5.4 Binary number system part 1 Mark Scheme

Q1.

(a) **Mark is AO2 (apply);**

30A;

R. More than one lozenge shaded

1

(b) **All marks AO2 (apply)**

0 •	1	0	1	1	0	0

Mantissa

1	1	1	0	1

Exponent

1 method mark for either:

- showing correct value of both mantissa and exponent in decimal (mantissa = 0.6875 // 11/16, Exponent = −3)
- showing binary point shifted 3 places to left in binary number
- indicating that final answer calculated using answer = mantissa × $2^{exponent}$

1 mark for correct answer

Answer = 0.0859375 // 11/128

If answer is correct and some working has been shown, award two marks, even if working would not have gained credit on its own.

2

[3]

Q2.

All marks AO2 (apply)

2 marks for working:

Correct (unsigned) representation of 608 in binary: 1001100000; **A.** leading 0s
Correct representation of -608 in two’s complement binary:10110100000; **A.** leading 1s

Showing the correct value of the exponent in decimal (10) or binary (1010) // showing the binary point being shifted 10 places left;

Max 2
1 mark for correct mantissa and exponent together:

1 •	0	1	1	0	1	0

Mantissa

0	1	0	1	0

Exponent

If answer is correct and some working has been shown, award three marks, even if working would not have gained credit on its own.

Working marks can be awarded for work seen in the final answer eg correct exponent.

[3]

Q3.

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

C;

1

(b) **Mark is AO1 (understanding)**

B;

1

(c) **All marks AO2 (apply)**

1 •	0	1	1	0	0	0	0

Mantissa

0	0	1	1

Exponent

1 method mark for either:

- showing correct value of both mantissa and exponent in denary (mantissa = -0.625 // -5/8, Exponent = 3)
- showing binary point shifted 3 places to right in binary number ie 1011.0000 or in the positive equivalent 0101.0000
- indicating that final answer calculated using answer = mantissa x $2^{exponent}$

1 mark for correct answer

Answer = −5

If answer is correct and some working has been shown, award two marks, even if working would not have gained credit on its own.

2

(d) **All marks AO2 (apply)**

2 marks for working:

Correct representation of 58.5 in fixed point binary: 111010.1; **A**. leading 0s.
Showing the correct value of the exponent in denary (6) or binary (110) // showing the binary point being shifted 6 places;

MAX 2

1 mark for correct mantissa and exponent together:

0 ●	1	1	1	0	1	0	1

Mantissa

0	1	1	0

Exponent

If answer is correct and some working has been shown, award three marks, even if working would not have gained credit on its own.

Working marks can be awarded for work seen in the final answer eg correct exponent.

3

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

0.05 // 13.8 – 13.75;

A. Award BOD mark if correct method has been shown i.e. 13.8 – 13.75 but candidate has then made an error performing the subtraction operation
R. −0.05 unless the accept point above also applies

1

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

0.36(%);

A. 0.0036 // 0.05 ÷ 13.8
A. Follow-through of incorrect answer to question part 11.5
A. Award BOD mark if correct method has been shown but candidate has then made an error performing the division operation

1

[9]

Q4.

		Answer	Carry	
0	0	0	0	
0	1	1	0	;
1	0	1	0	;
1	1	0	1	;

[3]

Q5.

(a) **1 mark** per correct answer:

Value description	Correct letter (A-D)
A negative value.	A;
The largest positive number of the four values.	C;

A value that is not valid in the representation because it is not normalised.	B;

If a letter is used more than once then mark as correct in the position that it is correct (if any).

3

(b)

0	●	1	0	1	1	0	1

Mantissa

0	0	1	0	0

Exponent

1 method mark for either:

- showing correct value of both mantissa and exponent in denary (Mantissa = 0.703125 // 45/64, Exponent = 4)
- showing binary point shifted 4 places to right in binary number
- indicating that final answer calculated using answer = mantissa x $2^{exponent}$

1 mark for correct answer

Answer = 11 1/4 // 11.25 // 45/4

If answer is correct and some working has been shown, award two marks, even if working would not have gained credit on its own.

2

(c)

0	●	1	0	1	0	1	0

Mantissa

1	1	1	0	1

Exponent

1 method mark for either:

- showing correct value of both mantissa and exponent in denary (Mantissa = 0.65625 // 21/32, Exponent = −3)
- showing binary point shifted 3 places to left in binary number
- indicating that final answer calculated using answer = mantissa x $2^{exponent}$

1 mark for correct answer

Answer = 21/256, 0.08203125 **A**. Rounded to at least 2dp

If answer is correct and some working has been shown, award two marks, even if working would not have gained credit on its own.

2

(d) **2 marks** for working:

Correct representation of 3008 in fixed point binary: 101111000000;
A. leading 0s.
Showing the correct value of the exponent in denary (12) or binary (1100) // showing the binary point being shifted 12 places;
Showing the correct value of the mantissa in binary: 0.101111; **A**. leading and trailing 0s.

MAX 2

1 mark for correct mantissa and exponent together:

0 ●	1	0	1	1	1	1

Mantissa

0	1	1	0	0

Exponent

If answer is correct and some working has been shown, award three marks, even if working would not have gained credit on its own.

Working marks can be awarded for work seen in the final answer eg correct exponent.

3

(e) 0.08 // 12.83 - 12.75 // 2/25
R. −0.08

A. Award **BOD** mark if correct method has been shown i.e. 12.83 − 12.75 but candidate has then made an error performing the subtraction operation

1

(f) 0.6235%
A. 0.006235 // 0.0062 // 0.08 ÷ 12.83
A. Follow-through of incorrect answer to part (e)
A. Award **BOD** mark if correct method has been shown but candidate has then made an error performing the division operation

1

(g) **Definition (2 marks):**
The result of a calculation is too large to store/represent // a number is too large to store/represent;
In the available number of bits / storage space (allow example eg data type, byte, word, example of a data type); **R.** Space **NE**

Example (1 mark):
Multiplying two numbers together;
Dividing a number by a number less than one / small number;
R. Zero
A. Adding two numbers (of same sign)
A. When number converted from one type to another that does not have suitable range/enough bits/enough storage space to represent it
A. Answers by example
MAX 1

3

[15]

Q6.

(a)

1 ●	0	0	0	0	0	0

Mantissa

0	1	1	1	1

Exponent

1 mark for correct mantissa
1 mark for correct exponent

2

(b)

1	• 0	1	0	1	1	0	

Mantissa

0	1	0	1	1

Exponent

1 method mark for either:

- showing correct value of both mantissa and exponent in denary (Mantissa = -0.65625 // -21/32, Exponent = 11)
- showing binary point shifted 11 places to right in binary number
- indicating that final answer calculated using answer = mantissa x 2^{exponent}

1 mark for correct answer

Answer = -1344

If answer is correct and some working has been shown, award two marks, even if working would not have gained credit on its own.

2

(c) **2 marks** for working:

Correct representation of 12¾ in fixed point binary: 1100.11;
A. any number of preceding 0s or succeeding 0s
Showing the correct value of the exponent in denary (4) or binary (100) // showing the binary point being shifted 4 places left;
Showing the correct value of the mantissa in floating point binary: 0.110011;

MAX 2

1 mark for correct mantissa and exponent together:

0	• 1	1	0	0	1	1

Mantissa

0	0	1	0	0

Exponent

If answer is correct and some working has been shown, award three marks, even if working would not have gained credit on its own.

Marks for working can be awarded in the answer e.g. correct mantissa and exponent

3

(d) (i) Maximises precision/accuracy for given number of bits;
Note: Must have concept of given number of bits or an example of this e.g. word length.
Unique representation of each number // simpler to test for equality of numbers;

2

(ii) There is no need to store both the 0 and the 1 // only one of the two bits needs to be stored (as one can be

inferred from / worked out from the other);
Therefore another bit could be freed up to use in <u>the mantissa</u> // one extra bit of precision could be stored;
A. A description that clearly implies the mantissa if mantissa is not explicitly stated

2

[11]

Q7.

(a) **All marks AO2 (apply)**

1 mark for working: conversion of D to 13 or multiplication of a number (even if not 13) by 16 and adding 6 to the result;
1 mark for answer: 214;

2

(b) **All marks AO2 (apply)**

1001; 0110;
1 mark: correct first four bits
1 mark: correct bits in position 5 – 8

2

(c) **All marks AO2 (apply)**

1;0111101;
2 marks: Correct answer only

2

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

10101011;

1

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

The result is too large to be represented;
(it causes) overflow;
The result represents a negative value;
Max 1 mark

1

[8]

Q8.

(a) **All marks AO1 (understanding)**

1 mark per correct response:

Value description	Correct letter (A-D)
A positive normalised value.	A
The most negative value that can be represented.	C

A value that is not valid in the representation because it is not normalised.	B

If a letter is used more than once then mark as correct in the position where it is correct (if any).

3

(b) **All marks AO2 (apply)**

0 ●	1	0	1	1	0	0	0

Mantissa

0	1	0	1

Exponent

1 method mark for either:

- showing correct value of both mantissa and exponent in denary (Mantissa = 0.6875 / / 11 / 16, Exponent = 5)
- showing binary point shifted 5 places to right in binary number
- indicating that final answer calculated using answer = mantissa x $2^{exponent}$

1 mark for correct answer

Answer = 22

If answer is correct and some working has been shown, award two marks, even if working would not have gained credit on its own.

2

(c) **All marks AO2 (apply)**

2 marks for working:

Correct representation of 6.75 in fixed point binary:
110.11; **A** leading 0s.
Correct representation of -6.75 in two's complement fixed point binary:
1001.01; **A** leading 1s.
Showing the correct value of the exponent in denary (3) or binary (11) / / showing the binary point being shifted 3 places;
Max 2

1 mark for correct mantissa and exponent together:

1 ●	0	0	1	0	1	0	0

Mantissa

0	0	1	1

Exponent

If answer is correct and some working has been shown, award three marks, even if working would not have gained credit on its own.

Working marks can be awarded for work seen in the final answer eg correct exponent.

3

(d) **All marks AO1 (understanding)**

1 mark: Reduced precision;
1 mark: Increased range; **A** can represent larger / smaller numbers

2

[10]

Q9.

(a) **1 mark** per correct answer:

Value description	Correct letter (A-D)
A negative value.	C;
The largest positive number of the four values.	B;
A value that is not valid in the representation because it is not normalised.	A;

If a letter is used more than once then mark as correct in the position that it is correct (if any).

3

(b)

0	• 1	1	0	1	0	1

Mantissa

0	0	1	0	0

Exponent

1 method mark for either:

- showing correct value of both mantissa and exponent in denary (Mantissa = 0.828125 // 53 / 64, Exponent = 4)
- showing binary point shifted 4 places to right in binary number
- indicating that final answer calculated using answer = mantissa × 2^{exponent}

1 mark for correct answer

Answer = 13 1 / 4 // 13.25 // 53 / 4

If answer is correct and some working has been shown, award two marks, even if working would not have gained credit on its own.

2

(c)

0	• 1	0	0	1	1	0

Mantissa

1	1	1	0	1

Exponent

1 method mark for either:

- showing correct value of both mantissa and exponent in denary (Mantissa = 0.59375 // 19 / 32, Exponent = -3)

- showing binary point shifted 3 places to left in binary number
- indicating that final answer calculated using answer = mantissa × $2^{exponent}$

1 mark for correct answer

Answer = 19 / 256, 0.07421875
A rounded to at least 2dp

If answer is correct and some working has been shown, award two marks, even if working would not have gained credit on its own.

2

(d) **2 marks** for working:

Correct representation of 2944 in fixed point binary: 101110000000;
A leading 0s.
Showing the correct value of the exponent in denary (12) or binary (1100) // showing the binary point being shifted 12 places;
Showing the correct value of the mantissa in binary: 0.101110;
A leading and trailing 0s.

MAX 2

1 mark for correct mantissa and exponent together:

0 •	1	0	1	1	1	0

Mantissa

0	1	1	0	0

Exponent

If answer is correct and some working has been shown, award three marks, even if working would not have gained credit on its own.

Working marks can be awarded for work seen in the final answer e.g. correct exponent.

3

(e) (i) 0.12 // 12.87-12.75 // 3 / 25
R -0.12
A Award **BOD** mark if correct method has been shown i.e. 12.87 - 12.75 but candidate has then made an error performing the subtraction operation

1

(ii) 0.9324%
A 0.009324 // 0.0093 // 0.12÷12.87
A 4 / 429
A Follow-through of incorrect answer to part (i)
A Award **BOD** mark if correct method has been shown but candidate has then made an error performing the division operation

1

(iii) Underflow;
A Underflow error

R Stack underflow

1

[13]

Q10.

(a) 182;

1

(b) -;74;

2

(c) -128; to (+)127;
Mark as follows:
Lowest value identified correctly;
Highest value identified correctly;

2

(d) 5 11 / 16 / /
5.6875;;
A 91 ÷ 16;;

Mark as follows:
Correct whole number part (5);
Correct fractional / decimal part (11 / 16 or 0.6875);

2

(e) B;6;

2

(f) Easier for people to read / understand;
R If implication is it easier for a computer to read / understand
Can be displayed using fewer digits;
More compact when printed / displayed;
NE Takes up less space
NE More compact

MAX 1

(g) Shift all the bits one place to the left; and add a zero / /
Add an extra 0; to the RHS of the bit pattern; / /
A Arithmetic left shift applied once / by one place;;

2

[12]

Q11.

(a)

0	• 1	1	1	1	1	1	1

0	1	1	1

Mantissa Exponent

1 mark for correct mantissa
1 mark for correct exponent

2

(b)

0	• 1	0	1	1	0	0	0

0	0	1	1

Mantissa Exponent

1 method mark for either:

- showing correct value of both mantissa and exponent in denary (Mantissa = 0.6875 / / 11 / 16, Exponent = 3)
- showing binary point shifted 3 places to right in binary number
- indicating that final answer calculated using answer = mantissa x $2^{exponent}$

1 mark for correct answer

Answer = 5 ½ / / 5.5

*If answer is correct and some working has been shown, award **2 marks**, even if working would not have gained credit on its own.*

2

(c)

1 ●	0	1	1	0	0	0	0

1	1	0	0

Mantissa Exponent

1 method mark for either:

- showing correct value of both mantissa and exponent in denary (Mantissa = -0.625 / / -5 / 8, Exponent = -4)
- showing binary point shifted 4 places to left in binary number
- indicating that final answer calculated using answer = mantissa x $2^{exponent}$

1 mark for correct answer

Answer = -5 / 128,-0.0390625 **A** rounded to at least 2dp

*If answer is correct and some working has been shown, award **2 marks**, even if working would not have gained credit on its own.*

2

(d) **2 marks** for working:
Correct representation of 108 in binary: 1101100; **A** any number of preceding 0s or succeeding 0s after a binary point
Correct representation of -108: 10010100; **A** any number of preceding 1s
Showing the correct value of the exponent in denary (7) or binary (0111) / / showing the binary point being shifted 7 places;
Showing the correct value of the mantissa in floating point binary: 1.0010100;

MAX 2

1 mark for correct mantissa and exponent together:

1 ●	0	0	1	0	1	0	0

0	1	1	1

Mantissa Exponent

*If answer is correct and some working has been shown, award **3 marks**, even if working would not have gained credit on its own.*
Marks for working can be awarded in the answer.

3

(e) (i) The results of a calculation; is a number that is too large to store; in the available storage space / number of bits;
Must get the middle point (about the number being too large) to be awarded any marks.
MAX 2

2

(ii)

Operation	**May cause overflow? (Tick One)**
Subtracting a very small number from a large number.	
Dividing a large number by a very small number.	✓
Multiplying a large number by a very small number.	

A alternative symbol which clearly indicates just one box eg cross, Y, Yes
R answers in which more than one row is ticked

1

[12]

Q12.

(a) 167;;

If final answer is incorrect **Max 1** can be awarded for some correct working out being shown by the candidate:

1010 0111;
10 * 16 // 160 // A * 16;
A = 10;
Multiplying a value by 16 and adding on 7;

2

(b) 0111.1010 // 01111010

Mark as follows:
4 bits before binary point are 0111;
4 bits after binary point are 1010;

2

(c) 1;110 1110;
R if not 8 bits

2

(d) 127;

1

(e) The number to subtract is converted into a negative number;
NE Convert into two's complement
This is then added to the first number;

Two marks for example:

23 = 00010111
-48 = 11010000;

11100111; (= -25)

A if not used 8 bits in examples
A 23 + - 48 is worth 1 mark only (if there is no description)

Note: for the first mark in the example to be awarded the two bit patterns must be correct. For the second mark in the example accept an incorrect answer as long as it is a correct addition using one of the two correct bit patterns.

4

(f) 11101110;
R 01110111

1

(g) 11101011;
DPT A. 11010111

1

(h) Get the two's complement (of a positive binary value) //
Converts a positive binary value into its negative equivalent;
A It inverts all bits after the first 1 is received;

1

(i)

Input	Original State	Output	New Stat
0	S0	0	S0
1	S0	1	S1
0	S1	1	S1
1	S1	0	S1

Mark as follows:
S0 as original state for 2nd row;
1 as output for 3rd row;
Final row correct;

3

[17]

Q13.

(a) One mark per correct answer:

Value description	Correct letter (A-D)
A negative value.	D;
The smallest positive value that can be represented.	A;
A value that is not valid in the representation because it is not normalised.	C;

If a letter is used more than once then mark as correct in the position that is correct.

3

(b) **1 method mark** for either:

- showing correct value of both mantissa and exponent in denary
- showing binary point shifted 6 places to right in mantissa
- indicating that final answer calculated using answer = mantissa x $2^{exponent}$

Mantissa = 0.625 // 5/8
Exponent = 6

1 mark for correct answer

Answer = 40

If answer is correct and some working has been shown, award two marks, even if working would not have gained credit on its own.

2

(c) **2 marks** for working:

Correct representation of 7.75 in fixed point binary: 111.11;
A leading and trailing 0s.
Bits flipped: 000.00 // 1000.00; **A** leading 1s
Correct representation of -7.75 in fixed point two's complement: 1000.01;
A leading 1s
Showing the correct value of the exponent in denary (3) or binary (11) // showing the binary point being shifted 3 places;

Note: Award both working marks if bit pattern 1.00001 is shown anywhere

Max 2

1 mark for correct mantissa and exponent together

1	0	0	0	0	1	0	0

Mantissa

0	0	1	1

Exponent

If answer is correct and some working has been shown, award three marks, even if working would not have gained credit on its own.

Working marks can be awarded for work seen in the final answer e.g. correct exponent.

3

(d) (i) 0.025 // 6.9-6.875 // 1/40
R -0.025
A award **BOD** mark if correct method has been shown i.e. 6.9-6.875 but candidate has then made an error performing the subtraction operation

1

(ii) 0.003623 // 0.025/6.9 // 1/276
A 0.3623%
A answers rounded to at least two significant figures
A follow-through of incorrect answer to part (d)(i)

A award **BOD** mark if correct method has been shown but candidate has then made an error performing the division operation
R if shown that incorrect method used e.g. dividing by 6.875, even though this arrives at an answer that is the same when written to 2 significant figures

1

(iii) **Alternative 1:**
Adjust the mantissa;
To use more bits;
A "longer" for "more bits" but **R** "larger", "increase size"
Alternative 2:
Reallocate (one) bit; from the exponent to the mantissa; **A** bits
Alternative 3:
Infer one of the two bits on either side of the binary point (from the other, as they must both be different); use the freed up bit to store one more significant digit in the mantissa // use the freed up bit to represent mantissa more accurately;

2

[12]

Q14.

		Answer	Carry	
0	0	0	0	
0	1	1	0	;
1	0	1	0	;
1	1	0	1	;

A 10 instead of 0 in the Answer column for the final row of the table

[3]

Q15.

(a) Correct variable declarations for `Bit`, `Answer` and `Column`;
I additional variable declarations
`Column` initialised correctly before the start of the loop;
`Answer` initialised correctly before the start of the loop;
`While/Repeat` loop, with syntax allowed by the programming language used, after the variable initialisations; and correct condition for the termination of the loop;
R `For` loop
A any `While/Repeat` loop with logic corresponding to that in flowchart (for a loop with a condition at the start accept >=1 or >0 but reject <>0)
Correct prompt `"Enter bit value:"` ;
followed by `Bit` assigned value entered by user;
followed by `Answer` given new value;
R if incorrect value would be calculated [followed by value of `Column` divided by 2;
A multiplying by 0.5

Correct prompt and the assignment statements altering `Bit`, `Answer` and `Column` are all within the loop;
After the loop – output message followed by value of `Answer`;
I Case of variable names, player names and output messages
A Minor typos in variable names and output messages
I spacing in prompts
A answers where formatting of decimal values is included e.g.
`Writeln('Decimal value is: ', Answer : 3)`
A initialisation of variables at declaration stage
A no brackets around column * bit

Pascal

```
Program Question;
    Var
         Answer : Integer;
         Column : Integer;
         Bit : Integer;
    Begin
          Answer := 0;
          Column := 8;
          Repeat
              Writeln('Enter bit value: ');
              Readln(Bit);
              Answer := Answer + (Column * Bit);
              Column := Column DIV 2;
        Until Column < 1;
        Writeln('Decimal value is: ', Answer);
        Readln;
    End.
```

VB.NET

```
Sub Main()
   Dim Answer As Integer
   Dim Column As Integer
   Dim Bit As Integer
   Answer = 0
   Column = 8
   Do
        Console.Write("Enter bit value: ")
        Bit = Console.ReadLine
        Answer = Answer + (Column * Bit)
        Column = Column / 2
   Loop Until Column < 1
   Console.Write("Decimal value is: " & Answer)
   Console.ReadLine()
End Sub
```

Alternative Answer

```
Column = Column \ 2
```

VB6

```
Private Sub Form_Load()
   Dim Answer As Integer
   Dim Column As Integer
   Dim Bit As Integer
   Answer = 0
   Column = 8
   Do
        Bit = InputBox("Enter bit value: ")
        Answer = Answer + (Column * Bit)
        Column = Column / 2
   Loop Until Column < 1
```

```
   MsgBox ("Decimal value is: " & Answer)
End Sub
```

Alternative Answer

```
Column = Column \ 2
```

Java

```
public class Question {
   AQAConsole console=new AQAConsole();
   public Question(){
         int column;
         int answer;
         int bit;
         answer=0;
         column=8;
         do{
              console.print("Enter bit value: ");
              bit=console.readInteger("");
              answer=answer+(column*bit);
              column=column/2;
         }while(column>=1);
         console.print("Decimal value is: ");
         console.println(answer);
       }
       public static void main(String[] arrays){
          new Question();
       }
}
```

Python 2.6

```
Answer = 0
Bit = 0
Column = 8
while Column >= 1:
   print "Enter bit value: "
   # Accept: Bit = int(raw_input("Enter bit value: "))
   Bit = input()
   Answer = Answer + (Column * Bit)
   Column = Column // 2
print "Decimal value is: ", Answer
# or + str(Answer)
```


Python 3.1

```
Answer = 0
Bit = 0
Column = 8
while Column >= 1:
   print("Enter bit value: ")
   # Accept: Bit = int(input("Enter bit value: "))
   Bit = int(input())
   Answer = Answer + (Column * Bit)
   Column = Column // 2
print("Decimal value is: " + str(Answer))
# or print("Decimal value is: {0}".format(Answer))
```

A. Answer and Bit not declared at start as long as they are spelt correctly and when they are given an initial value that value is of the correct data type

11

(b) ****SCREEN CAPTURE****
Must match code from 16, including prompts on screen capture matching those in code

Mark as follows:
"Enter bit value:" + first user input of 1
'Enter bit value: ' + second user input of 1
'Enter bit value: ' + third user input of 0
'Enter bit value: ' + fourth user input of 1
Value of 13 outputted;

3

(c) 15;

1

(d) 16 // twice as many // double;

1

(e) Design;
A Planning

1

(f) Implementation;

1

[18]

Q16.

(a)

Mantissa

1 •	0	0	0	0	0	0	0

Exponent

0	1	1	1	1

1 mark for correct mantissa
1 mark for correct exponent

2

(b) *1 method mark for either:*

- showing correct value of both mantissa and exponent in denary
- showing binary point shifted 6 places to right in binary number
- indicating that final answer calculated using answer = mantissa x $2^{exponent}$

Mantissa = – 0.6875 // –11/16
Exponent = 6
Answer = –44

1 mark for correct answer
If answer is correct and some working has been shown, award two marks, even if working would not have gained credit on its own.
Marks for working can be awarded in the answer.

2

(c) *1 mark for working:*
Showing a bit pattern including 1101 and any number of preceding or following 0s, but no other 1s;
Showing the correct value of the exponent in denary (9);
Showing the binary point being shifted 9 places;

Max 1

1 mark for correct mantissa and exponent together:

0 .	1	1	0	1	0	0

Mantissa

0	1	0	0	1

Exponent

If answer is correct and some working has been shown, award two marks, even if working would not have gained credit on its own.
Marks for working can be awarded in the answer.

2

(d) *2 marks for working:*
Correct representation of 12.5 in fixed point binary: 1100.1;
Bits flipped: 0011.0 // 10011.0;
A any number of preceding 1s
Correct representation of –12.5 in fixed point twos complement: 10011.1;
A any number of preceding 1s
Showing the correct value of the exponent in denary (4) or binary // showing the binary point being shifted four places;
Showing the correct value of the mantissa in floating point binary (1.001110)

Max 2

1 mark for correct mantissa and exponent together:

1 .	0	0	1	1	1	0

Mantissa

0	0	1	0	0

Exponent

If answer is correct and some working has been shown, award three marks, even if working would not have gained credit on its own.
Marks for working can be awarded in the answer.

3

(e)

Calculation	Type of Error
Multiplying two very large numbers together.	Overflow;
Dividing a number by a very large number.	Underflow;
Adding together two numbers of very different sizes e.g. a tiny number to a very big number.	Cancellation;

If same answer is used more than once and it is correct in one instance then award the mark for the correct instance.

3

[12]

Q17.

(a) 0111 1011;

1

(b) 256 // 2^{8}; 1

(c) 7;B; 2

(d) Easier for <u>people</u> to read / understand;
(Can be displayed using) fewer digits;
More compact when printed/displayed;
NE Takes up less space
NE More compact

Max 1

[5]

Q18.

(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]

Q19.

(a)

0	•	1	0	0	0	0	0	0	1	0	0	0

Mantissa Exponent

1 mark for correct mantissa
1 mark for correct exponent

2

(b) *1 method mark for either:*

- showing correct value of both mantissa and exponent in denary
- showing binary point shifted 2 places to right in binary number
- indicating that final answer calculated using answer = mantissa x 2^{exponent}

1 mark for correct answer

Mantissa = -0.625 // -5/8
Exponent = 2
Answer = -2.5 // -2½

2

(c)

0	• 1	1	0	0	1	1	0

Mantissa

0	1	0	0

Exponent

1 mark for correct mantissa
1 mark for correct exponent

2

(d) Maximises precision/accuracy for given number of bits;

Note: Must have concept of given number of bits or an example of this e.g. word length.

Unique representation of each number // simpler to test for equality of numbers;

2

(e) Reduced precision;
Increased range;
A can represent larger/smaller numbers
No effect on amount of memory required to represent a number;

Max 2

[10]

Q20.

(a) 167;

1

(b) $10.4375 // 10\frac{7}{16}$;;

1 mark for correct integer part
1 mark for correct fractional part

2

(c) -;89;

1 mark for correct sign
1 mark for correct integer value

2

(d) A7;

1

[6]

Q21.

(a)

0	• 1	1	1	1	1	1	1

Mantissa

0	1	1	1

Exponent

1 mark for correct bit pattern in both mantissa and exponent.

1

(b) Mantissa = –0.6875 // –11/16
Exponent = 3
Answer = –5.5 // –5½

1 method mark for either:

- showing correct value of both mantissa and exponent in denary
- showing binary point shifted 3 places to right within a correct binary pattern*
- indicating that final answer calculated using answer = mantissa x $2^{exponent}$ (**A** mantissa in denary or binary but exponent must be in denary)

1 mark for correct answer

* Correct binary patterns with the binary point shifted 3 places are:

1010.1000	0101.1000
1010.1	101.1000
	101.1

2

(c)

Mantissa

0	1	1	0	1	1	0	1

Exponent

0	1	0	0

1 mark for correct mantissa
1 mark for correct exponent

2

(d)

Mantissa

0	1	0	1	1	0	0	0

Exponent

1	1	1	1

1 mark for correct mantissa
1 mark for correct exponent

2

(e) **Definition** *(2 marks):*
The result of a calculation is too large to store/represent // a number is too large to store/represent;
In the available number of bits / storage space (allow example e.g. data type, byte, word, example of a data type);
R space **NE**

Example *(1 mark):*
Multiplying two numbers together;
Dividing a number by a number less than one / small number;
R zero

A Adding two numbers (of same sign)
A When number converted from one type to another that does not have suitable range / enough bits / enough storage space to represent it
A Answers by example

Max 1

3

[10]

Q22.

(a) Smallest ;
picture element // unit which can be drawn on screen // addressable / resolvable part / unit of a picture ;

2

(b) (i) 0010 1010 ;

1

(ii) 184 ;

1

(c) (i) pixels are stored as numbers // bit patterns / binary code // RGB bits ;

1

(ii) 8 ; **A** 1 byte

1

(d) (i) drawing is made up of drawing objects // or by example e.g. drawing is made up of circle / rectangle / straight line / etc. (must give at least two example objects) ;
different objects(**A** shapes) have a defined set of properties // or by example;

objects are stored as drawing commands / drawing list ;
some properties use mathematical equations / formulae ;

Max 2

(ii) object type ; co-ordinates / location of the centre **R** centre (only) ; radius / diameter ; fill colour ; fill style ; line thickness ; line colour ; line style ; anything reasonable ;
R colour (only) Position (only)

Max 3

[11]

Q23.

(a) Any 8-bit pattern padded with 0's ; ending in 1011 ;

2

(b) (i) 0111 0111 ;

1

(ii) 1 ; 111 0011 ;

2

(c) seven leading 0's ;
11001 ; (.) 1100 ;

3

[8]

Q24.

(a) (i) Picture element // smallest resolvable/rectangular area/unit (**A** smallest dot) which can be drawn on screen // smallest addressable part/unit of a picture;
smallest unit which is mapped to memory;

1

(ii) Pixels are stored as numbers/bit patterns (**A** values) which represent different colours;
A or by example;

1

(b) 1;

1

(c) (picture / image) width;
(picture / image) height;
A (picture / image) dimensions
R size
image resolution / colour depth / No. of bits per pixel;
colour palette / No. of colours in image;
offset to the start of image data;
compression type;

Max 2

(d) (i) loop counter / (loop) control variable // array subscript/index;
array of Byte;
A array of Integer

2

(ii) 1101;
I. any additional leading 0's

1

(e) (i) ThisWidth;
X;

2

(ii) 2-dimensional array (of Byte);

1

(f)

ThisWidth	ThisHeight	Counter	X	Y	ThisByte
8	5	0	1	1	255
				2	255
				3	255
				4	255
				5	255
				6	255

	Final
[0]	25
[1]	96
[2]	96
[3]	24
[4]	24
[5]	113

				7	255	[6]	
				8	255	[7]	
			2	1	255	[8]	
				2	25	[9]	
		1		3	25	[10]	
		2		4	96	[11]	
		3		5	96	[12]	
		4		6	24	[13]	
		5		7	24	[14]	
		6		8	113	[15]	

Mark as follows:
Counter has incremented from 0 to 6 (only);
X variable has incremented 1 and 2 (only);
Y variable has incremented 1 – 8 (only) at least once;
ThisByte contains first ten correct values;
Final[0] contains 25;
Final[1] to Final[5] are correct and with no other array subscripts used;

A correct six values (only) in Final array (in consecutive but wrong positions)

Max 6

(g) (i) program / constant / module / unit / user defined type / label /object / component / control / class;

A 'control' by example e.g. text box, drop down list
A any elements which are SQL specific

1

(ii) Maximum number of characters;
No punctuation characters;
No use of reserved words;
Must not start with a digit character;
case critical e.g. must start with lower case character;

A any answer which describes 'general' programming language restrictions.
identifier names must be unique;free-format not allowed for certain constructs, e.g. statement must not spread over two lines;
restrictions on identifiers used for labels;
loop control variable must be ordinal/integer;
array index range is restricted;
all variables must be pre-declared;

Max 2

[20]

Q25.

(a) 984;

1

(b) 984;

1

(c) (i) – 13.0;;;

Allow method marks

1 mark for correctly identifying negative number

1 mark for integer value correct

1 mark for fraction (dependant on correct integer value)// 01101.000

3

(ii) To Maximise precision <u>in a given number of bits</u> //

To minimise rounding errors;

A to Maximise accuracy <u>in a given number of bits</u>

1

(iii) leftmost 2 digits/bits are different//

a significant bit is stored after the binary point//

bit after point different from bit before point;

A the first bit after the sign bit is a '0';

A The second bit is a '0';

A *an answer that <u>clearly</u> implies a '0' follows the '1'*

1

(iv) 127// $2^7 - 1$;;

Max 1 for correct mantissa (01111111) or exponent (0111/7)

2

[9]

Q26.

(a) (Sound/voice) recording/er // sampling/er (software) // audio capture software;

Operating system **A** OS;

Driver;

Codec;

R Microphone software

R Analogue to digital converter

Max 2

(b) (i) Number of samples/measurements taken <u>per second/unit time</u>;

Frequency/how often samples are recorded/taken;

R Rate of …"

R "Intervals at which …"

Max 1

(ii) 1000 samples/measurements per second;

1 sample/measurement per millisecond (ms);

1000 Hz /1 KHz;

R 1000 (only)

Max 1

(c) 8 (bits);

1

(d) (i) (Sound) quality will be improved/clearer **R** Smoother // better/higher resolution //more accurate // higher fidelity;

the height of the wave will be measured more precisely/accurately;
R larger range of frequencies is possible

Max 1

(ii) The size of the sound file will increase // file uses more memory /disk space;
R 'uses more space'

1

(e) <u>0</u>110 1100;

1

(f) *All correct answers must fit the context of how the byte(s) are interpreted by the application program (not by the user of the application).*

<u>Program</u> instruction(s) // machine code;
Integer (number);
Real (number) / Floating point;
Exponent;
Mantissa;
(BUT Real/Floating Point + Exponent + Mantissa scores Max 2)
BCD (number);
R Number / denary / binary

ASCII (code);
Unicode;
EBCDIC;
Character (BUT not in addition to specific codes above) **R** Keystroke;

Address / pointer /memory reference **R** Location;
String **R** Word;
Format code // system setting / device status/signal;

A any 'data type' descriptor (e.g. Boolean) – any three data types gets but excluding any answers above;
A Colour;

Max 3

[11]

Q27.

(a) 35,37;

1

(b) 53,55;

1

(c) 00111001;

1 mark for sign, 1 mark for value
A 11000111 *1 mark*

2

(d) 0.1110010 ; 00000110;

2

(e) To allow a wider range of values to be stored;
To minimise rounding errors//
Greater accuracy/precision using a <u>given number of bits</u>;

2

[8]

Q28.

Address 56 - 55 ;

Address 57 – 137 ;

[2]

Q29.

(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 <u>signals</u> 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 ;
<u>Printer</u> driver ;
R ‘printing software’

Max 2

[10]

Q30.

(a) 40E

1

(b) 1038

1

(c) 64.875
1 mark for 64, 1 mark for .875
A 7/8

2

(d) (i) 0.125//⅛;;;
If incorrect part marks as follows
mantissa = 0.5 or ½

1

exponent = -2

1

times $2^{exponent}$

1

(ii) Leftmost 2 digits/bits are different//
A significant bit is stored after the binary point//
Bit after point different from bit before point;
A the first bit after the sign bit is a '1';
A The second bit is a '1';
A *an answer that clearly implies a '1' follows the '0'*

1

(iii) 127;;//1111111;;//0.1111111; x 2^7/2^{111};

2

[10]

Q31.

(a) 00011011;
1;1101101;

3

(b) 10000;1000; Allow FT from (a)
Stop marking when an error occurs

2

(c) (i) Carry Bit;
R Overflow bit

1

(ii) For multi-word/byte operations//test for overflow

1

(d) 1B;
ED;
Allow FT

2

[9]

Q32.

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

(ii)

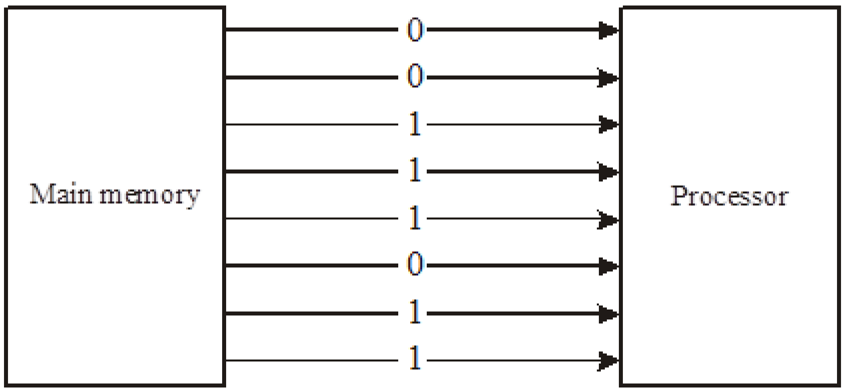

Diagram has 8 lines;
Diagram has the correct 1 bit per line; A**.** pulses to indicate 0/1's

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]

Q33.

(a) 974;

1

(b) 151.25;;

1 mark for integer part, 1 mark for fractional part

2

(c) –104.75;;

If answer not correct award 1 mark for attempt at complementing the binary pattern

2

(d) (i) –13.125;

Allow method marks
1 mark for 2 4 seen or correct 4 bit shift
1 mark for integer value correct including sign
1 mark for fractional part

3

(ii) To maximise precision <u>in a given number of bits</u> //
To minimise rounding errors //
To have just one representation of the number //
To simplify arithmetic operations;

A to maximise accuracy in a given number of bits 1

[9]

Q34.

(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]

Q35.

(a) 140 ¼ ;;

1 mark for correct integer part,

140.25;;

1 mark for correct fractional part 2

(b) (i) -14.5;;;

Give 2 marks for 14.5
Partial marks for workings if result incorrect:
1 mark for negative number;
1 mark for x2^4 (accept 16 instead of 2^4);
A showing that binary point moves 4 places right; 3

(ii) Leftmost 2 digits/bits are different;
A significant bit is stored after the binary point;
Bit after point different from bit before point;
(negative number) starts with 10…
(positive number starts with 01)….;
A the first but after the sign bit is a ‘0’;
A The second bit is a ‘0’;
A an answer that clearly implies a ‘0’ follows the ‘1’ Max 1

(iii) To maximise accuracy/precision for a given number of bits
// to minimise rounding errors;
A more accurate/precise for a given number of bits;

A given number can only be expressed in one way in a given number of bits
// a given number can only be expressed in one way in a given format;
To simplify arithmetic/logical operations;
I range

1

[7]

Q1.

(a) Program that calculated the score for a word – if done correctly full marks could be obtained by doing this but a number of students made mistakes when writing their own score-calculating code.

(b) This question part was satisfactorily tackled but only approximately half of students got both marks. Common mistakes were to shift the binary point in the wrong direction or to treat the exponent as an unsigned integer instead of as a two's complement value.

Q2.

Approximately half of the students achieved full marks on this question part. The vast majority were able to write an unsigned representation of 608 as an integer. Some students then made an error when converting this to -608, for example by simply flipping the bits or making simple errors like missing out a digit. The most common mistake however was to incorrectly shift the binary point when attempting to normalise the value. Often an exponent of 9 was given instead of the correct value of 10 and sometimes -10 was given.

Q3.

(a) This question was well answered with just over two thirds of students correctly identifying that C was a normalised negative value. The key to doing this was identifying that the digit to the left of the binary point was a 1 and the digit to the right was a 0.

(b) Just under two thirds of student recognised that B was the smallest positive normalised value. This was most quickly achieved by spotting that the value was the only positive normalised value as it had a 0 to the left of the binary point and a 1 to the right of it.

(c) This question was fairly well tackled and approximately half of students achieved both marks. Those who did not usually achieved one mark for shifting the binary point by the correct number of places but then made a calculation error when converting to decimal.

(d) This question was very well tackled, with two thirds of students achieving full marks. The most common mistakes were to shift the binary point by five places instead of six or to give the exponent as minus six instead of six.

(e) This question was well tackled with over three quarters of students achieving the mark. Students need to be aware that the absolute error is always expressed as a positive number; the most common error was to give the answer as −0.05 instead of 0.05.

(f) Just over two thirds of students could correctly calculate the relative error. Of the students who did make errors, the most common one was to divide 0.05 by 13.75, ie the value that actually was represented instead of 13.8 which was the original value to be stored.

Q6.

This question was about floating point representation of numbers. Parts (a) to (c) which required mathematical working were mostly well answered, but parts (d) and (e) were less well so.

The majority of candidates were correctly able to identify the most negative value for part (a). For part (b), candidates had to convert from floating point to decimal. Candidates seemed to find this question part harder than similar question parts in previous years as a result of the magnitude of the mantissa (11). Candidates who worked out the answer using the calculation answer = mantissa * 2exponent appeared to make fewer calculation errors than those who attempted to work out the answer by shifting the binary point by 11 places. Part (c) was very well answered with the vast majority of candidates achieving all three marks. Those who failed to achieve full marks had most commonly arrived at an incorrect representation of 12¾ in fixed point binary at the start of their working or gave the exponent as -4 instead of +4.

The advantages of normalising floating point numbers were not well known for part (d) (i). Important advantages include achieving the maximum precision within a given number of bits and facilitating simple/efficient comparison of two numbers because each number has a unique representation. It was not sufficient to state that normalisation offered improved or maximum precision; a candidate had to make clear that this was within a given number of bits. A small number of candidates gave advantages of floating point over fixed point or advantages of normalising a database.

For part (d) (ii) candidates were presented with some stimulus material about how floating point representation was used and had to analyse this and then use their understanding of the mantissa and exponent to explain how a more precise representation could be achieved. It was pleasing to see how many candidates correctly identified that one of the two bits was redundant and could be inferred and then went on to explain that the bit that was freed up could be used in the mantissa to store an extra bit of precision. To achieve the second mark, candidates had to make clear that the extra bit would be used in the mantissa, not the exponent, which some failed to do. Some candidates answered a question from a previous paper that had asked about moving a bit between the mantissa and exponent. Such responses were not markworthy on this occasion.

Q9.

This question was about floating point number representation and all parts were well responded to by students.

For part (d), the most common mistakes were to incorrectly convert 2944 into a binary integer, by missing out one zero from the string of seven zeros or to shift the binary point by eleven places instead of twelve.

This is the second time that absolute and relative errors have been asked about. Pleasingly, in part (e)(i) far fewer students gave a negative answer than was previously the case. In part (e)(ii) the most common error was to divide by 12.75 instead of 12.87.

Q10.

The topics covered by this question were generally well-understood. Most students were able to answer parts (a)-(b) and (e)-(f) well, though a number of students gave an answer of 74 instead of -74 as the answer for part (b). For part (c), most students were not able to state the correct range with the most common wrong answer being an upper limit of 128 (rather than 127). Many students did not read the question carefully for part 4 and assumed that four bits were being used before the binary point when the question said three bits before the binary point. A number of students also did not read the question carefully for part 7 and gave answers involving the use of binary addition.

Q11.

As in previous years, candidates demonstrated a very good understanding of floating point representation.

(a) This part was extremely well tackled, with over three quarters of students achieving full marks.

(b) This part was extremely well tackled, with over three quarters of students achieving full marks.

(c) This part was not tackled as well, with less than half of the candidates achieving both marks. Some candidates mistakenly moved the binary point four places to the right instead of the left or miscalculated the exponent to be 12 instead of -4. Others did not know how to deal with the binary point being moved to the left past the sign bit in a negative number and so ended up with a positive number instead of a negative number at the end.

(d) This part of the question was well tackled, with the majority of students achieving full marks and two thirds achieving at least two of the three marks. Those students who did not achieve full marks most commonly went wrong when converting 108 in unsigned binary into -108 in two's complement. Another mistake was to write out the binary place values backwards when working out the bit pattern for 108, ie 1,2,4,8 etc.

(e) The vast majority of candidates achieved at least one mark for this part, correctly recognising that overflow occurred when a value was too large to be stored in the number of bits available. The clearest answers explicitly stated that the number of bits available for storing the number was insufficient, rather than making more vague references to, for example, memory space. Some students failed to achieve a mark because they related the number of bits to the mantissa alone. Around two thirds of candidates were able to identify correctly that the situation which might cause overflow was division for this part.

Q12.

The topics covered by this question were generally well-understood. Most students were able to answer parts (a)-(c) well, with the most common error being not reading the question carefully and using a different number of bits from that specified. For part (d), a significant number of students seemed to have memorised that 255 was the largest positive number that can be represented using 8-bit unsigned binary integers and gave this as their answer, even though the question was about two's complement binary.

Answers to part (e) showed that a number of students were not familiar with binary subtraction. However, students who knew how two's complement can be used to subtract one number from another were normally able to describe the process clearly and get good marks. Some answers seen made it clear that students thought that a two's complement binary number means a negative number, not understanding that the two's complement system can be used to represent both negative and positive numbers.

Parts (f)-(i) about finite state machines with output were not answered well with a number of students unable to work out the correct output strings even though there were two completed examples given in the question. Most students were able to get some marks for completing the state transition table in part (i) but few obtained full marks.

Q13.

As in previous years, the majority of candidates demonstrated a sound ability to manipulate floating point numbers.

The majority of students achieved full marks for parts (a) and (b).

Part (c) was well answered. Candidates who made mistakes tended to do so because they had incorrectly represented -7.75 as a fixed point number at the start of the process.

For parts (d)(i) and (d)(ii) this was the first time that candidates had been asked to calculate absolute and relative error values and responses to these question parts were mixed. Part (i) was much better answered than part (ii) was, with about two thirds of candidates achieving the mark for the former but only one third for the latter. The most common mistakes in part (i) were to work out the relative error instead of the absolute error, or to just divide 6.9 by 6.875 (or vice-versa). Many candidates simply did not write a response to part (ii) and of those who did, a lot simply rewrote their answers to part (i), presumably in the hope that the value calculated would get them one of the two marks. A few candidates made the mistake of dividing by 6.875 rather than 6.9.

Question (d)(iii) asked about improving the accuracy of the representation and was very well answered with the majority of candidates recognising that one way to do this would be to include more bits in the mantissa. A common error was to state that mantissa needed to be “larger" or “increased" which was not creditworthy, as these responses suggested that the magnitude of the mantissa would need to be changed rather than the number of bits used to represent it.

Q14.

The majority of students got full marks for this question.

Q15.

For the first time a flowchart was used to represent an algorithm in a COMP1 exam. There was no increase in difficulty resulting from this and the standard of answers was the same as seen in the previous year.

Some students did not follow the algorithm given and instead developed their own program to convert binary to denary. This resulted in them not getting many marks as they had not answered the question.

Students using VB6 tended to get lower marks on this question than those using the other languages available for COMP1. This was partly due to not providing the correct evidence for the testing (screen captures needed to show the data entered for the test as well as the result of the test), although many students using VB6 also seemed to have weaker programming skills.

Students need to be aware that an algorithm is not the same as a program and that simply copying the algorithm into their development environment will not result in a working program in any of the COMP1 programming languages – the pseudo–code/flowchart needs to be adapted to match the syntax of the programming language they are using. As in previous years, a number of students simply copied parts of the algorithm into their program code eg trying to use a keyword of OUTPUT. These appeared to be less able students who generally struggled on the Section D programming as well. The vast majority of students were able to convert the algorithm successfully into working program code and the marks obtained on this question were virtually identical to those achieved on Section B on the 2011 COMP1 exam.

Q16.

Part (a): Students were required to write the most negative number that could be represented, which is the negative number that is furthest away from zero. The vast majority of students correctly identified that the exponent would be 01111. There was more confusion about the mantissa, which may have been caused by students failing to understand how negative values should be normalised. A particularly common but incorrect response for the mantissa was 1.000001

Part (b): Students were required to perform a conversion from floating point binary to denary. This was well tackled, with the only common error being treating the bit to the left of the binary point as being worth 1 in denary rather than -1.

Part (c): This question part was well attempted. Students were asked to represent a much bigger number than has been the case on previous papers, but this did not appear to cause them any problems.

Part (d): Students needed to convert a negative denary number into floating point binary. This conversion was more complex than those in parts b and c and, as a result, whilst nearly three quarters of students achieved two of the three marks, only around 15% of students scored all three. The most commonly made error was to identify the exponent incorrectly as being -4 instead of 4. Working mistakes were also made when converting a fixed point representation of +12.5 to a two's complement representation of -12.5.

Part (e): Students were required to identify standard types of error that occur when floating point arithmetic is performed. Overflow and underflow were well known, but very few students correctly identified that the last example would produce a cancellation error. Some students lost marks through imprecise naming such as "overload" or "stack overflow".

Q17.

Most candidates obtained the mark for part (a). Those who did not get the mark often used 7 bits instead of 8 – 1111011 – (missing out the 0 on the LHS) or miscounted the number of 1s to use – 0111011. Part (b) was generally well-answered. The most common incorrect answers were 255 (the highest decimal value that can be represented using 8 bits) and 128 (the number of different values that can be represented using 7-bit binary). This suggests that candidates are recognising that they need to do 2^8, but often then get confused about when they need to take a 1 away – getting muddled finding the highest value with finding out the number of values that can be represented. Some candidates are taking 1 away from 2^8 and some are taking 1 away from 8 and giving an answer of 2^7. A larger number of candidates got the correct answer to the hexadecimal question this year, but a significant proportion did not understand what hexadecimal is and made either no attempt or gave an answer that was not in hexadecimal. Answers for part (d) often lacked precision and did not convey the idea that it is easier for a person to read the hexadecimal equivalent of a binary bit pattern. A common misconception is that hexadecimal values will use less storage space.

Q18.

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.

Q19.

Part (a): Most candidates got at least one mark for this question part, usually for correctly identifying that the exponent should be 1000. The mantissa was harder to work out, as the representation used required that this should be normalised, so the correct answer was 0.1000000. Some candidates wrongly stated that the mantissa should be 0.0000001 which would not be valid in a normalised representation.

Part (b): Most candidates knew how to convert a floating point value into denary, with nearly half achieving both marks. The most common mistake was to forget that the most significant bit in the mantissa had a negative value and so arrived at the answer 5.5 instead of the correct answer of –2.5. Candidates who did this nevertheless usually achieved one mark for their working out. Different methods could be used to arrive at the final answer. More mistakes were made by candidates who used the method of converting the bit pattern from negative to positive by flipping the bits and then adding one. Often, candidates using this method forgot to add the one at the end.

Part (c): This question part was very well answered with nearly two thirds of candidates getting full marks. A small but surprising number of candidates put a 1 in the most significant bit of the mantissa which would have produced a negative number. This mistake should have been easy to spot if these candidates had checked their answers.

Part (d): This was the part of question 3 that candidates found hardest. The advantages of normalisation are that it maximises precision within a given number of bits and also that there is a unique representation of each number which makes it simpler to test for equality of numbers. Many candidates gave advantages of floating point over fixed point rather than answering the question that was asked. Some candidates who made responses regarding precision failed to get the associated mark because they did not refer to the important fact that precision is maximised in a fixed number of bits. This is important as a number can always be represented more precisely by using more bits.

Part (e): Responses to this question part were excellent. Many candidates correctly identified that the new representation would give increased range at the expense of less precision.

Q20.

Most candidates were able to answer this question. If mistakes were made they tended to be on the two's complement and hexadecimal questions. Some candidates wrote "10 7" as an answer for part (d) – the bits had been split into blocks of 4 but then converted into decimal values.

Candidates should be encouraged to check their answers carefully as marks were sometimes dropped due to arithmetic errors.

Q21.

Part (a): Most candidates were able to write correctly the largest positive number that could be represented in the given normalised floating point system.

Part (b): This question part was well answered, with many candidates getting both marks. The most common error was to forget that the most significant bit of the mantissa had a negative value, resulting in an answer of 10.5 rather than -5.5.

Part (c): The majority of candidates got full marks for this question part. Many of those who failed to do this clearly knew the method to use, but made arithmetic errors.

Part (d): This question part was well answered, with many candidates correctly calculating both the mantissa and exponent. A small number calculated the mantissa correctly, but failed to normalise this properly, resulting in an incorrect value for the exponent.

Part (e): This question part was very well answered. Most candidates correctly explained that overflow occurs when the result of a calculation is too large to store in the available number of bits. Many were also able to give an appropriate example of how this might occur.

Q22.

For question (a) the explanation of what is meant by a pixel was generally not well answered with very few candidates gaining the full 2 marks. The ‘smallest picture element’ was required for 2 marks to be awarded.

In question (b)(i) most candidates appreciated how the memory contents shown were arrived at from the grid of pixels given in the question. Some candidates did not read the rubric and gave the answer for question (b)(ii) in binary.

For question (c)(i) all that was required was a statement which described each colour being represented by a different number. Some candidates gave detail about numbers mapping to the various amounts of red, green and blue for each colour which was not expected, but was creditworthy.

In the final part of the question (d) despite being popular on the legacy CPT1 paper, answers describing vector graphics were disappointingly poor. Candidates failed to describe the two key points that any drawing is built up as a series of drawing objects and these drawing object types each have their own set of defined properties. Candidates were often unable to give a clear explanation for question (d), but were then able to name typical properties for a circle object.

Q23.

All of this question was well answered by the majority of candidates, including the encoding of a fixed point number which was new to the AS specification.

Q24.

(a) See earlier comment in the General section of this Report.

(b) All that was required in this question was the association between a number value and a colour, and hence that different numbers are used to represent different colours. The suspicion was that the candidates were not clear of the meaning of the word 'encoding' in the question stem. Some candidates described the idea that the picture was formed by putting together many pixels. A common misconception was that the pixel value stored its location.

(c) A common wrong answer – as seen in a previous examination – described 'data which is stored in the file directory' (not the file header).

(d) (i) Very poorly answered, despite a very similar question on a recent January series question paper.

(e) (i) Well answered.

(ii) Often candidates latched on to the term 'data structure' and then chose from the stack, queue options, failing to appreciate that an array is referred to as a data structure.

(f) On the one previous question paper on which the algorithm trace used a nested loop, the quality of answers seen was encouraging and the Report commented on this. Alas, the impetus was not maintained, and the number of candidates who were able to score 5 or 6 marks was small. The common error on the better scripts was not to make the final increment of the Counter variable value 6.

(g) (i) Most candidates came up with a valid answer from the large range deemed acceptable.

(ii) Many candidates were able to come up with two restrictions on the choice of identifier names. Some scored 1 mark only by quoting two near identical reasons e.g. cannot contain a 'comma' character followed by 'cannot contain a question mark' character. Some candidates answered their own question e.g. 'cannot store a text character in a integer data type variable'. Other candidates read the question as 'general restrictions' of the programming language and so gave answers such as 'variables must be declared before they can be used,' Answers of this nature were given credit.

Q25.

A large number of candidates were clearly not well-prepared for what is intended to be a straight forward starter question. In part (c) many candidates were unable to complement the negative mantissa. The role of the exponent was generally understood but converting the resulting bit pattern into a decimal value was not. Parts (c)(ii) and (c)(iii) should have been standard bookwork, but they presented a real challenge with few candidates obtaining full marks. Part (c)(ii) presented a particular challenge. In part (c)(iv) there were far too many answers of 255 or 256 showing a lack of understanding.

Q26.

This question framework was different to that seen on previous papers but candidates generally answered well and were able to relate the diagram given to their basic definitions.

(a) The most popular answers were 'sound recording software' and 'the operating system'.

(b) (i) Many answers simply rearranged the word in the question stem e.g. 'rate and which samples are taken' and so failed to score.

(ii) This was poorly answered, often when the candidate had been unable to write a worthy answer for (i).

(c) Well answered.

(d) (i) Generally candidates scored the one mark. Any answer which suggested that a more 'faithful' recording was obtained was given credit. However 'smoother' sound suggested more samples would be required and so did not score.

(ii) Again, generally well answered. Wrong answers included that it would slow down the processing time or (worst) take more time to sample.

(f) There is a statement at the start of the mark scheme for this question which is indicative of what was expected. Some candidates carried forward answers from a previous similar question which, because of their different context, were deemed unacceptable and included 'part of a word processed file, etc'. The key discriminator was how the bytes(s) would be interpreted by the processor or application software

– not by the user sat in front of the application. The computer scientist who wrote typically, ‘binary integer, memory address, ASCII character code,’ pocketed three very accessible marks, and quickly moved on.

Q27.

Candidates found this question surprisingly difficult. Parts (a) and (b) were generally done well but there were a significant number of candidates who were not able to convert binary numbers into hexadecimal and/or denary. Part (c) proved to be more challenging. A sizeable number of candidates provided a negative value. Even fewer candidates were unable to obtain the correct bit pattern for part (d). A common error was to fail to normalise the mantissa. Part (e) was standard bookwork but a substantial number of candidates were unable to give two advantages.

Q28.

By and large was well answered with most candidates scoring the full 2 marks.

Q29.

(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.

Q30.

Well-prepared candidates found this very straight forward and many obtained full marks. Some candidates, however, were clearly not well prepared. Parts (a), (b) and (c) were generally well done. Part (d) presented more of a challenge. Many candidates were unable to convert the negative exponent into a decimal value in (d)(i). There were also a number of candidates who were unable to recognise that, although the exponent was negative, the mantissa was positive. In part (d)(iii) there were many candidates who failed to recognise that the largest positive value could not start with a 1. Marks were often lost in (d)(ii) due to the candidates being unable to express themselves properly.

Q31.

Most candidates were able to convert 27 into binary. Fewer candidates were able to deal with a negative number satisfactorily. Many candidates who failed to obtain the correct value for –19 did not add together their binary values but simply converted 8 into binary for their answer to part (b). Most of the candidates who managed to obtain the correct answers for part (a) also obtained the correct answer for part (d). From time to time in part (d) the correct answers for 27 and -19 were given in hexadecimal although incorrect binary patterns were provided in part (a). This suggested, as did a number of the other incorrect responses that were seen, that some candidates attempted to use electronic calculators to find the answers to question one without really understanding what they were being asked to do.

It was very disappointing to see how few candidates had any clear idea of the name or the use of the additional bit.

Q32.

(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.

Q33.

This question gave most candidates a good, often very good, start to the paper. Surprisingly, the floating-point question (part (d)(i)) was answered correctly more often than the fixed point one (part (b)). Some candidates lost marks for partially correct answers through not showing all of their working.

Part (b)(ii) was not well answered. Many candidates mentioned improved precision or accuracy but failed to state that this also depends on the number of bits used for the representation.

Q34.

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.

Q35.

(a) It is very worrying that many candidates gave a negative answer when the question clearly stated that the bit pattern represented an unsigned number.

(b) (i) Two's complement is largely understood but some candidates missed out on marks by not showing working and getting their final answer incorrect, so gaining no marks. Part marks were awarded for showing a correct method of conversion.

(ii) Many candidates correctly stated that the bit pattern starting with 10 showed that the number was normalised, but some candidates, incorrectly, stated that it was because there were no leading zeros, clearly not appreciating that a negative number will always start with a 1 but may not be normalised.

(iii) Many candidates, correctly, stated that normalised form would maximise precision with a given number of bits.