**Q1.**

Shade **one** lozenge to indicate to which category of system software a virus checker belongs.

|  |  |
| --- | --- |
| **Category** | **Shade one lozenge** |
| Operating systems |   |
| Translators |   |
| Utilities |   |

**(Total 1 mark)**

**Q2.**

The operating system is responsible for resource management.

Describe **two** different types of resource management that an operating system is responsible for.

Type 1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Type 2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(Total 2 marks)**

**Q3.**

A student has written a computer program using an imperative high-level programming language. The program could be translated using either a compiler or an interpreter.

Describe the steps that must be completed to translate and execute the program.

Your description should include:

•   why translation is necessary

•   the differences between how a compiler and an interpreter would translate the program

•   how the machine code instructions that are used to carry out the program will be fetched and executed by the processor from main memory.

**(Total 12 marks)**

**Q4.**

Identities are often applied to help simplify Boolean expressions. One such identity is:



Without using a truth table, explain why this identity is true.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(Total 2 marks)**

**Q5.**

Complete the truth tables for the OR and NAND gates.

|  |  |  |
| --- | --- | --- |
| **OR Gate** |   | **NAND Gate** |
| **Inputs** | **Output** |   | **Inputs** | **Output** |
| 0 | 0 |   |   | 0 | 0 |   |
| 0 | 1 |   |   | 0 | 1 |   |
| 1 | 0 |   |   | 1 | 0 |   |
| 1 | 1 |   |   | 1 | 1 |   |

**(Total 1 mark)**

**Q6.**

Using the rules of Boolean algebra, simplify the following Boolean expression.



You **must** show your working.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Answer \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(Total 4 marks)**

**Q7.**

State **one** role of the operating system, other than resource management.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(Total 1 mark)**

**Q8.**

The diagram shows a logic circuit.



(a)  Complete the part of the truth table for the circuit in the diagram that is shown below.

|  |  |  |
| --- | --- | --- |
| **Inputs** |   | **Outputs** |
| **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** |
| 0 | 0 | 0 |   |   |   |   |   |
| 0 | 0 | 1 |   |   |   |   |   |
| 0 | 1 | 1 |   |   |   |   |   |
| 1 | 1 | 1 |   |   |   |   |   |

**(3)**

(b)  Using the diagram above, write a Boolean expression to show how the output **G** is calculated from the inputs **A**, **B** and **C**.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

G = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(3)**

(c)  Explain the purpose of the circuit.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

**(Total 7 marks)**

**Q9.**

Employees at a bank use client computers to access data that is stored on a database server.

The database server uses software to query and modify data stored in a database on hard disk drives. It returns the results of these queries to the clients over the bank’s computer network.

The performance of the system is unsatisfactory: the time-delay between a client sending a query to the server and the client receiving the results is unacceptably long.

Explain how the performance of the system might be improved. You should consider the following factors that might be affecting the performance:

•   the hardware of the server

•   the design of the computer network

•   the database and software running on the server.

In your answer you will be assessed on your ability to follow a line of reasoning to produce a coherent, relevant and structured response.

**(Total 12 marks)**

**Q10.**

An operating system is designed to hide the complexities of the hardware from the user and to manage the hardware and other resources.

State **three** different types of management of either hardware or other resources that are performed by an operating system.

Type 1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Type 2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Type 3 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(Total 3 marks)**

**Q11.**

(a)     Complete the table below and draw the symbol for an AND gate in the box.

|  |  |  |
| --- | --- | --- |
| Truth table for an AND gate |   | AND gate symbol |
| **Input A** | **Input B** | **Output** |
|  |   |  |
|  |   |  |
|  |   |  |
|  |   |   |

**(2)**

(b)     Using the laws of Boolean algebra, simplify the following Boolean expression.

A.B. (A + B)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Answer \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(3)**

(c)     Using the laws of Boolean algebra, simplify the following Boolean expression.

(X + Y).(X + )

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Answer \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(3)**

**(Total 8 marks)**

**Q12.**

The tables below show two versions of the same segment of a program.

|  |  |  |
| --- | --- | --- |
| **Version A** |   | **Version B** |
| if x > 0:    y = y + 2else:    y = y - 1 |   | 00011100 0011000000101010 1001001011101010 0000001000101100 10010001 |

(a)     Shade in **one** lozenge to indicate which version, **A** or **B**, in the tables above represents object code.

|  |  |
| --- | --- |
| Version A  | Version B  |

**(1)**

(b)     Describe **two** differences between a compiler and an interpreter.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(2)**

(c)     Explain what intermediate code is **and** why some compilers will produce intermediate code as the final output.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(2)**

**(Total 5 marks)**

**Q13.**

A burglar alarm system is to be implemented that has the following sensors:

•        a door sensor **D** that outputs TRUE when the door is open and FALSE when the door is shut

•        a pressure mat sensor **M** that outputs TRUE while a weight is detected on it and FALSE when no weight is detected on it.

The alarm also has a key **K** that turns the alarm on and off. **K** outputs a TRUE signal when the alarm is switched on and FALSE when the alarm is off.

The alarm output **A** sounds a bell. It should be TRUE if:

•        the alarm is on AND

•        either of the sensors **D** or **M** are set to the value TRUE.

(a)     Draw a logic circuit that will behave as described above for the inputs **D**, **M** and **K** and the output **A**.



**(2)**

(b)     Write a Boolean expression to represent the logic of this alarm system.

**A=**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(2)**

(c)     In this alarm system, the alarm bell will sound only while the door is open or a weight is placed on the pressure mat. If someone who has stepped on to the mat moves off it, or an open door is closed, the alarm bell will stop ringing.

A D-type flip-flop could be incorporated into the logic circuit so that the alarm bell would continue to sound after a person closed the door or moved off the pressure mat.

Explain how this could be achieved. In your answer refer to:

•        why a D-type flip-flop would be suitable for this task

•        where the D-type flip-flop would need to be inserted into the circuit

•        what additional input the D-type flip-flop would need.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(3)**

**(Total 7 marks)**

**Q14.**

(a)     Complete the truth table below for a NAND gate.

|  |
| --- |
|      **NAND gate** |
| **Input A** | **Input B** | **Output** |
| 0 | 0 |   |
| 0 | 1 |   |
| 1 | 0 |   |
| 1 | 1 |   |

**(1)**

(b)     Multiplexors are used in electronic switching.

A 2-to-1 multiplexor has a Boolean equation where A and B are two inputs, S is the selector input, and Q is the output.



(i)     Complete the truth table for the above Boolean equation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S** | **A** | **B** |  |  | **B.S** | **Q** |
| 0 | 0 | 0 |   |   |   |   |
| 0 | 0 | 1 |   |   |   |   |
| 0 | 1 | 0 |   |   |   |   |
| 0 | 1 | 1 |   |   |   |   |
| 1 | 0 | 0 |   |   |   |   |
| 1 | 0 | 1 |   |   |   |   |
| 1 | 1 | 0 |   |   |   |   |
| 1 | 1 | 1 |   |   |   |   |

**(3)**

(ii)     Draw a circuit for the Boolean equation in the rectangle below.



**(4)**

(iii)     By considering its inputs and outputs, describe what the 2-to-1 multiplexor circuit does.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

**(Total 9 marks)**

**Q15.**

Explain the role of the operating system and state the important tasks which are carried out by most operating systems.

Beyond those of a standard operating system, describe the additional operational characteristics that you would expect a real time operating system to have.

In your answer you will be assessed on your ability to use good English, and to organise your answer clearly in complete sentences, using specialist vocabulary where appropriate.

**(Total 7 marks)**

**Q16.**

The diagram below shows the classifications of various types of software used on a computer system and some examples of these types.



(a)     Complete the diagram by suggesting labels for the boxes numbered **1** to **4** in the diagram.

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(4)**

(b)     (i)      Machine code is the first generation of programming language.

What is the second generation of programming language?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(ii)      A program written in a second generation programming language has been loaded into a computer. In this form it cannot be directly executed on this computer.

What has to be done to make an executable form of the program, which can be directly executed by this computer, and what would be used, typically, to do this?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(2)**

(iii)     A programmer then finds that when the executable form of the program is transferred unaltered to another computer, the program does not run and an error message is displayed.

Why might the executable form of the program not be able to run on this computer?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

**(Total 8 marks)**

**Q17.**

(a)     Complete the truth tables for the following logic gates.

OR gate

|  |  |  |
| --- | --- | --- |
| **Input A** | **Input B** | **Output** |
| 0 | 0 |   |
| 0 | 1 |   |
| 1 | 0 |   |
| 1 | 1 |   |

NAND gate

|  |  |  |
| --- | --- | --- |
| **Input A** | **Input B** | **Output** |
| 0 | 0 |   |
| 0 | 1 |   |
| 1 | 0 |   |
| 1 | 1 |   |

**(2)**

(b)     Represent the following Boolean equation as a logic circuit by completing the diagram below.





**(5)**

(c)     Simplify the following expression.



Show each stage of your working.

**(2)**

Final answer \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

**(Total 10 marks)**

**Q18.**

For each problem in the table below, place a tick in the appropriate box to indicate the generation of programming language best suited to developing a solution to the problem.

Do **not** tick more than one box in each row.

|  |  |
| --- | --- |
|   |       **Generation** |
| **Problem** | **1st** | **2nd** | **4th** |
| Developing a diagnosis program for medical symptoms |   |   |   |
| Developing a program for an embedded microprocessor for a washing machine |   |   |   |

**(Total 2 marks)**

**Q19.**

A school has recently launched a ‘Parent Portal’ which is a website that provides information from the school. By logging on to the portal a parent can access the information that is stored about their son or daughter. This information includes academic reports, discipline records and other personal data.

(a)     A parent recently contacted the school because he was concerned that when he logged on to read his daughter’s report he could access the reports of all the other students.

The school should immediately look into this concern as a law has been broken.
State the **full name** of the law that has been broken.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(b)     Which principle of the law identified in your answer to part (a) has been broken?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(c)     State another principle of the law identified in your answer to part (a).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(d)     A parent also noted that the website was using HTTP (HyperText Transfer Protocol).

Why should the school be concerned about the use of this protocol and which protocol would you recommend that the school should use instead?

Why concerned \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Suggested protocol \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(2)**

(e)     The process of writing reports and then allowing access to these reports via the parent portal involves the use of many different categories of software.

Below is a list of different categories of software:

Operating system, Utility program, Special purpose application software, Bespoke application software, General purpose application software

Complete **Table 1** by writing the correct category from the list above in the **Category** column next to the appropriate **Software**.

You should **not** use a category more than once.

**Table 1**

|  |  |
| --- | --- |
| **Software** | **Category** |
| Word processor used to write the pupil reports |   |
| The parent portal web application which was programmed for this school |   |
| The web server software run by the school |   |

**(3)**

**(Total 8 marks)**

**Q20.**

When writing a program, a programmer could use an assembly language, a high level imperative language or a high level declarative language.

Outline the major differences in each of these **three** approaches. For each language type, your answer could include:

•        advantages and disadvantages compared to other language types

•        how the programmer would express their programs

•        what translation software could be used, if applicable

•        a situation where it might be the most appropriate choice.

In your answer you will be assessed on your ability to use good English and to organise your answer clearly in complete sentences, using specialist vocabulary where appropriate.

**(Total 8 marks)**

**Q21.**

(a)     Represent the Boolean equation Q =   as a logic circuit by drawing a diagram using **only** the following symbols:



|  |  |
| --- | --- |
| AND | NOT |



**(3)**

(b)     Use the following truth tables to demonstrate that A + B = 

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **A + B** |   | **A** | **B** |  |  |  |  |
| 0 | 0 |   |   | 0 | 0 |   |   |   |   |
| 0 | 1 |   |   | 0 | 1 |   |   |   |   |
| 1 | 0 |   |   | 1 | 0 |   |   |   |   |
| 1 | 1 |   |   | 1 | 1 |   |   |   |   |

**(4)**

(c)     What is the name commonly associated with the statement A + B =   ?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(d)     Simplify the Boolean expression below.

A.B. + A.

Show each stage of your workin.

**(2)**

Final answer \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

**(Total 11 marks)**

**Q22.**

There are many third generation programming languages available. Some of these can be classified as imperative high level languages.

(a)     Explain what is meant by the term *imperative high level language*?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(2)**

(b)     Give **one** reason for there being so many third generation programming languages.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

**(Total 3 marks)**

**Q23.**

(a)     Complete the truth tables for the following logic gates.

|  |  |  |
| --- | --- | --- |
| **AND Gate** |  | **XOR Gate** |
| **Input X** | **Input Y** | **Input Q** |  | **Output X** | **Input Y** | **Output Q** |
| 0 | 0 |   |   | 0 | 0 |   |
| 0 | 1 |   |   | 0 | 1 |   |
| 1 | 0 |   |   | 1 | 0 |   |
| 1 | 1 |   |   | 1 | 1 |   |

**(2)**

(b)     A line-following robot has three sensors. It moves along a black line on a white background whilst the following conditions are met:

•        the ultrasonic sensor U does not detect any obstacle

•        either, but not both, of the infrared sensors L and R are on the black line.

Sensor U returns 1 if it detects an obstacle and 0 if the path is clear.
Sensors L and R each return 1 if they detect black and 0 if they detect white.

A logic circuit will process the input from the sensors and produce an output M.

M should be 1 if the robot is to move and 0 if the robot should stop.

(i)      Represent the output M as a Boolean expression.

M = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(3)**

(ii)     The following symbols are used to represent logic gates:

|  |
| --- |
|  |
|       NOT                          OR                            AND |

|  |
| --- |
|  |
|       XOR                        NOR                        NAND  |

Using a combination of any of the above logic gates draw a logic circuit for this system in the box below. You will **not** need to use all of the different types of logic gates.



**(3)**

(c)     Apply De Morgan’s Law(s) to the following expression and simplify the result.



Show the stages of your working.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(2)**

Final answer \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

**(Total 11 marks)**

**Q24.**

**Figure 1** and **Figure 2** show different versions of the same program.

|  |  |
| --- | --- |
| **Figure 1** | **Figure 2** |
| **(x)               (y)                 (z)** | **(x)                (y)                        (z)** |
| 200     LOAD      7201     ADD       3202     ADD       6203     STORE     255 | 200     01010110     00000111201     11010000     00000011202     11010000     00000110203     11110000     11111111 |

(a)     What generation of programming language is shown in **Figure 1**?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(b)     In both figures above there is a column labelled **(x)**.

What would be a suitable heading for this column?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(c)     In both tables the instruction is split into two parts.

What are the names of the instruction parts in columns **(y)** and **(z)**?

**(y)** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(z)** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(2)**

(d)     What is the relationship between the instructions in **Figure 1** and **Figure 2**?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

**(Total 5 marks)**

**Q25.**

(a)     One type of software can be described by the phrase "*performs tasks needed to operate the hardware*".

What type of software is being described?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(b)     (i)      Explain what is meant by *general purpose application software.*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(ii)     Give **one** example of general purpose application software.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(c)     An IT manager needs to buy software to manage stock control.

(i)      Why might the manager choose a special purpose application package rather than a bespoke solution? Give **two** reasons.

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(2)**

(ii)     Although special purpose application packages for stock control are available, explain why the IT manager might choose to order a bespoke piece of software.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

**(Total 6 marks)**

**Q26.**

Write the following Boolean expressions in their simplest forms.

|  |  |
| --- | --- |
| (a) |   |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

|  |  |
| --- | --- |
| (b) |   |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

|  |  |
| --- | --- |
| (c) |   |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(d)    A . (B+1)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

**(Total 4 marks)**

**Q27.**

What are machines good and bad at, in comparison to humans?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(Total 4 marks)**

**Q28.**

The diagram below shows the processor registers and busses that are used during the fetch part of the fetch-execute cycle, together with the main memory. The values stored in memory locations 0 to 6 in the main memory are machine code instructions.



(a)     Name the components that are labelled with the numbers 1 to 4. In the case of register names, the full names must be stated.

|  |  |
| --- | --- |
| **Number** | **Component Name** |
|  |   |
|  |   |
|  |   |
|  |   |

**(4)**

(b)     Explain what happens during the decode and execute stages of the fetch-execute cycle.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(3)**

(c)     The machine code instructions in the main memory in the diagram above are shown in binary.
When programmers look at machine code instructions they usually prefer to view them in hexadecimal.

State **one** reason why this is the case.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(d)     The machine code instructions in the main memory in the diagram above were produced when an assembly language program was translated into machine code.

(i)      What type of program translator was used to do this?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(ii)     Most computer programs are initially written in an imperative high level language rather than assembly language.

Explain why this is the case.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(3)**

**(Total 12 marks)**

**Q29.**

(a)     Complete the truth tables for the following logic gates.

|  |  |  |
| --- | --- | --- |
| **OR Gate** |   | **XOR Gate** |
| **Input A** | **Input B** | **Output Q** |  | **Input A** | **Input B** | **Output Q** |
| 0 | 0 |   |   | 0 | 0 |   |
| 0 | 1 |   |   | 0 | 1 |   |
| 1 | 0 |   |   | 1 | 0 |   |
| 1 | 1 |   |   | 1 | 1 |   |

**(2)**

(b)     Represent the Boolean equation Q = A + B·. as a logic circuit by drawing a diagram of it.



**(3)**

(c)     Simplify the Boolean expression:

B·(A + )

Show your working.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(3)**

**(Total 8 marks)**

**Q30.**

Simplify the Boolean expression:



Show your working.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(Total 3 marks)**

**Q31.**

(a)     Some of the basic components of a computer system are processor, main memory, and secondary storage.

(i)      What connects the processor and main memory?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(ii)     What is the purpose of secondary storage?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(iii)     Describe what happens during the fetch-execute cycle.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(2)**

(b)     (i)      Machine code is the first generation programming language. What is the second generation?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(ii)     A programmer writes a program in a second generation programming language.

What has to be done to this program before it can be executed?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(2)**

(iii)     Some high level languages are classified as *imperative*. What is meant by imperative?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(iv)    Give an example of an imperative high level language.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(v)     What is the relationship between an imperative high level language statement and its machine code equivalent?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(vi)    Give **two** disadvantages of programming in first and second generation programming languages compared with imperative high level languages.

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(2)**

**(Total 12 marks)**

**Q32.**

(a)     Two classifications of software are System Software and Application Software.

What is meant by:

(i)      System Software; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(ii)     Application Software? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(b)     Give an example of:

(i)      System Software; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

(ii)     Application Software? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1)**

**(Total 4 marks)**

Mark schemes

**Q1.**

**Mark is AO1 (knowledge)**

**1 mark:** Lozenge for “Utilities” shaded.

**R.** if more than one lozenge shaded

**[1]**

**Q2.**

**All marks AO1 (knowledge)**

Allocate processors/cores to processes // schedule processes // decide which process to carry out when;

Allocate memory/RAM to processes // moving data into and out of RAM / to a paging file for virtual memory // ensuring processes can only write to memory that they have been allocated;

Allocate I/O devices to processes // manages communication between processes and I/O devices // automatic installation of drivers for new I/O devices; **A.** examples of devices (but no more than one mark) **NE.** manages I/O devices

Allocate space on a storage device to files // organising files into directories // determines where on a device to save a file // recognising storage devices when they are connected; **A.** defragmentation of disks **NE.** saving a file

Installation of new software // automatic/managing updating of software;

**A.** “programs” or “tasks” for “processes”

**R.** handling interrupts

**R.** hides complexity

**Note:** Students must describe the type of resource management – phrases such as “processor management”, “allocating memory” etc are not enough.

**Max 2**

**[2]**

**Q3.**

**All marks AO1 (understanding)**

**Level of response question**

|  |  |  |
| --- | --- | --- |
| **Level** | **Description** | **Mark Range** |
| 4 | A line of reasoning has been followed to produce a coherent, relevant, substantiated and logically structured response. The response covers all three areas indicated in the guidance below and in at least two of these areas there is sufficient detail to show that the student has a good level of understanding. To reach the top of this mark range, a good level of understanding must be shown of all three areas. | 10-12 |
| 3 | A line of reasoning has been followed to produce a coherent, relevant, substantiated and logically structured response which shows a good level of understanding of at least two areas indicated in the guidance below. | 7-9 |
| 2 | A limited attempt has been made to follow a line of reasoning and the response has a mostly logical structure. Either a good level of understanding of one area from the guidance has been shown or a limited understanding of two areas. | 4-6 |
| 1 | A few relevant points have been made but there is no evidence that a line of reasoning has been followed. The points may only relate to one or two of the areas from the guidance or may be made in a superficial way with little substantiation. | 1-3 |

**Guidance – Indicative Response**

**1.   Why translation is necessary**

Processor can only execute machine code instructions **A.** computer for processor

High-level instructions cannot be executed directly // high-level instructions are not machine code **A.** must be converted to machine code to be executed

**NE.** “Understand” for “execute”.

*Good level of understanding = at least one point made*

**2.   Differences between compilation and interpreting**

Compiler analyses program as a whole

Interpreter analyses program on a line-by-line basis

Compiler produces object code/executable file/machine code/bytecode

Interpreter calls subroutines within its own code to carry out commands

Compiler will not translate any of the program if it encounters an error

Interpreter translates/executes program until first error is encountered

If (unchanged) program executed twice/multiple times, compiler will only need to translate it once

Interpreter translates a program each time it is executed

Interpreter executes each line immediately after translating it

If student has written about compiler outputting bytecode then: bytecode will later be interpreted // executed by a virtual machine // just-in-time-compiled

Once translated, compiled code does not need the compiler to be present to run

An interpreter must always be present for a program that is interpreted to run

Once compiled, code will only run on one type of processor / virtual machine

Interpreter could translate the same instruction multiple times (eg if it is in a loop)

Good level of understanding = at least four points made

**3.   How machine code instructions fetched and executed**

F-E Stage 1 Fetch:

Contents of Program Counter / PC transferred to Memory Address Register / MAR

**R.** if implied the instruction is stored in the PC

Address bus used to transfer this address to main memory

Transfer of main memory content uses the data bus

Contents of addressed memory location loaded into the Memory Buffer Register / MBR

Increment (contents of) Program Counter / PC **A.** at any part of fetch process after transferring PC to MAR

Increment Program Counter / PC and fetch instruction simultaneously

Contents of MBR copied to CIR

F-E Stage 2 Decode:

Decode instruction held by the (Current) Instruction Register / (C)IR

The control unit decodes the instruction

Instruction split into opcode and operand

F-E Stage 3 Execute:

If necessary, data is fetched

If necessary, data is stored in memory

The opcode identifies the type of operation/instruction to be performed (by the processor)

Result (may be) stored in register/accumulator

The operation (identified by the opcode) is performed by the processor. **A.** ALU

Status register updated

If jump / branch required Program Counter/PC is updated

General:

Instructions will be for program (if compiled) or interpreter (if interpreted)

*Good level of understanding = at least five points made and at least two of the three stages of the F-E cycle are covered.*

**[12]**

**Q4.**

**All marks AO1 (understanding)**

If input A is 0 then NOT A will be 1 and if A is 1 then NOT A will be 0 // one of the inputs to the AND operator will always be 0 // the inputs can only be 0,1 or 1,0;

**NE.** if only expressed one way around eg if A is 0 then NOT A is 1

**NE.** NOT A is always the opposite of A unless clarified that possible values are 0/1

**NE.** if only presented as a truth table

**A.** on/off, true/false for 1/0

An AND gate only outputs 1 if both inputs are 1 // an AND gate always outputs 0 if one of its inputs is 0 // when inputs to AND are 1 and 0 then output is 0;

**[2]**

**Q5.**

**Mark is AO1 (knowledge)**

|  |  |  |
| --- | --- | --- |
| **OR Gate** |  | **NAND Gate** |
|  |  |  |

**1 mark**: All values in both **Output** columns are correctly completed.

**[1]**

**Q6.**

**All marks AO2 (apply)**

**Marking guidance for examiners**

•   Award marks for working out until an incorrect step has been made.

•   If, in any one step, a candidate is simplifying different parts of an expression simultaneously award all relevant marks for this multiple stage but don’t award any further marks for working in any parts simplified incorrectly. For example, if the expression P.P.(P+Q) + P.P.1 was changed to P.(P+Q)+P.0, the candidate would get one mark for simplifying the first part to P.(P+Q) and could get further marks for correctly simplifying this part of the expression further but should not be awarded marks for simplifying the incorrectly changed part P.0 (ie to 0)

**1 mark** for final answer: A + B

**Max 3** for working. Award up to two marks for applying each one of the three techniques (one mark per application):

•   a successful application of De Morgan’s Law (and any associated cancellation of NOTs) that produces a simpler expression.

•   applying an identity other than cancelling NOTs that produces a simpler expression.

•   successfully expanding brackets.

**Note**: A simpler expression is one that is logically equivalent to the original expression but uses fewer logical operators.

**Max 3** if correct final answer but any incorrect working

**Example Solution (1)**

****

**Example Solution (2)**

****

**Example Solution (3)**

****

**[4]**

**Q7.**

**Mark is for AO1 (knowledge)**

The role of the operating system is to hide the complexities of the hardware from the user;

**A.** other reasonable answers that are not resource management

**[1]**

**Q8.**

(a)  **All marks AO2 (apply)**

|  |  |  |
| --- | --- | --- |
| **Input** |  | **Outputs** |
| **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |

**1 mark:** Column D correct or column E or column F correct

**1 mark:** Column G correct

**1 mark:** Column H correct

**Max 2** if any incorrect values in table

**3**

(b)  **All marks AO2 (apply)**

B⋅C+A⋅(B⨁C)

**1 mark:** B⋅C or B⨁C somewhere in expression

**1 mark:** A is ANDed with B⨁C

**1 mark:** Fully correct expression

**A.** award second mark even if brackets around B⊕C are missing

**A.** use of AND, OR, XOR instead of symbols

**A.** (B̅⋅C)+(B⋅C̅) for (B⨁C)

**If a student has written an expression but then tried to simplify it and made an error then mark the original expression that the student has written down and ignore the simplification.**

**3**

(c)  **Mark is for AO2 (analyse)**

It adds together its inputs // it is a full adder circuit;

**N.E.** half-adder, adder

**1**

**[7]**

**Q9.**

**All marks AO1 (understanding)**

|  |  |  |
| --- | --- | --- |
| **Level** | **Description** | **Mark Range** |
| 4 | A line of reasoning has been followed to produce a coherent, relevant, substantiated and logically structured response. The response covers all three areas indicated in the guidance below and in at least two of these areas there is sufficient detail to show that the student has a good level of understanding. To reach the top of this mark range, a good level of understanding must be shown of all three areas. | 10-12 |
| 3 | A line of reasoning has been followed to produce a coherent, relevant, substantiated and logically structured response which shows a good level of understanding of at least two areas indicated in the guidance below. | 7-9 |
| 2 | A limited attempt has been made to follow a line of reasoning and the response has a mostly logical structure. At least four points have been made. Either a good level of understanding of one area from the guidance has been shown or a limited understanding of two areas. | 4-6 |
| 1 | A few relevant points have been made but there is no evidence that a line of reasoning has been followed. The points may only relate to one or two of the areas from the guidance or may be made in a superficial way with little substantiation. | 1-3 |

**Guidance – Indicative Response**

**For each guidance point, if the student expands on the point to explain in what way the measure will improve performance then this can be considered to be a second point.** For example:

•   “Using a processor with more cores” is one point.

•   “Using a processor with more cores which will be able to execute multiple instructions simultaneously” is two points.

Note that just “faster” is not enough to count as an expansion point without an explanation of why.

**1. Server Hardware**

Replace the processor with one which has more cores

Replace the processor with one which has more cache memory // increase the amount of cache memory

Replace the processor with one which runs at a faster clock speed **NE.** faster processor

Use a parallel processor architecture // use more processors which can work in parallel

Use a processor with a bigger word size

Use a processor that makes (better) use of pipelining

Install more RAM // main memory // primary memory

Use RAM // main memory // primary memory with a faster access time

Replace HDDs with SSDs // Replace HDDS with HDDs that can read data at a faster rate

Defragment the HDD

Replace the motherboard with one which has buses which run at a faster clock speed

Replace the motherboard with one which has more lines in the data bus

Use the Harvard architecture

Distribute the processing across multiple servers

**2. Network**

Replace the network cable with cable that has a higher bandwidth // replace copper cable with fibre-optic cable **A.** Ethernet cable for fibre-optic NE. higher bandwidth network

Replace any wireless / WiFi connections with wired ones

Replace the network cards with ones that can transmit data at a higher bitrate

Consider the overall network design eg how the network is divided into subnets **A.** split the network into subnets

Use a star topology (instead of a bus)

Consider using a more efficient protocol for the data across the network

Add additional wireless access points

**3. Database and Software**

Use a more efficient technique for controlling concurrent access to the database // replace record/table locks with serialisation/timestamp ordering/commitment ordering

Replace the database software with software that uses more efficient algorithms for tasks **A.** examples eg replace linear search with binary search

Use the index feature of the database to speed up searching on fields that are commonly used for this purpose

Rewrite the database software in a language that is suitable for concurrent execution // use a functional programming language for the database software

Ensure the software is compiled rather than executed by an interpreter // rewrite the software in assembly language/machine code

Review the conceptual model of the database to see if it contains any inefficiencies such as data redundancy that could be eliminated **A**. normalise the database design

Consider if it would be appropriate to sacrifice normalisation of the conceptual model to improve performance

Use a non-relational database system **A.** examples eg NoSQL

Distribute the data across multiple servers

Try to reduce the amount of other (unrelated) software that might be running on the database server at the same time

Try to reduce the number of database accesses that need to be made simultaneously // run some tasks at quiet times / overnight

Purge / archive data that is no longer necessary / in use

**[12]**

**Q10.**

Processor management // Allocation of processors // Allocation of processor time // (process) scheduling // thread management;

**A.** Processing management, CPU management

Allocation/management of RAM / memory // allocation of buffers;

Allocation/management of / control of I/O devices/peripherals //

I/O management // device driver management;

File / backing store / secondary store management / access / organisation;

Power / battery management;

Interrupt handling;

**A.** Provision of Application Program Interface / API

**A.** interface between hardware and applications

**A.** Provision / management of (windows in) user interface

**A.** Management of system security

**A.** Answers by example, only one example of each type

**A.** A description of a type of software management but not just "software management". e.g. loading of programs, software installation, registering DLLs.

**A.** Managing network connections but **R.** Network management

**R.** Software management alone unless role of OS in this is clear e.g. installation of new software, updating registry

**MAX 3**

**[3]**

**Q11.**

(a)     **Marks are for AO1 (knowledge)**

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **Q** |  |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**1 mark:** Table completed correctly;

**1 mark:** AND gate symbol drawn;

**2**

(b)     **Marks are for AO2 (apply)**

A.B.(A + B)

A.B.A + A.B.B ; [expansion of brackets]

B.A + A.B ;        [use of A.A = A]

A.B ;                  [use of A + A = A]

**1 mark:** Final answer: A.B;

**Max 2 for working**

**3**

(c)     (**Marks are for AO2 (apply)**

X + Y).(X + NOT Y)

XX + X(NOT Y) + XY + Y(NOT Y) ; [expansion of brackets]

X + X(NOT Y) + XY ; [use of X.X = X or use of Y(NOT Y) = 0 ]

X ( 1 + NOT Y + Y ) ; [use of 1 + X = 1]

**1 mark:** Final answer - X;

**Max 2 for working**

**3**

**[8]**

**Q12.**

(a)     **Mark is for AO1 (understanding)**

Version: B;

**1**

(b)     **Marks are for AO1 (understanding)**

A compiler produces object code whilst an interpreter does not;

A compiler translates the whole of the source code into object code whilst an interpreter translates line by line;

The object code produced by a compiler will execute faster, (once it is compiled) than interpreting the source code (every time the program is run)

An interpreter can run (syntactically correct) parts of a program whilst there are syntax errors in other parts of it, which a compiler cannot;

**Max 2**

**2**

(c)     **Marks are for AO1 (understanding)**

Intermediate code is not (directly) executable / /

Intermediate code will by run / interpreted by a virtual machine / / Compiled into an executable just before running / just in time;

Intermediate code can be run on different computing platforms / / One solution can be targeted at multiple platforms;

**Max 2**

**2**

**[5]**

**Q13.**

(a)     **All marks AO2 (apply)**

****

**1 mark:** inputs D and M connected to an OR gate;

**1 mark:** inputs K and output of OR gate connected to AND gate plus output connected to A;

**A** a logically equivalent circuit

**2**

(b)     **All marks AO2 (apply)**

A = (D + M) · K

**1 mark:** D + M somewhere in expression, even if full expression incorrect

**1 mark:** fully correct expression

**A** A logically equivalent expression

**2**

(c)     **1 mark for AO1 (understanding), 1 mark for AO2 (application) and 1 mark for AO1 (knowledge)**

**AO1 (understanding):1 mark:** Flip-flop will store the state of its input / / Flip-flop acts as memory;

**AO2 (application):1 mark:** Insert into circuit between the output of the OR gate and the AND gate / / after the AND gate;

**AO1 (knowledge):1 mark:** Clock signal / / trigger / / signal to indicate when the value (of the input) should be stored / read;

**3**

**[7]**

**Q14.**

(a)

|  |  |  |
| --- | --- | --- |
| **Input A** | **Input B** | **Output** |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

One mark for having correct values in Output column;

**1**

(b)     (i)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S** | **A** | **B** |  | **A.** | **B.S** | **Q** |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 |

**Marking:**One mark for the **A.** column being correct;
One mark for the **B.S** column being correct;
The final **Q** column should follow through from the previous two columns as an OR statement;

**3**

(ii)



One mark for NOT gate with input from S;
**A** just a circle on AND gate input from S

One mark for AND gate with input from NOT S and A;
**A** if no NOT gate from S

One mark for AND gate with input from B and S;

One mark for output from AND gates going into OR gate with output connect to Q;

**4**

(iii)    A multiplexor selects one of several input lines / wires and forwards / duplicates the Boolean value on this one line onto a single line / wire;

If S is 1 then input B is output otherwise input A is output // if S is 0 then input A is output otherwise input B is output;

Note:
1 can be mapped to on / true / high
0 can be mapped to off / false / low

**MAX 1**

**[9]**

**Q15.**

**SUBJECT MARKING POINTS:**

**Role of the operating system (1 point):**

To hide the complexities of the hardware from the user // provision of virtual machine;

**Tasks carried out by most operating systems (MAX 3 points):**

Processor scheduling // allocation of processors // process scheduling;
**A** processor management
Thread management;
Context switching;
Inter-process communication;
Memory management // allocation of memory / RAM;
Virtual memory provision // a brief explanation of what virtual memory is;
I / O management // allocation of I / O devices // management of / communication with (device) driver software ;
**A** examples of devices
File management // organisation of files on storage devices // loading and saving of files;
**A** examples of devices
Interrupt handling;
Power / battery management;
Provision of a user interface // allows user to interact with computer;
Provision of an Application Programming Interface / API (so that application software can call operating system routines);
**A** management of system security
**A** managing communications over a network
**A** just names of tasks e.g. "memory management"
**A** “storage management” for one of “memory management” or “file management”

**Additional functionality of a real time operating system (MAX 3 points):**

Must be able to deal with many events occurring simultaneously;
Must be able to deal with events that occur at unpredictable times;
Must produce output / perform processing within a specified / predictable / known time interval // Must produce output quickly enough to affect (the source of) the inputs;
**A** quickly enough for task, in a timely manner
**NE** quickly, instantly, in a reasonable time
Must be fail-safe;
Must be able to quickly switch between threads / processes // quickly allocate memory // quickly handle interrupts;
Must support non-sequential application programs;

**HOW TO AWARD MARKS:**

**Mark Bands and Description**

*To achieve a mark in this band, candidates must meet the subject criterion (SUB) and 4 of the 5 quality of written communication criteria (QWCx).*

SUB           Candidate has made seven mark-worthy points and successfully covers all of the three topic areas (role, tasks, real time).

QWC1       Text is legible.

QWC2       There are few, if any, errors of spelling, punctuation and grammar. Meaning is clear.

QWC3       The candidate has selected and used a form and style of writing appropriate to the purpose and has expressed ideas clearly and fluently.

QWC4       Sentences (and paragraphs) follow on from one another clearly and coherently.

QWC5       Appropriate specialist vocabulary has been used.

**7**

*To achieve a mark in this band, candidates must meet the subject criterion (SUB) and all 5 of the quality of written communication criteria (QWCx).*

SUB           Candidate has made at least four mark-worthy points and successfully covers at least two of the three topic areas (role, tasks, real time).

QWC1       Text is legible.

QWC2       There may be occasional errors of spelling, punctuation and grammar. Meaning is clear.

QWC3       The candidate has, in the main, used a form and style of writing appropriate to the purpose, with occasional lapses. The candidate has expressed ideas clearly and reasonably fluently.

QWC4       The candidate has used well-linked sentences (and paragraphs).

QWC5       Appropriate specialist vocabulary has been used.

**4-6**

*To achieve a mark in this band, candidates must meet the subject criterion (SUB) and 4 of the 5 quality of written communication criteria (QWCx).*

SUB           Candidate has made a small number of relevant points but only successfully covers one or two of the three topic areas (role, tasks, real time).

QWC1       Most of the text is legible.

QWC2       There may be some errors of spelling, punctuation and grammar but it should still be possible to understand most of the response.

QWC3       The candidate has used a form and style of writing which has many deficiencies. Ideas are not always clearly expressed.

QWC4       Sentences (and paragraphs) may not always be well-connected.

QWC5       Specialist vocabulary has been used inappropriately or not at all.

**1-3**

Candidate has made no relevant points

**0**

Note: Even if English is perfect, candidates can only get marks for the points made at the top of the mark scheme for this question.

If a candidate meets the subject criterion in a band but does not meet the quality of written communication criteria then drop mark by one band, providing that at least 4 of the quality of language criteria are met in the lower band. If 4 criteria are not met then drop by two bands.

**[7]**

**Q16.**

(a)     1         special purpose (application software);

  **A** specific purpose

  **R** special (software) / specialist (software)

2         word processor / / spreadsheet / /

  presentation software / program / / database;

  **A** any other sensible answer

  **R** (web) browser

  **R** text editor

3         translator (software / program);

  **A** translating / translation

4        utility (software / program);

**R** just trade name of a specific piece of software unless used as an example (ie MS Word)

**4**

(b)     (i)      assembly (language);

**A** assembly code

**R** assembler

**1**

(ii)     has to be translated into machine code / / each assembly language instruction will be translated into machine code equivalent;

by an assembler;

**A** converted for translated

**A** first generation for machine code

**2**

(iii)    Because it does not have the same processor (type) / / the instruction set is different / / different architecture / platform;

(Assembled / linked for a) different operating system;

**NE** operating software

The program refers to a memory address that does not exist on this computer / / relocatable code used but addressing system on new machine different;

not enough memory space;

required peripherals are not available;

required library (code / program) missing;

**MAX 1**

**[8]**

**Q17.**

(a)     OR gate

|  |  |  |
| --- | --- | --- |
| **Input A** | **Input B** | **Output** |
| 0 | 0 | **0** |
| 0 | 1 | **1** |
| 1 | 0 | **1** |
| 1 | 1 | **1** |

NAND gate

|  |  |  |
| --- | --- | --- |
| **Input A** | **Input B** | **Output** |
| 0 | 0 | **1** |
| 0 | 1 | **1** |
| 1 | 0 | **1** |
| 1 | 1 | **0** |

**1 mark** for correct output OR gate;

**1 mark** for correct output NAND gate;

**2**

(b)


**1 mark** for inputs A and B connected to AND gate;

**1 mark** for inputs B and C connected to AND gate;

**1 mark** for output of AND (A,B input) as only connection going to NOT gate;

**1 mark** for output of NOT gate plus the AND gate (B,C input) going to OR gate;

**1 mark** OR gate as only connection going to NOT gate and output only connection to Q;

**5**

(c)     **MAX 2 if working out is not logically sound**

**Example 1**:





Having applied De Morgan’s correctly;



Having factorised;

Final answer: **B** ;

**Example 2**:





Having applied De Morgan’s correctly;



Expanded bracket;



Simplified elements



Having used C + C.D = C to simplify



Having used C + C.D = C to simplify again

**Final answer: B ;**

***Truth Table Answer***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** |  |  |  |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 |
|   |   | **X** | **Y** | **Z** |

**1 mark** for both columns marked X and Y above; (column X could be labelled **A.B** )

**1 mark** for final column Z;

**1 mark** for final answer: B;

**3**

**[10]**

**Q18.**

First row tick in '4th' column;
Second row tick in '2nd' column;

**A** Other symbols instead of ticks
**R** Responses where more than one box is ticked on the same line

**[2]**

**Q19.**

(a)     Data Protection (Act);

**1**

(b)     Data should be kept securely;

**1**

(c)     Data should be fairly and lawfully processed;
Data should be obtained for specified and lawful purposes. (**A** Data should be processed for limited purposes);
Data should be adequate, relevant and not excessive;
Data should be accurate // kept up to date;
Data should not be kept longer than necessary;
Data should be not transferred to other countries without adequate protection;
Data should be processed in accordance with the rights of the data subjects.

**Max 1**

(d)     That data is not being encrypted // data is not being sent securely // that hackers might be able to see personal data;

**A** the protocol / it is not secure
**R** website not secure

HTTPS // HyperText Transfer Protocol Secure;

**2**

(e)     Word processor : General purpose (application software);
Parent portal : Bespoke;
Web server : Special purpose (application software);

**3**

**[8]**

**Q20.**

**General:**

Idea of ‘quicker to write ’ or ‘easier to write ’ [ONE MARK]
EXAMPLES:
Assembly language is quicker to write than machine code //
HLL is quicker to write (compared to assembly code) //
Assembly language is easier to write than machine code // HLL is easier to write (compared to assembly);
[or opposites – slower to write / harder to write]

Idea of ‘understanding’ [ONE MARK]
*EXAMPLES:*
Assembly code easier to understand than machine code //
HLL easier to understand than assembly code;

Idea of ‘debugging’ [ONE MARK]
*EXAMPLES:*
Assembly code easier to debug than machine code // HLL easier to debug (than assembly code);

**Assembly language:**
Solution expressed in terms of mnemonics;
**A** an example of a full instruction (operand and opcode)
Easier to make mistakes in assembly language;
Instruction composed of op-code and operand;
Solution translated by using an assembler;
Code is hard to port to other types of computer // machine-oriented languages;
One assembly language instruction relates to one machine code instruction;
***Situation*** – working on embedded hardware // need for small object code size //
need for fast execution // need to access hardware / registers directly;

**Imperative language:**
Imperative is where the programmer gives the computer a sequence of instructions to perform; Selection / Sequence / Iteration constructs available;
**A** a full example of a selection / iteration construct
Library of pre-written functions available;
Solution translated by using a compiler / interpreter;
A compiler might not be available for a specific processor (disadvantage);

**Situation** – anything sensible that would need a HLL (for example games programming)

**Declarative language:**
(Certain languages) define what is to be computed rather than how the computation is to be done;
(Certain languages) lack side effects;
(Certain languages) have a clear link to mathematical logic;
(Certain languages) express solutions in terms of facts and rules // rule-based;
(Certain languages) will use an inference engine to work out the answer;
The user asks a question of the system rather than provide an algorithm of the solution;
Uses back-chaining / backtracking;
(Certain languages) express solutions using markup languages (such as HTML);
(Certain languages) express solutions as CSS / regular expressions / (subset of) SQL;
**A** example code from part of a declarative program (ie an SQL statement)
***Situation*** – medical diagnosis // expert systems // database query // creating a web page / website;

**Imperative and Declarative language:**
Solution expressed in terms of statements written using English-like keywords;
Code easier than assembly language to port to other types of computer;
One language statement maps to many (more than one) machine code instruction;

Note: accept any sensible situation for each area

**Mark Bands and Description**

*To achieve a mark in this band, candidates must meet the subject criterion (SUB) and all 5 of the quality of language criteria(QWCx).*

*SUB*      Candidate has covered all three language generations and made at least 7
              subject-related points.
*QWC1*  Text is legible.
*QWC2*  There are few, if any, errors of spelling, punctuation and grammar.
              Meaning is clear.
*QWC3*   The candidate has selected and used a form and style of writing
              appropriate to the purpose and has expressed ideas clearly and
              fluently
*QWC4*    Sentences (and paragraphs) follow on from one another clearly and
              coherently.
*QWC5*    Appropriate specialist vocabulary has been used.

**7-8**

*To achieve a mark in this band, candidates must meet the subject criterion (SUB) and 4 of the 5 quality of language criteria (QWCx)*.

*SUB*      Candidate has covered at least 2 of the 3 generations and has made at
             least 3 subject-related points.
*QWC1*  Text is legible.
*QWC2*  There may be occasional errors of spelling, punctuation and grammar.
              Meaning is clear.
*QWC3*  The candidate has, in the main, used a form and style of writing appropriate
             to the purpose, with occasional lapses. The candidate has expressed
              ideas clearly and reasonably fluently.
*QWC4*  The candidate has used well-linked sentences (and paragraphs).
*QWC5*  Appropriate specialist vocabulary has been used.

**3-6**

*To achieve a mark in this band, candidates must meet the subject criterion (SUB) and 4 of the 5 quality of language criteria (QWCx).*

*SUB*      Candidate may not have covered all generations, but has covered at least
              one of them. At least one valid point has been made.
*QWC1*  Most of the text is legible.
*QWC2*  There may be some errors of spelling, punctuation and grammar but it
              should still be possible to understand most of the response.
*QWC3*  The candidate has used a form and style of writing which has many
              deficiencies. Ideas are not always clearly expressed.
*QWC4*  Sentences (and paragraphs) may not always be well-connected.
*QWC5*  Specialist vocabulary has been used inappropriately or not at all.

**1-2**

Candidate has made no relevant points.

**0**

Note: Even if English is perfect, candidates can only get marks for the points made at the top of the mark scheme for this question.

If a candidate meets the subject criterion in a band but does not meet the quality of written communication criteria then drop mark by one band, providing that at least 4 of the quality of language criteria are met in the lower band.
If 4 criteria are not met then drop by two bands.

**Max 8**

**[8]**

**Q21.**

(a)


1 mark – logic of first part satisfies NOT A, NOT B;

1 mark – inputs into an AND gate;

1 mark – output from AND gate passes through a NOT gate and connected to Q;

**3**

(b)

|  |  |  |
| --- | --- | --- |
| A | B | A + B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

1 mark for correct A + B column;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B |   |   |  |  |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 |

1 mark for columns  and  column being correct;
1 mark for  column being correct;
1 mark for  column being correct;

Note: Can follow through into  column from previous two

**4**

(c)     De Morgan's (law);

**1**

(d)     **Mark allocation:**

One mark for taking either A, NOT C or A AND NOT C
outside of brackets to produce a correct expression;
One mark for eliminating B in a valid way;
One mark for correct final answer;

**Example One:**A.B. + A.

A (B. + ) - taking A outside of brackets;

A ((B + 1))    ( B + 1 ) = 1
Simplifying to remove B using B + 1 = 1 ;

B. +  = 
Simplifying to remove B using B. +  = ;

**A** A( (B + 1))→ A.;

Final answer A.

**Example Two:**

A.B. + A.

A.(B + 1) – taking outside of brackets;

(B + 1 ) = 1; - simplifying to remove B

**A**  A.(B + 1)→ A.

Final answer A.

**Truth Table Method**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C |   | A. | A.B.+ A.  |
| 0 | 0 | 0 |   | 0 | 0 |
| 0 | 0 | 1 |   | 0 | 0 |
| 0 | 1 | 0 |   | 0 | 0 |
| 0 | 1 | 1 |   | 0 | 0 |
| 1 | 0 | 0 |   | 1 | 1 |
| 1 | 0 | 1 |   | 0 | 0 |
| 1 | 1 | 0 |   | 1 | 1 |
| 1 | 1 | 1 |   | 0 | 0 |

(student answer may have more columns than this)

A mark for having correct column for A.B. + A.;
A mark for having correct column for A.;

Final answer A.

**3**

**[11]**

**Q22.**

(a)     **Imperative:**Instructions are executed in a programmer defined sequence //
Instructions specify how to solve the problem;

**A** executed line by line (in sequence)

**HLL:**A language that uses English-like/more meaningful keywords // one instruction maps to several machine code instructions // has structures for assignment/iteration/selection ;

**NE** a language that is like English

**2**

(b)     Languages used for a specific problem type/domain;

**A** different uses / purposes / tasks

Access to specific data types;
Providing different function libraries;
Languages developed for specific hardware / devices ;
Languages developed for visual applications / GUIs;
Competition between different companies who develop languages;

**Max 1**

**[3]**

**Q23.**

(a)

|  |
| --- |
| **AND Gate** |
| **Input X** | **Input X** | **Output Q** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

|  |
| --- |
| **XOR Gate** |
| **Input X** | **Input X** | **Output Q** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

*1 mark for each of the output columns*

**2**

|  |  |  |
| --- | --- | --- |
| (b) | (i) |  |

[Brackets are not necessary]

*1 mark for use of correct operands (L,R,U);
1 mark for use of XOR with L,R;
1 mark for NOT U anded with other part;*

|  |  |
| --- | --- |
| alternative: |  |

*1 mark for use of correct operands (L,R,U);
1 mark for alternative XOR expression;
1 mark for AND NOT U;*

|  |  |
| --- | --- |
| alternative: |  |

*1 mark for use of correct operands (L,R,U);
1 mark for alternative XOR expression;
1 mark for AND NOT U;*

*Acceptable notation for symbols
~ for NOT
X.Y allow X AND Y, XэY,X)Y, XY
X+Y allow X OR Y, X(Y, X\*Y*

**3**

(ii)



L, R connected to XOR gate;
U connected to NOT gate;
Output of a two input AND gate connected to M;

*Max 2 if circuit does not reflect the correct logic*

**Alternative :**

****

U connected to NOT gate;
Correct gates used for L and R before last AND gate;
Output of a two input AND gate connected to M;

**Alternative :**

****

Marked as above alternative.

**3**

(c)     **Solution 1:**

|  |  |
| --- | --- |
|  | [Application of De Morgan’s Law –**1 mark**][allow simplification of double nots at same time][Simplification of A.A to A –**1 mark**][Correct solution – **1 mark**] |

**Solution 2:**

|  |  |
| --- | --- |
|  | [Application of De Morgan’s Law –**1 mark**][allow simplification of double nots at same time][Simplification of NOT A OR NOT A to NOT A – **1 mark**][De Morgan’s again to correct solution – **1 mark**] |

*1 mark for De Morgan;
1 mark for simplification;
1 mark for final answer;
Other notations as for section (b)*

*No working marks for truth table solution (asked to use De Morgan’s in question)*

**3**

**[11]**

**Q24.**

(a)     Second (generation);
**A** 2
**R** assembly code / language

*Note: Adding “assembly” / “assembler” does not talk out a valid mark for second / 2*

**1**

(b)     (memory) Address / location / offset;
**A** line number
**R** instruction number

**1**

(c)     (y) Opcode / operation code;
**A** op-code
**NE** operation
(z) Operand;

**2**

(d)     **Individual Instructions:**One to one / each assembly language instruction translates to one machine code instruction;

**Programs:**Figure 1 assembly language equivalent of figure 2 // figure 2 machine code version of figure 1 // figure 2 is assembled version of figure 1;
**NE** figure 2 “binary version” of figure 1
**NE** different generations of language

**1**

**[5]**

**Q25.**

(a)     System (software) / operating system;
**R** OS
**A** (device) driver

**1**

(b)     (i)      Software that can carry out many tasks// has many purposes/uses;
**NE** “many things” // “many functions” // “general purposes”
**R** for everyday use/tasks

**1**

(ii)     Word processor, spreadsheet, database, desktop publishing/DTP, presentation or other suitable example;

**R** web browser // image/photo editor

**1**

(c)     (i)      (It is) immediately available;
**A** “off the shelf”
(It has a) lower cost as development costs shared amongst many;
**NE** cheaper
Wide range of training/support available;
Regular upgrades available;
Can hire staff with experience of package;
More likely to be bug-free as already in widespread use/“tested” by many users;
**A** it has been “tried and tested”
**R** no bugs / tested more

**Max 2**

(ii)     No appropriate software to solve the problem exactly;
The only way to obtain software to do exactly/precisely what they want // may need particular features not in special purpose software // will be developed to match their needs;
To ensure competitors do not get access to the software;
To run on/support existing hardware/software;
No unnecessary features;

**Max 1**

**[6]**

**Q26.**

**** becomes A + B ;
**A** (A+B);
**A** A OR B;

B+ B. becomes B ;

**A** B+ A.  becomes A ;

**A** (B+1) becomes A ;

*1 mark for each*

**[4]**

**Q27.**

**Good at:**Can make precise/accurate / complex calculations / actions;
**NE** “good at maths / logic” – need the concept of complex
More consistent than humans;
Repetitive tasks;
Can work in conditions too dangerous for a human;
Working with large volumes of data;
Fast processing of data / calculations;
Can perform task without breaks / / for longer than humans;
**R** don’t get bored

**Bad at:**Image recognition;
Shape detection;
If the conditions change they adapt poorly / / not very adaptable / / learning;
**A** “can’t think for themselves”
Poor at coping with emergencies / unexpected circumstances;
Creativity / / invention / / lateral thinking;
Bad at discriminating;
Processing qualitative data;
Recognising human concepts e.g. emotion;
**A** Cannot recognise when it makes mistakes;

*Above are exemplars only. Award credit for other valid points.*

*Max 3 if all points are about just good or just bad.*

***[4]***

**Q28.**

|  |  |
| --- | --- |
| (a)     **Number** | **Component Name** |
| 1 | Memory Address Register |
| 2 | Address Bus |
| 3 | Memory Data / Buffer Register |
| 4 | Data Bus |

**4**

(b)     The instruction is held in the CIR;
**A** IR
The control unit / instruction decoder decodes the instruction;
The opcode identifies the type of instruction it is;
Relevant part of CPU / processor executes instruction;
**A** ALU
Further memory fetches / saves carried out if required;
Result of computation stored in accumulator / register / written to main memory;
Status register updated;
If jump / branch instruction, PC is updated;
**A** SCR

**Max 3**

(c)     Can be displayed in less space;
**R** takes up less space **NE**Easier to remember / learn / read / understand;
Less error prone;

**Max 1**

(d)     (i)     Assembler;

**1**

(ii)     HLLs are problem oriented;
HLL programs are portable // machine / platform independent ;
English like **keywords / commands/ syntax / code**;
**R** closer to English
Less code required // less tedious to program //
one to many mapping of HLL statements to machine code commands;
Quicker/easier to understand / write / debug /learn / maintain code;
**R** just quicker/easier
HLLs offer extra features e.g. data types / structures // structured statements // local variables // parameters // named variables/constants;
**R** procedures / modular
**A** example of a data structure
**NE** “extra features” without example
Speed of execution not crucial for most tasks so faster execution of assembly language not required;
Most computer systems have a lot of (main) memory / RAM so compact object code not essential;
**A** converse points for Assembly Language

**3**

**[12]**

**Q29.**

(a)



*1 mark for each correct output column***A** True for 1, False for 0

**2**

(b)



*1 mark for NOT gate correctly linked to input C;
1 mark for AND gate correctly linked to B and  as input;***A** if AND gate linked directly to C
*1 mark for OR gate with inputs from A and the output of an AND gate and output connected to Q;*

**3**

(c)     **Algebraic solution:**

B.( A +  )

*B.A + B. *         [*1 mark for expansion of brackets* ]

*B.A + 0*              [*1 mark for identifying that B. = 0* ]

*B.A*                    [*1 mark for correct answer*]

**Truth table solution:**

****

*1 mark for both columns X and Y correct
1 mark for column Z correct
1 mark for correct answer (B.A)*

***Any other method:****If student has used any other method to arrive at correct answer then award marks as follows:
1 mark for correct answer, no working out
2 marks for correct answer with working out, not all steps shown.
3 marks for correct answer with all steps of working out shown.*

**A** True for 1, False for 0
**A** alternative notations :

•        For X.Y allow X AND Y, X∧Y ,X∩Y, XY

•        For X+Y allow X OR Y, X∨Y , XUY

•        For  allow NOT X, ¬X

**3**

**[8]**

**Q30.**

**Algebraic Solution:**

|  |  |
| --- | --- |
| **Method 1** | **Method 2** |
|      + A      =  +  + A     = 1 +       = 1 |      + A      =      =       =       = 1 |

*1 mark for an application of a DeMorgan’s law
1 mark for realisation that* *A* +  +  = 1 +  or  =  (must be written in method, not just inferred that student has done this if arrives at correct answer)
*1 mark for correct answer*

**Truth table solution:**

****

*1 mark for column Y correct
1 mark for column Z correct
1 mark for correct answer*

***Any other method:****If student has used any other method to arrive at correct answer then award marks as follows:
1 mark for correct answer, no working out
2 marks for correct answer with working out, not all steps shown.
3 marks for correct answer with all steps of working out shown.*

**A** True for 1, False for 0
**A** alternative notations :

•        For X.Y allow X AND Y, X∧Y ,X∩Y, XY

•        For X+Y allow X OR Y, X∨Y , X∪Y

•        For  allow NOT X, ¬X

**[3]**

**Q31.**

(a)     (i)      (Data/address/control/internal/system) bus;

**R** just a description of a bus

**R** names of buses which don’t exist e.g. memory bus

**1**

(ii)     Store programs and/or data/files when not in use/

When computer is off permanent/long term storage

Of programs and/or data; save programs/data;

**R** offline/backup **R** ROM **R** temporary storage

**A** save on magnetic disk/ tape storage;

**A** information instead of data

**1**

(iii)     (Machine code) instruction/data is fetched from main memory;

**A** what is fetched or from where

Instruction is decoded;

Instruction is executed (by the processor); **R** data executed

**Max 2**

(b)     (i)      Assembly language; mnemonic code; mnemonics; assembly code;

**R** low level language **A** assembler;

**1**

(ii)     Translated/assembled/converted/decoded; into machine code (instructions);

**R** compiled **R** interpreted **A** object/target code;

**A** binary instructions;

**2**

(iii)     Computer executes instructions in programmer defined sequence;

**A** the programmer tells the computer how to do it;

**R** user *instead of* programmer

**1**

(iv)    Pascal /Visual Basic/Basic/C/C++/Cobol/Fortran/Ada/Delphi/Lylix/Modula /or any other imperative HLL

**R** Prolog
**R** Lisp
**R** Pop11

**1**

(v)     One statement/instruction/command in a high level language translates into several machine code instructions; 1 to many;

**1**

(vi)    Laborious/time-consuming to write; hard to debug; harder to program; easier to make mistakes; more difficult to understand/ learn; difficult to maintain; different assembler/instruction set for different type of computer; machine dependent; low level programs not portable;

**Max 2**

**[12]**

**Q32.**

(a)     (i)      Layer of software which enables users to operate computer; software to make hardware work; operating system;

**Max 1**

(ii)     Program written to perform end user task; problem-oriented program  to enable user to do specific task; program produced by end use;

**Max 1**

         (b)     (i)      Operating system / utility programs / library programs / compilers / assembler / interpreter;

**1**

(ii)     Word processor / DTP / Spreadsheet / Database / stock control / payroll / web browser / or other suitable

**1**

**[4]**