5.5 Information coding systems Mark Scheme

Q1.

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

A character code uses a unique number / code to represent each different character;

1

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

1 mark: `b` = 1100010;
1 mark: `e` = 1100101;

2

(c) **Mark is for AO2 (apply)**

1000011

1

[4]

Q2.

(a) Keyboard / / keypad / / concept keyboard / / numberpad;
Touch-screen;
R mouse

2

(b) A light source / laser is shone at bar code / / a bar code is illuminated; **NE** beam / photons

(moving) mirror / prism moves light beam across bar code / / user moves reader across bar code / / user moves the bar code across the reader;
NE beam

Light reflected back;

Black / white bands reflect different amounts of light / / black reflects less light / / white reflects more light;

Light sensor / photodiode / CCD (measures amount of reflected light);

Light reflected converted into an electrical signal;
A convert reflection to (binary) numbers / characters / ASCII

Check Digit:
The (12) data digits are passed through a function to calculate a check digit;

The result is compared against the check digit read in / / check digit compared to rest of bar code;

If they do not match an error is indicated;

If they match the bar code is accepted and processed;

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 made at least five subject-related points. Candidate has made valid points about both scanning **and the check digit** in their answer.
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.

5-6

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 has made at least three 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.

3-4

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 made at least one subject-related point.
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 language 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 6

[8]

Q3.

(a) (i) Zero or more `b`s followed by a / one `c`;
A answers by example but must be at least `c`, `bc`, `bbc` and indicate the sequence continues.

1

(ii) Zero or one `b`s followed by (a / one) `c` / / the strings `c` or `bc`;

1

(b) Correct expression: `b(cd)*(e|fg)`
A use of incorrect bracket types
A accept brackets around `fg`
A `(cd)`$^{+}$`?` for `(cd)*`
I `^` at start, `$` at end of expression
2 marks for the full correct expression.
1 mark for including either `(cd)*` or `(e|fg)` in an incorrect expression.

2

[4]

Q4.

(a) Unicode uses more bits for each character //
ASCII uses fewer bits for each character //
Unicode can represent a wider range of characters //
ASCII can represent a smaller range of characters //
Unicode uses 16 / 32 bits, ASCII uses 7 bits (**A** 8 bits);

1

(b) **Role of sender:**
Sender counts / checks the number of 1s in the bit pattern/value/data; and adds an extra bit to ensure even number of 1s;
//
Sender adds a 0 parity bit if there are an even number of 1s in the bit pattern / value / data; if odd number of 1s then a 1 parity bit is added;

Role of receiver:
Receiver counts / checks the number of 1s in the bit pattern / value / data received;
if there are an odd number of 1s it identifies that an error has occurred;
A if even number of 1s it accepts the data received;
A if even number of 1s data is assumed to be correct;
A if odd number of 1s it requests that the data be resent;
R if even number of 1s, data is correct
//
Receiver regenerates parity bit from data received; compares generated parity bit with received parity bit – if different it identifies that an error has occurred;
A if the same it accepts the data received;
A if the same data is assumed to be correct;
A if different it requests that the data be resent;
R if the same, data is correct
A an odd number of errors (in the bit pattern received) will be detected;

Marking Guidance
R Implication that sender or receiver are people
Max 2 if role of sender <u>and</u> receiver not included in answer
R if mark point is about bit pattern being even / odd rather than the number of 1s being even / odd

Max 4

(c) Error correction (not just error detection) (for single errors);
Can detect when two errors have occurred in data transmission;
Reduces the need for the retransmission of data;
Decreases the likelihood of an undetected error // improved error detection;
R Implication that the parity bits are calculated by a person

Max 1

(d) 0;

1

(e) 1, 4, 8;

Note: order of answers unimportant

1

[8]

Q5.

(a) 011 0010;
R If not 7 bits

1

(b) 1011 0000

Mark as follows:
Correct data bits;
Correct parity bit for the candidate's data bits;
R If not 8 bits

2

(c) Error correction (not just error detection) (for single errors);
Can detect when two errors have occurred in data transmission;
Reduces the need for the retransmission of data;
Decreases the likelihood of an undetected error // improved error detection;
Can locate an error (not just detect that an error has occurred);

Max 1

[4]

Q6.

(a) 011;

1

(b) 010;

1

(c) 110;

1

(d) Gray code counters consume half the/less electrical power;
Prevents some errors that can happen when the value changes;
(When a value is incremented only one bit changes at a time therefore) there is less likelihood of an error occurring;
A Fewer errors

Max 1

[4]

Q7.

(a) 128 // 2^7;

1

(b) 1000010;
R more than 7 bits used

1

(c) 01000001

Mark as follows:
Correct parity bit for the candidate's data bits;
Correct data bits;
R if not 8 bits

2

(d) <u>Sender</u> counts/checks the number of 1s in the bit pattern/value/data; (If even number of 1s then 0 parity bit is added; if odd 1 is added;) // Extra bit added to ensure even number of 1s;

<u>Receiver</u> counts/checks the number of 1s in the bit pattern/value/data received; If odd it identifies that an error has occurred; and requests for data to be resent;
A If even it accepts the data received
A if even data is assumed to be correct;
A an even number of errors will be detected;
R if even, data is correct
// receiver regenerates parity bit from data received; compares generated parity bit with received parity bit; if different requests for data to be resent
R Implication that sender or receiver are people.

Max 4

[8]

Q8.

(a) **For Photodiode System:**
Light / laser / LED / Infra-red light shone at bar code; **NE** beam
(Moving) mirror / prism moves light beam across bar code / / user moves reader across bar code; **NE** beam
Light reflected back;
Black/white bands reflect different amounts of light / / black reflects less light / / white reflects more light;
Light sensor / photo sensor / photo diode / CCD measures amount of reflected light;
Light reflected converted into an electrical signal; **A** convert reflection to (binary) numbers / characters
(Electrical form of) reflection analysed to determine value encoded in bar code;
Data transmitted as binary codes to till / computer;
These values are often sent as ASCII codes;

For Camera / CCD System:
Camera / CCD measures (ambient) light reflected from bar code;
Camera / CCD converts light into an electrical signal;
Light reflected back;
Black areas reflect less light than white;
Raw image data transmitted to computer;

Image analysis software analyses image to determine value encoded in bar code;

Max 4

(b) Validate data entry//check bar code is valid/reasonable;
Verify if bar code has been "input" accurately/correctly //check bar code not damaged / altered;
R validate the item

Max 1

(c) Keyboard/Keypad/Touch screen/Concept Keyboard/Electronic Scales
NE scales

1

[6]

Q9.

(a) acknowledge data received by the printer;
error (signal);
busy / free / ready /'status' / acknowledge / strobe / off-line / powered/ switched off / out of paper;
A ground / earth / return;
R Interrupt / clock

Max 1

(b) (i) 110 1010;

1

(ii) 0110 1010;

1

(iii)

	A	B	C
8	**0**	**0**	**0**
7	**0**	**1**	**1**
6	**1**	**0**	**1**
5	**0**	**1**	**0**
4	**1**	**0**	**1**
3	**0**	**1**	**0**
2	**1**	**1**	**1**
1	**1**	**0**	**0**

8 parallel bits; (above, below or between the lines)
pattern A scores 2;
pattern B scores 2;
pattern C scores 1;

Max 3

(c) (i) set of rules (about the way devices communicate);

A standards **R** Instructions

1

(ii) sending <u>signals</u> between devices + implication of 2-way; test to see if the device is ready to receive /'are you ready?'; inform device that the data has been sent / 'here it is …'; receiver informs the sender that the data has been received / acknowledge that a transfer is completed;

Max 2

(d) (i) Universal Serial Bus;

1

(ii) Line 1 used with 7/8 bits shown (above or below);
Correct code **0110 1010**;

2

(iii) The number of bits transferred per sec / per unit of time;
I. speed
A frequency at which bits are transmitted

1

[13]

Q10.

(a) 255 / 11111111 / FF / 2^8 – 1 ;

1

(b) (i) 81 ;

1

(ii) 1000 1010 ;

1

(c) 522

1

(d) (i) 100 0010 ;
R leading 0 as 8^{th} bit

1

(ii) The total is calculated <u>before transmission</u> ;
0 or 1 bit is added (to the 7-bit code) ;
So that, the total number of 1 bits must compute to an even number ;
The number of one bits is <u>re-calculated after received</u> ;
An odd number of 1 bits indicates an error ;

Max 2

[7]

Q11.

(a) E X A M ;

Mark as follows:
1 or 2 correct (1) ;
3 correct (2) ;
4 correct (3) ;
R lower case

3

(b) (i) Universal Serial Bus;

1

(ii) Parallel ;

1

(iii) Set of rules ;
Sending signals between devices;
(Computer) asks are you ready? ;
(Printer) acknowledges yes I am ;
(Computer) responds here comes the data ;
(Printer) 'thank you received' ;

Max 2

(iv) Acknowledge data received by the printer ;
Error ;
Line is busy / free / ready /'status' / ACK Request ;
Timing / strobe;
Interrupt;
R Ground

Max 1

(v) Operating system ;
Word processing software / text editing software / any sensible Application ;
Print spooler ;
Printer driver ;
R 'printing software'

Max 2

[10]

Q12.

(a) Lower case letters / digits /symbols;
A numerals, number, integer, lower case characters, denary char., space
A by example
R 'lower case', 'decimal'

1 mark for 2 correct answers

1

(b) Tabs (indent) / line feeds / carriage returns(enter);
A EoF, backspace, form feed (page break)

1 mark for 2 correct answers

1

[2]

Q13.

(a) (i) 101 0110;

1

(ii) 1101 0110 / or follow-through from 'their 7 bit code' from (a);
A Parity bit positioned in bit position 0

1

(b) (i) 'D';

R Lower case 1

(ii) ‘J’;
R Lower case 1

(c) (i) FirstName // Surname; 1

(ii) Surname // FirstName; (i) and (ii) must be different 1

(iii) FullName;
I. any incorrect case 1

(d)

Position	NextNumber	NextChar	FinalString
			‘’
1	(65)	‘A’	‘A’
2	78	‘N’	‘AN’
3	32	<Space>	‘AN ‘
4	69	‘E’	‘AN E’
5	82	‘R’	‘AN ER’
6	82	‘R’	‘AN ERR’
7	79	‘O’	‘AN ERRO’
8	82	‘R’	‘AN ERROR’

Position values incrementing to at least 4;to maximum 8;
NextNumber[2] has value 78;
Remaining NextNumber values are all correct and in correct positions;
NextChar[3] has <Space> character + NextNumber[3] is 32;
FinalString correct / f/t from their NextChar column; 6

[13]

Q14.

(a) (i) **Do not mark this part**

(ii)

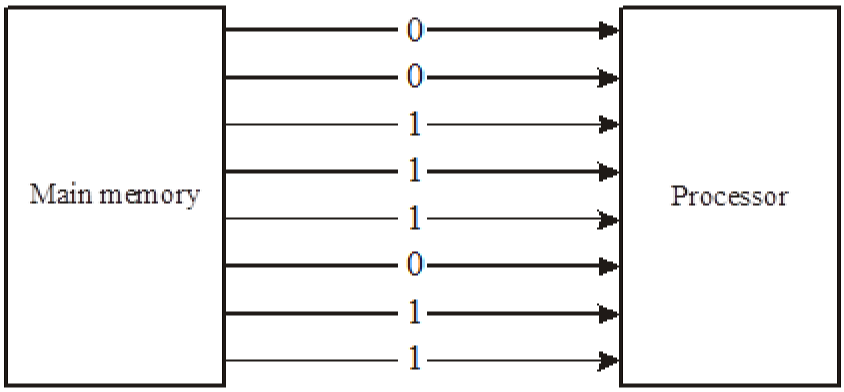

Diagram has 8 lines;
Diagram has the correct 1 bit per line; A. pulses to indicate 0/1's

2

A either 'top to bottom' or 'bottom to top' labelling of the bits
I. the direction of any arrows

(b) Interpretation:

- program instruction/command;
- character / ASCII code / 7 data bits + parity bit;
- integer / 59 / number;
- real / number;
- byte/pixel from a graphics file; R. 'part of
- byte/sample from a sound file; R. 'part of
- an address;

R BCD digits

Max 3

[5]

Q15.

(a) B<u>arcode</u> reader / wand / scanner;

If 2 answers given – TO

1

(b) (i) Less chance of error / greater accuracy; BOD 'No Error'// scanning by device faster;
A Quicker (allow in this case)
R harder to forge

1

(ii) The bars can be read up-side-down/ has vertical symmetry;
A can be read even if not in perfect condition;

1

[3]

Q16.

(a) (i)

0	0	1	1	0	0	1	0	0	0	1	1	0	1	1	1

1

(ii)

0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1

1

(b) Unicode is 16 bit code; 1

(c)

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

1

[4]

Q17.

(a) (i) 52; 1

(b) (i) '4' // 4 ; 1

(ii) UNICODE // EBCDIC // EBCD // extended binary coded decimal // extended binary coded decimal interchange code;
A minor misspelling of EBCDIC 1

(c) (i) Each pixel stored in several bits/one byte/one word;
Each colour represented by a different value; 2

(ii) Endpoints // a pair of / two (x,y) co-ordinates // start point, direction and length;
Type of object / shape;
Thickness of shape / line;
Colour of shape/line;
A Properties of shape/line on its own; 3

[8]

Q18.

(a) (i) Read Only Memory; 1

(ii) Random Access Memory; 1

(b) (i) <u>Disk</u> <u>Controller;</u> 1

(ii) Network Interface Card//Network Adapter;
A Network Card 1

(c) <u>Address</u> Bus;
Control Bus;
Data Bus; 3

(d) Program stored in <u>main</u> memory;
A RAM/IAS
R ROM
Instructions fetched and executed by processor *(concept)*;
A CPU
Can be replaced by another program;
R cache

2

(e) (i) The number of 1s (including the parity bit) comes to an even number;

1

(ii) Used to check for errors when data is read / transferred;
Parity bit regenerated / recalculated;
Compared with parity bit;

Any 2

2

[12]

Q19.

(a) EBCDIC/EBCD;
ASCII;
UNICODE;
A minor spelling variations

Any 2

2

(b) (i)

X	Index	Result			
		[3]	[2]	[1]	
835	0	–	–	–	
83	1	–	–	53	
8	2	–	51		
0	3	56			

1 mark for each correct entry

6

(ii) Convert a number into its character codes;

1

[9]

Q20.

(a) 1024 / 2^{10};
A 10000000000_2 *(10 0's)*

1

(b) (i) 1111111111111111; *(16 1's)*

A FFFF;
A 65,53<u>5</u> / 2^{16}<u>-1</u>;

1

(ii) 0000 0000 0010 0101; *accept if leading zeros not given*

1

(c) (i) 0011 0011 1011 0111;;; *accept 37 transposed*: 1011 0111 0011 0011;;;

1 mark for parity bits - one mark for each correct character code

f.t. for parity bits: if even number of 1's in each byte ;

3

(ii) Parity bit is set when character first generated;
(Parity bit is adjusted to make) number of 1's /on-bits even;
Parity bit is regenerated / the number of 1's is checked by receiver;
If parity bit does not match / if there are an odd number of 1's an error has occurred;

2

[8]

Q21.

(a) (i) 23;

1

(b) (i) 1010 0001;;

1 mark for correct ASCII code, one mark for odd parity bit (follow through)

2

(ii) 11010 00010 *OR* 01010 00011 *OR*

Allow stop bit to be 1 or 0 but stop and start bits must be different
Follow through if (i) wrong

01000 01011 *OR* 11000 01010;

Allow both ways round for transmission

1

[4]

Q22.

(a) (i) 54;

1

(b) (i) '4' / 4 ; ;

1 mark for ASCII value 52; 2 marks for correct character 4 ; ;

Max 2

(ii) UNICODE / EBCDIC / EBCD /extended binary coded decimal ;
A minor misspelling of EBCDIC

1

(c) (i) B<u>it-mapped</u> graphic;

R as pixels
R jpeg *etc*

1

(ii) Image broken down into separate pixels;
Each pixel is either black or white / on or off;
Use 2 different values for black and white / 1 for black and 0 for white (or vice versa);
Store in one bit / bits / byte of <u>computer memory</u>;
A diagram which maps onto above points
A follow through from (i) a .gif or .jpeg image:

Max 2

[7]

Q23.

(a) (Parity bit is adjusted to make) number of 1's / on/off bits even;
Parity bit is regenerated/checked by receiver;
Check parity bit after transmission;
If number of 1-bits is now odd, there was an error in transmission;

Max 3

(b) The greater the bandwidth the greater the rate at which data can be sent;
Bit rate increases as bandwidth increases;
Bit rate (directly) proportional to bandwidth;

Max 1

[4]

Q24.

(a) 0000 0000 0001 1001;

Note: possible use of misprinted scripts: if answers are in the right boxes mark as above. If marks are against the marks allocated, interpret 1st answer as pure binary and 2nd answer as BCD.

1

(b) (i) 53;

1

(ii) 0011 0010; 0011 0101;

2

[4]

Q25.

(a) (i) Bits are sent one after another / bits are sent one at a time/ bit by bit / bits sent singly / bits along a single wire / line; **R** data

1

(ii) Bits transferred simultaneously / concurrently / bits sent down many wires at the same time; data is sent one word/byte at a time;
R sending blocks of data
R down 2 wires

1

(b) Speed of (serial) transmission/rate at which data is sent; the number of

signal/voltage changes per second; **A** units of time one baud is (approx) one bits per second;
R rate of data transmission

1

(c) (i) 1 0 1 0 0 1 1 0

ASCII

1 mark for correct ASCII binary code in 6 rightmost bits
1 mark for even number of 1s in 8 bits

2

(ii) 0 0 1 0 1 0 0 1 1 0 1
OR
0 1 0 1 0 0 1 1 0 1 1;
OR
1 1 1 0 1 0 0 1 1 0 0
OR
1 1 0 1 0 0 1 1 0 0 0;

1

[6]

Q26.

(a) A group of bits representing a single character / (usually) 8 bits; 1

(b) (i) A 1

(ii) 9 1

(c)

Code	Number
–	0
55	7
51	73
50	732
49	7321

given

1 mark for each correct entry

6

[9]

Q27.

(a) Optical Mark Recognition/Reading. (Not Optical mark reader) **1**

(b) Extra digit added to the transaction code (1)
To detect if data has been corrupted (1) **2**

(c) (i) Unique field of a record/filed used to identify record **1**

(ii) Transaction code **1**

(iii) Not indexed sequential / Serial or Sequential (1)
Because all the records have to be examined (1)

Or

Direct access based on a hash code of the chosen numbers(1)
Only a few records will need to be checked (when collisions occurred)(1) **Max 2**

(iv) Random or direct access(1)
Record can be located by simple transformation of transaction code /hashing technique used/algorithm used to store and retrieve records(1)
Indexed sequential with transaction code as key field(1)
Rapid access via the index is possible to find the necessary record(1) **Max 2**

(d) *Any 4 points × 1 each*
Ticket scanned/ Read ticket
Check digit used to check accuracy of scanning Ticket validated,
(e.g. not out-of-date, draw not yet made)
Operator informed if ticket does not scan/is invalid
Transaction code sent to central computer
Correct file selected
Ticket's record found/Look up ticket's record/Look up record with given transaction code
Get draw date from transaction record
Get numbers from system (for the correct draw date)
Ticket numbers checked against draw
If a winning ticket prize money determined
Result sent to point of sale machine
Result displayed at point of sale machine **Max 4**

[13]

Q28.

(a) Part of a program
Numeric
Graphics data
Sound data
Address of memory location
Textual data
Logical data
Characters
Binary integers

Floating point numbers
BCD numbers

1 mark per clearly <u>distinct</u> interpretation to max 3

3

(b) The program instructs it to take the contents of that location and use it accordingly

1

[4]

Q2.

(a) This part was answered very well by candidates with the majority responding with keyboard and touch-sensitive screen as manual input methods for a barcode. A few candidates responded with mouse as one of their devices and this was not accepted as this would not be an appropriate input device, with the context of a supermarket checkout. Weaker responses also included a variety of scanning devices and just plain monitor / VDU.

(b) This part provided candidates with a good opportunity to demonstrate an understanding of the principles of operation of a bar-code scanner. It was pleasing to see candidates structure their work well and provide clear statements about a light source, the reflection of light, the sensing of the reflection and the conversion of this into an appropriate digital form.
The role of the check digit was less well known and this separated the candidates by marks achieved. It was pleasing to see that candidates could describe how a function would be applied to the first 12 digits to generate a digit which would then be compared to the original check digit. For some candidates there was confusion as to what a check digit actually was and a few decided to talk about parity and the numbers of 1s and 0s in the binary. Weaker candidates spent time discussing the looking up of a product in a database to find the details and also sometimes included some discussion around stock control.

Q3.

Overall, students demonstrated a good understanding of the use of regular expressions.

(a) For this part, students needed to explain that the expression would accept any string that consisted of zero or more b characters followed by a c. Common mistakes were to confuse the + and * operators and therefore to state that one or more b characters would be required, and to apply the * operator to the c to its right rather than the b to its left. Some students wrote “any number of b characters” which was not enough for a mark. It had to be made clear that zero or more characters were required to achieve the mark.

(b) For this part, students needed to explain that the expression would accept any string that consisted of zero or one b characters followed by a c. Many candidates simply listed the two strings that would be accepted, which were bc and b, which was an acceptable alternative response. Some candidates confused the meaning of the ? operator with either the + or the | operators.

Q4.

Part (a) was generally well-answered. Part (b) was a question that has been asked before (in the 2010 COMP1 exam) but answers remain quite vague with the role of the receiver being particularly poorly described. The questions about Hamming code were answered reasonably well, with more students getting marks for these questions than on the part (b) question about parity bits. Many students were able to give an advantage of Hamming code although, as on the 2012 COMP1 exam, answers were occasionally too vague, such as “It can detect errors”. There were some students who clearly had no understanding of the topic and were just guessing, for example, “It uses less memory."

Q5.

This question was generally well-answered. For part (a), some students did not use the number of bits specified in the question and some used even parity instead of odd parity. Part (b) was the first COMP1 question about Hamming code. Many students were able to give an advantage of Hamming code although occasionally answers were too vague, eg, "It can detect errors" and there were some students who clearly had no understanding of the topic and were just guessing eg, "It uses less memory."

Q6.

This was the first question that had been asked about Gray codes and it was clear that most candidates have had little exposure to this topic. Few candidates were able to get all the marks on parts (a-c). For part (d), the most common correct answer was that Gray code counters use less electrical power. A number of candidates simply stated that only 1 bit changes at a time; this is true but does not answer the question which was to give an advantage of Gray code counters. A common incorrect answer was to state that Gray code counters use fewer bits (they would use the same number of bits), that they are more efficient (not a precise enough answer) and that they are easier to read.

Q7.

There was a wide range in the quality of answers given to this question.
For part (a), many candidates correctly stated that 128 different characters could be represented. The most common incorrect answer to give the highest value that can be represented using 7 bits (127) rather than the number of values that can be represented. Other commonly seen incorrect answers were 256, 255, 65 and 64.

Part (b) was generally well answered, with the most common error arising from candidates not reading the question carefully and using more than 7 bits in their answer. The incorrect number of bits were also sometimes used for part (c) and some candidates gave the ASCII character for 'B' rather than 'A' -again indicating that the question had not been read carefully. Another incorrect answer seen quite often for parts (b) and (c) gave responses as decimal values (even though the number of bits to be used was clearly indicated).

For part (c), a significant number of candidates only had a vague understanding of the parity system and their answers lacked detail and used incorrect terminology. A common misconception was that if the data received had an even number of 1s in it then the data was correct – if an even number of errors have occurred during transmission this would not be true. Quite a few candidates had obviously prepared for a question on Hamming Codes and wrote about this in their answers; others described the odd parity system. Good answers described clearly the roles of both the sender and the receiver.

Q8.

This question was often answered by describing the entire system including the database lookup and printing the till roll which was clearly not what the question was asking. Candidates should read the question carefully before starting to answer. The question was quite clear in that it asked about the bar code reader and even gave an end point to the process stating, “excluding the use of the check digit.” Many answers concerned general stock control and automatic ordering systems which are more of an ICT-type answer than one expected from Computing candidates. A good number of candidates thought that the reader played some role in the security system somehow switching off RFID tags in the goods that were being scanned. At the other extreme, there were also many answers gaining full marks for very cogent descriptions.

Part (b) was answered quite weakly and approximately 15% of candidates did not attempt to answer it at all. Where the question was answered it was often incorrect with candidates believing that the check digit was the whole of the printed number below the bar code and its purpose was to allow the cashier to enter the number if the bar code had not scanned correctly. The check digit is in fact used to validate the input of the bar code. The idea of deactivating RFID tags also appeared here quite often.

Q9.

(a) Most candidates were able to score the one mark, although a common suggestion was to use the line to send additional data, or to send data when the other eight lines were not working.

(b) Most candidates were able to score marks in part (iii). Owing to a possible ambiguity in the use of the lines, a variety of bit patterns scored marks and most candidates appreciated the significance of parallel transfer in the positioning of the bits on the diagram.

(c) Generally well answered.

(d) (ii) Although presented in this way with a figure to write on for the first time, many candidates scored the 2 marks.

(iii) Better answers were seen than on previous question papers, although some candidates wrongly framed their answer around the concept of the 'speed' of transfer.

Q10.

Parts (a) (b) and (c) were often poorly answered which is at odds with the general level of answers seen on previous papers. Was it because the computations were put into a context that students were unable to identify with?

(d) (i) Some candidates did not read the rubric and wrongly gave an 8-bit code.

(ii) Again the candidates' communication skills often let them down. Their answer was often sufficient to suggest to the examiner that the candidate probably did know how parity worked but they were unable to express the key parts of the process – there were many points the candidate could have described to secure the two available marks.

Q11.

(a) The majority of candidates scored the full 3 marks.

(b) (i) A surprising number of candidates did not score marks on this question. There were many different wrong answers including, for example, “Ultra Slim Build” and “Uniform Byte Synchroniser”.

(ii) Most correctly stated parallel.

(iii) There were a variety of ways the candidate could score the 2 marks. For example, by focussing on the word, protocol and describing this as a set of rules for communication. The most common answers gave particular signals which are exchanged between the two devices.

(iv) There were few correct answers seen here despite an exhaustive list of possibilities on the mark scheme. Many candidates confused this scenario with the use of signals on the control bus of the motherboard.

(v) This was poorly answered. The vague term 'printer software' was not considered acceptable. Printer driver was common from the stronger candidates together with "word processing software" or even "the applications program from which the document is being printed". Few mentioned the operating system. Other common wrong answers were the suggestion that the data file to be printed was software, or describing the ASCII code table (referring back to part (a) of the question) as software.

Q12.

Most candidates gave two out of the three expected answers for part (a) of lower case characters, digits and symbols or punctuation characters. Those who gave valid examples of these were also credited. Incorrect answers included control characters, despite part (b), and formatting terms such as bold or italics. Fewer candidates were able to suggest two examples of control characters which might appear in a text file.

Q13.

(a) Most candidates were able to write the correct binary pattern and successfully add a correct parity bit. Candidates need reminding that it is normal practice to use bit position 7 (the left-most bit) for the parity bit.

(b) (i) Most candidates wrote character 'D'.

(ii) Some candidates left the answer as 74 failing to use the Chr( ) function as the final step.

(c) All candidates scored well for the structure chart appreciating that such a diagram can be used to represent the interface of a function. The mark scheme was generous in allowing a mixture of case used in the identifier labels in the diagram. This might not gain credit in future papers.

(d) This question was well answered by the majority of candidates - possibly as there had been a 'build up' to the algorithm in the earlier parts of the question.

This is an encouraging trend as the algorithm contained a loop, the use of functions, the use of arrays (which had been poorly tackled on previous papers), yet candidates scored highly.

Q14.

(a) Due to an error in the question paper for part (i), examiners were instructed not to mark this part question. The total mark for the paper was therefore reduced to 64. Despite a different style of question for part (ii), the vast majority of candidates were clear as to what was required. Common errors were a diagram which showed 9 lines instead of 8 or the lines drawn as 'tubes'. Some candidates drew pulses on the lines instead of labelling each one with a 1 or 0 and still scored both marks.

(b) This was surprisingly badly answered with candidates either not clear as to what was being asked for, or not understanding the phrase "data representation". From past examination paper responses this appears to have been well understood previously. If the candidate has covered the specification content fully then it is hoped that students would be fascinated to learn how the same binary number can be interpreted in so many different ways: a basic machine code instruction to add two numbers together, a musical note, the colour of a pixel, one of the characters in a text string, one of several different number types etc.

Q15.

Most candidates gained the first two marks available for this question but few gave a creditable answer to the third point. The label was a barcode, which would be read by a barcode reader or bar code scanner. ‘Scanner’ on its own was deemed insufficient. A few candidates suggested the use of an OMR. The advantages of having the label read by an input device rather than the code being typed in by the shop assistant were speed and accuracy. This was one of the very few occasions when it was decided to give credit to a one-word answer such as quicker or faster.

Candidates were then asked to give one advantage that a bar code has over a character code that makes it suitable for the identification of items in many different situations. Some candidates repeated their answers to part (b)(i), which suggested that they had not read the question carefully. Better candidates knew that the two main reasons for using bar codes rather than character codes are that bar codes can be read from any angle, unlike character codes, and can often be read even though partly damaged.

Q16.

Candidates rarely obtained good marks on this question. It would appear that they had a reasonable knowledge of pure binary but many candidates failed to show any understanding of other coding systems.

Many candidates failed to obtain credit by leaving boxes empty.

(a) There were few correct answers to part (i). The majority of candidates were able to provide the correct answer to part (ii).

(b) Candidates showed very little understanding of Unicode. Some of those that seemed to realise that Unicode uses sixteen bits were unable to express this satisfactorily.

(c) Most candidates were able to provide the correct answer to the pure binary value.

Q17.

(a) Many candidates were able to work out the correct answer to this part showing that they have an understanding of data encoding.

(b) Candidates who answered part (i) generally gave the correct result. ASCII was sometimes given for part (ii) showing that the candidate had not read the question. There continues to be a problem with the spelling of EBCDIC. This is a technical term that should be understood by the candidates and there is no guarantee that misspellings will be given credit in the future.

(c) It was disappointing to see how many candidates were unable to answer this part satisfactorily.

(i) Candidates should be aware that each pixel is stored separately in bit-mapped graphics. Although many candidates stated that the colour would have to be stored, few were able to explain how. A common misconception was that one bit could store a range of numbers.

(ii) There was even less understanding shown of vector graphics. Candidates should appreciate what needs to be stored. Stating that the line would be stored as an equation is insufficient.

Q18.

(a) Good marks were obtained on this part of the question but there were a surprising number of candidates who could not state the full names correctly.

(b) The term hard disk controller was not well known. Candidates were more aware of the network interface card. This was one situation where a number of candidates failed to score by giving a brand name as their response.

(c) Another high scoring part although a large number of candidates only obtained two marks. Both the data and address buses were well known but the control bus was often mistakenly given as either a system bus or a serial bus.

(d) This was not answered well. Candidates often failed to obtain credit by weak description, for example “memory” could apply to a number of different methods of storage. It was important that the candidates made it clear that programs must be in the main memory from where they will be fetched and executed by the processor.

(e) Parity concepts do not seem to be well known by the candidates. Few candidates were able to explain what is meant by even parity and even fewer were able to explain how the computer system might use the parity bits.

Q19.

(a) Most candidates gave ASCII and many were able to state either EBCDIC or Unicode. A common error was to give BCD.

(b) It was pleasing to see that few candidates were unable to get some of the trace table correct. The Index entries were nearly always correct. Most candidates also obtained the correct entries for X. Result [2] was often correct but Result [3] was often incorrect. Unfortunately few candidates were able to express the purpose of the algorithm.

Q20.

A significant number of candidates do not know that 1024 bytes make 1 Kilobyte. Even fewer candidates could state correctly that the largest pure binary integer that can be stored in 2 bytes is 65,535 (or 1111 1111 1111 1111 1111). Incorrect responses ranged from as low as 3 to many thousands.

The bit pattern asked for was largely well answered, but candidates should be made aware that leading zeros are required when bit patterns to a specified length are asked for. The whole purpose of binary is that only two states, 0 and 1, can exist.

The concept of parity checking eludes many candidates. Few could explain how a computer system would use a parity bit. The parity bit is set when the character is first generated, by adjusting the parity bit to make the number of 1s even (for even parity). Then the parity is checked at the receiving end and if the parity is now odd an error has occurred. The question clearly stated that in the given computer system the parity bit was the most significant bit of each byte. However, many candidates only looked at the parity across the whole 16 bits. Many candidates were not able to translate the characters 3 and 7 into ASCII codes with the code ranges for digits explicitly provided by the question.

Q21.

Nearly all candidates obtained some marks for this question.

(a) Almost all candidates could convert from ‘Pure Binary’.

(b) Nearly all candidates gave the correct bit pattern for 33 but a large number failed to get the parity bit correct (the question setter had ensured that the parity bit had to be set to ‘1’). In part (ii) few candidates realised that the start and stop bits needed to be different and in some cases these were simply left blank. When a binary pattern is asked for, all places must be filled in with either a 0 or a 1. The parity bit must not change; just because start and stop bits are added. These will be stripped off before parity checks are performed at the receiving end.

Q22.

Nearly all candidates scored some marks on this question.

(a) (i) Nearly all gave 54.

(b) (i) The majority correctly identified 4 as the encoded character.

(ii) Depending on the centre, Unicode or EBCDIC were the correct answers given (with some highly original spellings of EBCDIC), while ‘encryption’ and ‘hexadecimal’ were very popular incorrect answers.

(c) (i) Most gained credit with ‘bitmap’ as their answer.

(ii) Nearly all candidates gained at least one mark but many ignored the fact in the question stem that a black-and-white image was to be stored and went into details about storing coloured images. Resolution was also often described which was not asked for here. The description that the image is broken down into pixels, and these are either black or white, that a one could be stored for each white pixel and a nought for each black pixel or vice versa would have gained full marks.

Q23.

(a) Most candidates gained at least one mark for stating that with even parity the parity bit is set so that the number of 1s in the byte is even. However, few candidates had a clear idea about parity, many assuming that parity guaranteed that an ASCII character was correct. The process is in three parts (hence the three marks).
(1) The parity bit is generated and attached to the other seven bits.
(2) The code received goes through the same process and compares the parity bits and if they differ, then (3) an error has occurred in the transmission and a retransmission is requested.

(b) Many candidates failed to read the question which asked for the ‘relationship’ between ‘bit-rate’ and ‘bandwidth’ and simply wrote definitions of the two terms. Credit was given for responses such as ‘bit rate increases as bandwidth increases’.

Q24.

(a) The majority of candidates correctly converted 25 to pure binary integer. However, when asked how this would be stored in a 16-bit word, candidates were expected to write down leading zeros to make up 16 bits. To help candidates with this, a box to complete the bit pattern was provided.

(b) (i) Even with the clue in the question a large number of the candidature did not give 53 as the ASCII code for ‘5’. A very common wrong answer was 85.

(ii) Few included the pair of 11s in the left hand nibble of each number character. The right hand side of each pattern was correct but obviously incomplete, i.e. 0000 0010 0000 0101. Again, a similar question had been set in previous papers.

Q25.

(a) Some candidates confused serial and parallel transmission with simplex and duplex (not in the specification of CPT I). Many candidates just referred to data being transmitted one after the other and therefore did not gain credit. In serial transmission bits are sent one at a time along a single wire and in parallel transmission several bits are transferred simultaneously down several wires.

(b) Baud rate was mostly defined adequately as bits per second. Turning the question into the answer "rate of data transmission" is not worthy of credit. Some quality answers referred to the number of state changes of a signal in a second.

(c) Most candidates correctly converted 38 into the binary pattern 0100110, but many candidates do not appear to know that the most significant bit is on the left and therefore the 8-bit pattern requested should read as 10100110. Many answers did not show eight bits as required. A pleasing number of candidates understood correctly that start and stop bits need to be different in asynchronous data transmission and therefore the bit pattern should be prefixed with 1 and 0 and suffixed with 00 or 11 respectively. Credit was given for the start bit at either end, as long as the stop bits were at the other end. Some candidates got so carried away with the start and stop bits that they forgot about the bit pattern for the character in between. Worrying was the fact that some candidates used symbols such as * or # as alternatives to ones and zeros.

Q26.

(a) Most candidates knew that a byte was 8 bits. Some quoted January's paper again and (wrongly) stated that a kilobyte was 1000 bytes.

(b) The majority of candidates correctly converted the bit patterns to 'A' and '9', but a significant minority quoted the ranges given in the question.

(c) Most candidates who attempted the dry run correctly calculated the codes, but only the better candidates noticed that Number grew in size with each digit processed.

Q27.

Many candidates correctly identified the method as Optical Mark Recognition. Candidates who answered "Optical Mark Reader" referred to the device, not the method, and so were not rewarded. Some candidates answered incorrectly that a check digit checked that the chosen numbers were unique. An answer that stated that a check digit is an extra digit added to the transaction code obtained a mark. The mark scheme allocated a second mark to an answer that stated that the check digit was used to detect if data was corrupted. The emphasis was on error detection, hence the non-specificity in stating what was being corrupted. Several candidates went into detail and described how the check digit is calculated using a modulo- II method. This was really answering more than was required for one mark, as this response described both what it is and how it is generated.

The majority of candidates correctly defined the term 'primary key'. However, several of these candidates then incorrectly identified "Point of Sale Identification Code" as the primary key for the transaction records instead of "Transaction Code". These candidates appeared to lack an understanding of the term 'transaction'.

In part (c) (iii) the better candidates realised that all the records have to be examined to find the ticket(s) with the winning numbers. These candidates then showed good knowledge of file organisations by answering “serial”. Weaker candidates seemed to be unfamiliar with the term ‘file Organisation’, responding with answers such as “use a database or spreadsheet”. Other successful candidates answered that if each transaction’s chosen numbers were hashed, then the generated address could be used to store each transaction’s details. After the draw, the winning numbers could be hashed to locate the transaction(s) that had won.

In part (d) many candidates did not appreciate the detail of the processing steps that have to take place if the computing system is to check if the ticket is a winning ticket. These candidates showed a distinct lack of insight into the operation of the computer. Their answers were superficial and from the perspective of the ticket holder not the computer system. The better answers were on a different analytical plain. These answers used appropriate technical terms e.g. ticked “scanned”, “check digit” used to check accuracy of “scanning”, “transaction code” sent to “central computer”, correct “file” located/ “record” with given “transaction code” found or “transaction code” compared with winning “transaction codes”, etc. Compare this with an answer pitched at the level of a ticket holder. “The numbers on the ticket are entered. The computer system checks if these are the winning numbers. The user is informed”. Such an answer gained no marks.

Q28.

Candidates gained no marks for offering ‘binary’ as an interpretation, as all data in the location would be in binary. Hexadecimal and Octal were incorrect alternatives that were seen too often. ‘A binary number’ or ‘integer’ followed by ‘twos complement’ were not considered distinct answers. Few candidates were then able to give a sensible answer to part (b).