)**

**Q4.**

The algorithm, represented using pseudo-code below, describes a method to rearrange three numbers in a data structure.

Numbers [0] ← 43

Numbers [1] ← 17

Numbers [2] ← 85

FOR X ← 1 TO 2

  MyValue ← Numbers[x]

  y ← x - 1

  WHILE (y > -1) AND (Numbers[y] < MyValue)

    Numbers [y + 1] ← Numbers[y]

    y ← y - 1

  ENDWHILE

  Numbers [y + 1] ← MyValue

ENDFOR

(a)  Complete the table below by hand-tracing the algorithm above. You may not need to use all the rows in the table.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **x** | **MyValue** | **y** | **y > -1 ? (True/False)** | **Numbers[y]** | **Numbers[y] < MyValue ? (True/False)** | **Numbers** |
| **[0]** | **[1]** | **[2]** |
|   |   |   |   |   |   | 43 | 17 | 85 |
|   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |

**(4)**

(b)  What type of rearrangement does this algorithm perform?

**(1)**

**(Total 5 marks)**

**Q5.**

ℝ denotes the set of real numbers, which includes the natural numbers, the rational numbers and the irrational numbers.

(a)     Give **one** example of a natural number.

**(1)**

(b)     Give **one** example of an irrational number.

**(1)**

**(Total 2 marks)**

**Q6.**

Convert the decimal number 6.34375 into an **unsigned fixed point binary number** using 8 bits with 5 bits after the binary point.

You may use the space below for working.

**(Total 2 marks)**

**Q7.**

To convert sound into a form which can be stored in a computer system, a device called an A-to-D converter is required.

(a)     What type of signal is sound?

**(1)**

(b)     What does an A-to-D converter do?

**(1)**

**(Total 2 marks)**