CHAPTER 2 – REVISION WORKSHEET
ANSWERS

2.1 Types and methods of data transmission

0478/11/M/J/15

1 (a) parallel

any **one** from:

- 8 bits/1 byte/multiple bits sent at a time
- using many/multiple/8 wires/lines (1 mark)

serial

any **one** from:

- one bit sent at a time
- over a single wire (1 mark) [2]

(b) parallel

- faster rate of data transmission (1 mark)

serial

any **one** from:

- more accurate/fewer errors over a longer distance
- less expensive wiring
- less chance of data being skewed/out of synchronisation/order (1 mark) [2]

(c) parallel

any **one** from:

- sending data from a computer to a printer
- internal data transfer (buses) (1 mark)

serial

- connect computer to a modem (1 mark) [2]

0478/11/M/J/15

2 (a) – universal serial bus
– description of USB [1]

(b) Any **two** from:

– devices are automatically detected and configured when initially attached
– impossible to connect device incorrectly/connector only fits one way
– has become the industry standard
– supports multiple data transmission speeds
– lots of support base for USB software developers
– supported by many operating systems
– backward compatible
– faster transmission compared to wireless [2]

0478/12/M/J/16

6 (a)

Type	Tick (✓)
simplex	
half-duplex	
full-duplex	✓

Method	Tick (✓)
serial	
parallel	✓

Type	Tick (✓)
simplex	✓
half-duplex	
full-duplex	

Method	Tick (✓)
serial	✓
parallel	

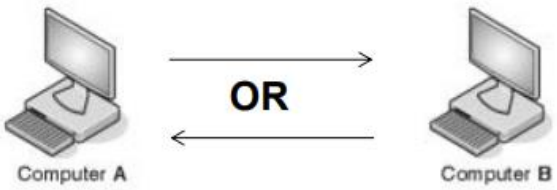
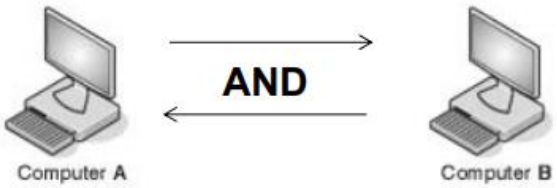
Type	Tick (✓)
simplex	
half-duplex	✓
full-duplex	

Method	Tick (✓)
serial	✓
parallel	

[6]

(b) Any **two** from:
– single wire means there is less chance of interference/data corruption
– single wire reduces costs
– more reliable over greater distances
– bits will still be synchronised after transmission [2]

0478/12/M/J/17

Question	Answer	Marks
7(a)	1 mark for correct arrow(s), one mark for correct description **Simplex data transmission**  (Direction of data is) one way only // unidirectional **Duplex data transmission**  (Direction of data is both ways) at same time / simultaneously / concurrently **Half-duplex data transmission**  (Direction of data is both ways) but at different times / not at the same time / not simultaneously / not concurrently	6

Question	Answer	Marks
7(b)	1 mark each use, must be different. **Simplex** e.g.: Microphone to computer Sensor to computer Computer to printer Computer to speaker Computer to monitor Webcam to computer Sending data to a device // sending data from a device **Duplex** e.g.: Telephone call Voice over IP Computer to printer (only award once) Instant messaging Broadband connections Video conferencing Sending data to and from devices e.g wireless technology Computer to modem	2

7(c)	2 marks for IC, 2 marks for USB **IC** ∞ parallel transmission // description of parallel ∞ for sending data internally **USB** ∞ serial transmission // description of serial ∞ for sending data externally (to and from peripherals / between devices)	**4**

0478/12/F/M/23

5(a)(i)	Any **two** from: e.g. • **Destination** IP/address • Packet number • **Originators** IP/address • Error detection method	**2**
5(a)(ii)	**One** mark each: • Payload • Trailer	**2**

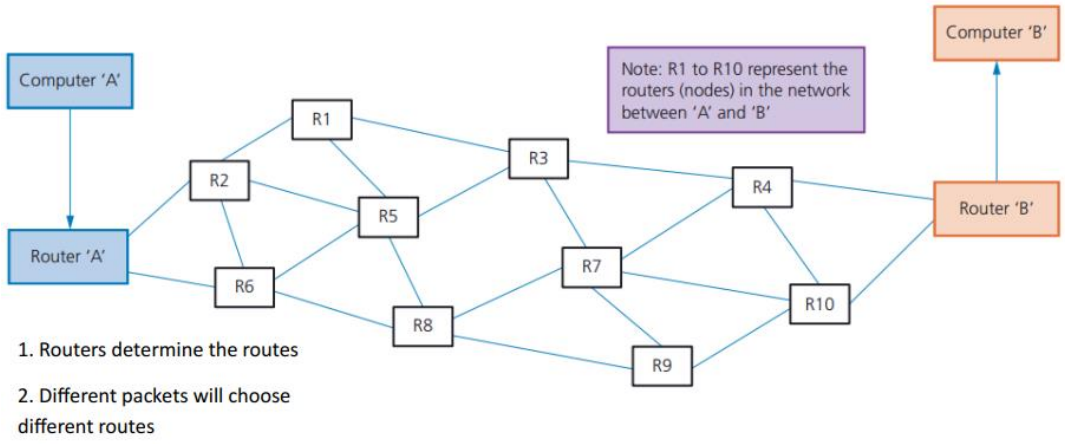
0478/11/M/J/23

2(c)(ii)	Any **five** from: • Data is **broken/split/divided** into packets • Each packet (could) take a different route • A **router** controls the route/path a packet takes • … selecting the **shortest/fastest** available route/path • Packets may arrive out of order • Once the **last packet** has **arrived**, packets are **reordered** • If a packet is missing/corrupted, it is requested again

0478/13/M/J/23

5	The diagram demonstrates (**one** mark for each): • Packets sent through several routers • … taking different routes from device A to device B • Packets arrive out of order • Packets being reordered when all arrived at device B

Example:



0478/11/MJ/25

4(a)(i)	**Two** from (one mark for serial, one mark for full-duplex): **serial** • Serial is suitable as data may be travelling a long distance (over 5 metres). • Serial is suitable to make sure that bits don't arrive skewed/arrives in order. • Serial is suitable as there is less chance of error. • Serial has sufficient transmission speed as not much data being sent. **full-duplex** • Full-duplex is suitable as data needs to go between customer and kitchen computer // data needs to go in both directions // so notification of errors can be sent back to customer.
4(a)(ii)	The data may be transmitted faster.
4(c)(i)	D
4(c)(ii)	Any **Two** from: • The packets **arrive** in the wrong order. • Each packet could take a different route. • Some packets may take longer than others to be transmitted. • Some packets may not arrive/be corrupted and need to be requested again.

0478/11/ON/25

7	**One** mark for each correct data transmission method. • Serial • Simplex Any **three** from: Serial • Data will need to travel long distances • One wire is used // bits are sent in order // sent one bit at a time • Less chance of data being skewed // less chance of error • Transmission speed would be adequate Simplex • Data **only** needs to be sent in **one direction** • Data does not need to be **sent back** from the **speakers** to the microphone

0478/12/MJ/25

(c)(ii)

One mark from:

- Serial
- Parallel

One mark from:

- Half-duplex
- Full-duplex
- Simplex

Any **four** from (for descriptions matching transmission types given):

serial

- Serial would send bits in order // serial uses only one wire.
- … so won't be skewed // less likely to have errors
- Serial transmission speed would be adequate.

parallel

- Parallel would transmit data faster.
- … as multiple bits are sent at the **same time** // … as multiple wires are used.
- For parallel, chance of **skewing/errors** would be low as **short distance** transmission only required.

half/full duplex

- To allow data to be sent in both directions
- … so any interrupts/notifications for errors can be sent back to the computer.

simplex

- Data **only** needs to be sent one direction // Data transmission doesn't need to be two-way.
- … as the printer may not need to send errors back to the computer

0478/12/ON/25

(c)(i)	• It is sent one bit at a time • It is sent down a single wire
(c)(ii)	Any **two** from: • The mouse can be powered by the device • It is a universal connection • It is backward compatible • Less chance of skewing // Less chance of errors • Supports different data transmission speeds • Adequate transmission speeds • The cable cannot be inserted incorrectly • The device is **automatically detected** // **Driver** is **automatically installed** when device is connected
(c)(iii)	Any **one** from: • Slower transmission than a **parallel** connection • Wire may get in the way // wire may limit movement of mouse

2.2 Methods of error detection

0478/12/M/J/15

5 (a) 1 mark per correctly placed tick

Received byte	Byte transmitted correctly	Byte transmitted incorrectly
1 1 0 0 1 0 0 0		✓
0 1 1 1 1 1 0 0		✓
0 1 1 0 1 0 0 1	✓	

[3]

(b) (i) byte number: **7**

column number: **6**

[2]

(ii) Any **two** from:

- letter "A"(byte 7) transmitted as odd parity (three 1s)
- column 6 has odd parity (seven 1s)
- intersection of byte 7 and column 6 indicates incorrect bit value

[2]

(d) Any **one** from:

- 2 bits interchanged (e.g. 1 → 0 and 0 → 1) that won't change parity value
- even number of bits/digits are transposed
- If there are multiple errors in the same byte/column, that still produce the same parity bit, the error will not be detected

[1]

0478/12/O/N/15

7 (a) (i) 1 mark for correct check digit and 1 mark for showing the calculation

(4 × 1) + (2 × 2) + (4 × 3) + (1 × 4) + (5 × 5) + (0 × 6) + (8 × 7)

= 4 + 4 + 12 + 4 + 25 + 0 + 56 = 105

105/11 = 9 remainder 6

check digit is: **6** [2]

(ii) 1 mark
- No/incorrect check digit

2 marks
- Total is 78
- 78/11 …
- … gives 7 remainder 1
- check digit should be 1

[3]

(b) (i) 1 mark for each correct parity bit

parity bit

0	1	1	0	0	1	1	0

parity bit

1	0	0	0	0	0	0	1

[2]

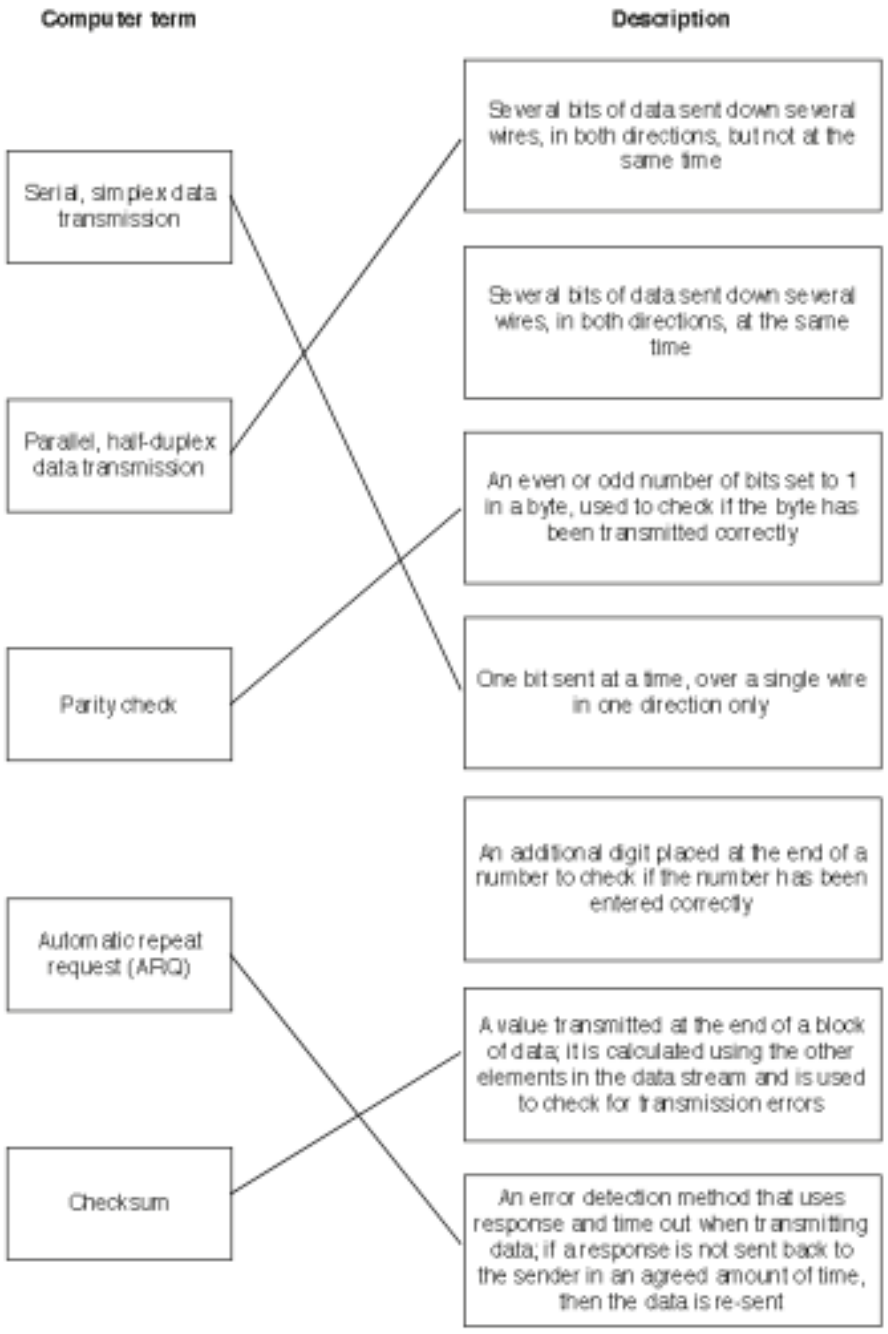
(ii) Any **one** from:
- an even number of digits are changed
- a transposition error(s) has occurred

[1]

0478/13/O/N/23

5(a)	– Interference // crosstalk
5(b)	– C
5(c)	Any **five** from: – Timer is started when sending device transmits a data packet to receiver – Receiving device checks the data packet for errors – Once the receiving device knows the packet is error free it sends an acknowledgement back to the sending device … – … and the next packet is sent – If the sending device does not receive an acknowledgement before the timer ends … – … a timeout occurs – … the data packet is resent … – … until acknowledgement received // until max number of attempts reached

3


Computer term
Description
Serial, simplex data transmission
Parallel, half-duplex data transmission
Parity check
Automatic repeat request (ARQ)
Checksum
Several bits of data sent down several wires, in both directions, but not at the same time
Several bits of data sent down several wires, in both directions, at the same time
An even or odd number of bits set to 1 in a byte, used to check if the byte has been transmitted correctly
One bit sent at a time, over a single wire in one direction only
An additional digit placed at the end of a number to check if the number has been entered correctly
A value transmitted at the end of a block of data; it is calculated using the other elements in the data stream and is used to check for transmission errors
An error detection method that uses response and time out when transmitting data; if a response is not sent back to the sender in an agreed amount of time, then the data is re-sent

[5]

0478/11/M/J/17

5(a)	1 mark per correct tick **Received byte** / **corrupted during transmission (✓)** / **not corrupted during transmission (✓)** 10110100 / ✓ / 01101101 / / ✓ 10000001 / ✓ /	**3**
5(b)	**Four** from: ∞ Uses acknowledgement and time out ∞ Check performed on received data // error is detected by e.g. parity check, check sum ∞ If error detected, request sent to resend data // negative acknowledgment is used ∞ If no acknowledgement is sent that data is received // positive acknowledgement is used ∞ Data is resent / Resend request repeated, till data is resent correctly … ∞ … or request times out // limit is reached	**4**

0478/11/MJ/25

4(b)(i)	Any **five** from: • The checksum is **calculated** from/using the **data**. • … using an algorithm // by example. • The (checksum) value is transmitted with the data. • The (checksum) value is recalculated **after transmission/by recipient**. • If the **checksum values** do not match, an error is detected // If the **checksum values** match, no error is detected … • … request for data to be resent if there is an error.
4(b)(ii)	Any **one** from: • Parity byte/block check • Echo check

0478/12/MJ/25

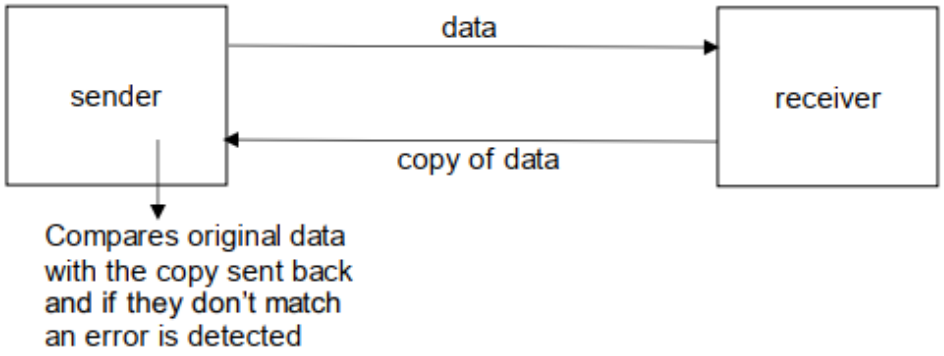
(c)(iii)	• A parity bit is added to each **byte.** • … to make the number of 1s/0s even // that will be 1 if the number of 1s/0s is odd // that will be 0 if the number of 1s/0s is even. • The number of 1s/0s in each byte is counted **after transmission.** • If any bytes have an odd number of 1s/0s an error is detected

0478/11/ON/25

(d) Any **three** from:

The diagram shows:

- (Data being sent to the receiver) and a copy sent back from the receiver to the sender
- Sending device comparing the original data and the copy
- If original data and copy don't match an error is detected



2.3 Encryption

2210/13/O/N/15

10 symmetric encryption

encryption key

plain text

encryption algorithm

cypher text [5]

0478/11/O/N/22

(b)(i)	• The data will be **meaningless** if it is stolen
(b)(ii)	**One** from: • Data is **encrypted** and **decrypted** using the **same** key/algorithm Any **three** from: • Data before encryption is known as plain text • Data after encryption is known as cypher text • Key is sent to receiver (to allow data to be decrypted) // Values are sent to receiver that are used to generate key

0478/13/M/J/19

3(b)	**Four** from: – An encryption algorithm is used – … to scramble data – The original data is called the plain text – A key is used to encrypt the data – The key is applied to the plain text – Plain text is encrypted into cypher text

0478/13/O/N/19

1(b)(i)	∞ Increase the length of the key // make key 12-bit, etc.
1(b)(ii)	∞ Cypher text

0478/11/MJ/25

6(a)	**One** mark for each correct term in the correct position • algorithm • plain text • cipher text • meaningless • public/private • private/public (must be opposite of MP5) • public • private Data is encrypted using an encryption key. This is a type of **algorithm** that scrambles the **plain text** and turns it into **cipher text**. This makes the data **meaningless**. A **public/private** key is used to encrypt the data. This key cannot be used to decrypt the data. A **private/public** key is used to decrypt the data. Any device is able to request the **public** key, but only your device knows the **private** key.
6(b)	Any **one** from: • The process is more secure. • Do not need to worry about key exchange.