Identification number 1:  \[
\text{working} = (4 \times 6) + (2 \times 5) + (1 \times 4) + (9 \times 3) + (2 \times 2) + (3 \times 1)
\]
\[
= 24 + 10 + 4 + 27 + 4 + 3
\]
\[
= 72 \div 11 = 6 \text{ remainder } 6
\]
valid/not valid: NOT valid

Identification number 2:  \[
\text{working} = (8 \times 6) + (2 \times 5) + (0 \times 4) + (1 \times 3) + (5 \times 2) + (6 \times 1)
\]
\[
= 48 + 10 + 0 + 3 + 10 + 6
\]
\[
= 77 \div 11 = 7 \text{ remainder } 0
\]
valid/not valid: VALID

(b) 1 mark for correct working + 1 mark for check digit

\[
\text{working} = (5 \times 6) + (0 \times 5) + (2 \times 4) + (4 \times 3) + (1 \times 2)
\]
\[
= 30 + 0 + 8 + 12 + 2
\]
\[
= 52
\]
need to add 3 to make the total 55 (i.e. exactly divisible by 11)

check digit: 3

(c) 1 mark for each description and example

2 digits transposed  
(e.g. 280419 becomes 280149/two digits have been switched)
incorrect digit  
(e.g. 280419 becomes 250419/one of the digits has been mistyped)

2  
direct access because of concentric tracks
  can read and write at the same time because it has a read/write head
3  (a) 1 mark for each logic gate correctly connected

\[ \text{Diagram of logic gates} \]

(b) 1 mark for each digit

<table>
<thead>
<tr>
<th>A</th>
<th>T</th>
<th>S</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

1 mark

1 mark

1 mark

1 mark

1 mark

1 mark

4  (a) 1 mark for hours; 1 mark for minutes

1 6 : 4 9

1 mark  1 mark

(b) 1 mark for each digit

| 0 0 0 1 | 1\(^{\text{st}}\) digit |
| 0 1 1 1 | 2\(^{\text{nd}}\) digit |
| 0 0 1 0 | 3\(^{\text{rd}}\) digit |
| 1 0 0 1 | 4\(^{\text{th}}\) digit |

1 mark

1 mark

1 mark

1 mark

1 mark
(c) Any **two** from:
- microprocessor compares present time with stored time
- if the values are the same
- sends signal to sound alarm

5 (a) Yes
(b) No
(c) – re-reading the byte that was sent
    – request that the byte is resent

6 (a) Only answers:
- temperature (sensor)
- oxygen (sensor)
(b) Any **four** from:
- information from the sensors sent to microprocessor
- the ADC converts the analogue data into digital form
- if temperature < 25°C  OR  temperature checked against stored value
- ...microprocessor sends signal to heater/actuator/valve...
- ...to switch on heater
- if oxygen level < 20 ppm  OR  oxygen level checked against stored value
- ...to open valve/oxygen supply
- use of DAC between microprocessor and devices
- sounds an alarm if system unable to respond
- continuously monitors sensor inputs
- any reference to feedback
(c) Any **one** from:
- unsafe limit stored in memory
- warning sound/signal if too high a value reached
- fail safe switch off in case of a malfunction
7  (a)

- authoring language used to create documents to be viewed on the World Wide Web
- computer that responds to requests to provide information and services over the Internet
- defines how messages are transmitted and formatted over the Internet
- numerical ID for each device on the Internet
- software that enables users to access/view documents and other resources on the Internet
- unique ID for a network interface card

Browser
HTML
MAC address
Internet Server
IP address
http

5/6 matches – 5 marks
4 matches – 4 marks
3 matches – 3 marks
2 matches – 2 marks
1 match – 1 mark

(b) any two from:
- to enable logon information to be kept on his computer
- to provide pages customised for Ahmed the next time he logs on
- to implement shopping carts and one-click purchasing
- to be able to distinguish between new and repeat visitors to the website
8  (a)  (i) Any **one** from:
   - unit of data/memory
   - 8 bits
   - used to represent a character  \[1\]

   (ii) Any **one** from:
      - \(2^{30}\) bytes
      - 1 073 741 824 bytes
      - 1 048 576 kilobytes
      - 1024 megabytes  \[1\]

(b) Any **two** from:

- **Flash memory**
  - solid state memory
  - no formatting issues
  - plugs directly into the USB port
  - direct transfer of data

- **CD-RW**
  - optical media
  - slower access speed/flash memory has faster access speed
  - requires a separate drive
  - data needs to be burnt/finalised/finished (before being used on another device)  \[2\]

9  (a) Any **one** from:
   - buffer
   - RAM  \[1\]

(b) – interrupt  \[1\]

10 (a) 1 mark for each correct word

(i) Hello World  \[2\]

(ii) Nmilozgu Pnwgyng  \[2\]

(b) – use of Secure Socket Layer
    - the key itself is encrypted using strong encryption  \[2\]
data loss caused by hard disk head crash

hacking into files and changing or deleting data

introduction of software that self-replicates and can cause data loss

reading of illegally accessed documents

software that logs/records all key presses on your computer without you knowing

anti-spyware software

anti-virus software

back-up files

encryption

passwords and a firewall

5/4 matches – 4 marks
3 matches – 3 marks
2 matches – 2 marks
1 match – 1 mark

12 (a) code B

(b) Any one from:
– no need to understand workings of a computer
– easier to understand for programmer/closer to English
– much easier to debug
– much easier to test
– one-to-many when writing commands
– not machine-specific/portable

(c) Any one from:
– can address memory addresses directly
– no need for compilers/interpreters
– shorter code/code requires less storage/RAM
– can be written to run faster
(d) – compiler produces object code / interpreter doesn’t produce object code
– compiler translates whole program in one go / interpreter translates and executes line at a time
– compiler produces list of all errors / interpreter produces error message each time an error encountered
– compiler produces “stand alone code” / interpreter doesn’t produce “stand alone code”
– compilation process is slow but resultant code runs very quickly / interpreted code runs slowly

13 (a) (i) 01000001 01000011
(ii) 41 43

(b) FA97

(c) – easier to identify values
– easier to spot errors