9608/31/O/N/15

1 In a particular computer system, real numbers are stored using floating-point representation with:

- 8 bits for the mantissa, followed by
- 8 bits for the exponent

Two's complement form is used for both mantissa and exponent.

(a) (i) A real number is stored as the following two bytes:

Mantissa

0	0	1	0	1	0	0	0

Exponent

0	0	0	0	0	0	1	1

Calculate the denary value of this number. Show your working.

....

....

....

....

....

....[3]

(ii) Explain why the floating-point number in part (a)(i) is not normalised.

....

....[2]

(iii) Normalise the floating-point number in **part (a)(i).**

Mantissa

Exponent

[2]

(b) (i) Write the largest positive number that can be written as a normalised floating-point number in this format.

Mantissa

Exponent

[2]

(ii) Write the smallest positive number that can be written as a normalised floating-point number in this format.

Mantissa								Exponent							

[2]

(iii) If a positive number is added to the number in part (b)(i) explain what will happen.

....

....

....

....[2]

(c) A student writes a program to output numbers using the following code:

```
X ← 0.0
FOR i ← 0 TO 1000
   X ← X + 0.1
   OUTPUT X
ENDFOR
```

The student is surprised to see that the program outputs the following sequence:

0.0 0.1 0.2 0.2999999 0.3999999 ......

Explain why this output has occurred.

....

....

....

....

....

....[3]

9608/32/O/N/15

1 In a particular computer system, real numbers are stored using floating-point representation with:

- 8 bits for the mantissa, followed by
- 4 bits for the exponent

Two's complement form is used for both mantissa and exponent.

(a) (i) A real number is stored as the following 12-bit binary pattern:

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

Calculate the denary value of this number. Show your working.

....

....

....

....

....

....[3]

(ii) Give the **normalised** binary pattern for **+3.5**. Show your working.

....

....

....

....

....

....[3]

(iii) Give the **normalised** binary pattern for **–3.5**. Show your working.

....

....

....

....

....

....[3]

The number of bits available to represent a real number is increased to 16.

(b) (i) If the system were to use the extra 4 bits for the mantissa, state what the effect would be on the numbers that can be represented.

....

....[1]

(ii) If the system were to use the extra 4 bits for the exponent instead, state what the effect would be on the numbers that can be represented.

....

....[1]

(c) A student enters the following expression into an interpreter:

```
OUTPUT (0.1 + 0.2)
```

The student is surprised to see the following output:

0.30000000000000001

Explain why this output has occurred.

....

....

....

....

....

....[3]

9608/31/O/N/16

1 In a particular computer system, real numbers are stored using floating-point representation with:

- 12 bits for the mantissa
- 4 bits for the exponent
- two's complement form for both mantissa and exponent

(a) Calculate the floating-point representation of + 2.5 in this system. Show your working.

Mantissa

	•										

Exponent

....

....

....

....

....

.... [3]

(b) Calculate the floating-point representation of **− 2.5** in this system. Show your working.

Mantissa

•											

Exponent

....

....

....

....

....

.... [3]

(c) Find the denary value for the following binary floating-point number. Show your working.

Mantissa

0 •	0	1	1	0	0	0	0	0	0	0	0

Exponent

0	0	1	1

....

....

....

....

....

.... [3]

(d) (i) State whether the floating-point number given in part (c) is normalised or not normalised.

.... [1]

(ii) Justify your answer given in part **(d)(i).**

....

.... [1]

(e) The system changes so that it now allocates 8 bits to both the mantissa and the exponent.
State two effects this has on the numbers that can be represented.

1 ....

....

2 ....

.... [2]

9608/31/M/J/18

1 In a computer system, real numbers are stored using normalised floating-point representation with:

- 12 bits for the mantissa
- 4 bits for the exponent
- Two's complement form for both mantissa and exponent.

(a) Find the denary value for the following binary floating-point number.

Mantissa

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

Exponent

0	1	0	1

Show your working. Working

....

....

....

....

....

Answer .... [3]

(b) Calculate the normalised floating-point representation of 5.25 in this system. Show your working.

Working ....

....

....

....

....

....

Mantissa **Exponent**

[3]

(c) The size of the mantissa is decreased and the size of the exponent is increased. State how this affects the range and precision of the numbers that the computer system can represent.

....

....

....

....[2]

9608/32/M/J/18

3 In a computer system, real numbers are stored using normalised-floating point representation with:

- 8 bits for the mantissa
- 4 bits for the exponent
- two's complement form for both mantissa and exponent.

(a) Calculate the **normalised** floating-point representation of **+ 21.75** in this system. Show your working.

Working ....

....

....

....

....

....

Mantissa **Exponent**

[3]

(b) Find the denary value for the following binary floating-point number.

Mantissa							
1	0	1	1	0	0	0	0

Exponent			
1	1	1	0

Show your working.

Working ....

....

....

....

....

Answer .... [3]

9608/31/M/J/19

1 In a computer system, real numbers are stored using normalised floating-point representation with:

- twelve bits for the mantissa
- four bits for the exponent.

The mantissa and exponent are both in two's complement form.

(a) Calculate the denary value for the following binary floating-point number. Show your working.

Mantissa											
1	0	0	1	0	1	1	1	0	0	1	1

Exponent			
0	1	1	1

Working ....

....

....

....

....

Answer .... [3]

(b) Calculate the normalised floating-point representation of +1.5625 in this system. Show your

Working ....

....

....

....

....

....

Mantissa | **Exponent**

[3]

(c) (i) Write the largest positive number that can be stored as a normalised floating-point number using this format.

Mantissa | **Exponent**

[2]

(ii) Write the smallest non-zero positive number that can be stored as a normalised floating-point number using this format

Mantissa | **Exponent**

[2]

(d) The developer of a new programming language decides that all real numbers will now be stored using 20-bit normalised floating-point representation. She must decide how many bits to use for the mantissa and how many bits for the exponent.

Explain the trade-off between using either a large number of bits for the mantissa, or a large number of bits for the exponent.

....

....

....

....

....

.... [3]

Answers
9608/31/O/N/15

1 (a) (i) 00101000 00000011
=0.0101 × 2 ↑3 [1]
=10.1 [1]
=2.5 [1]

(ii) For a positive number (mantissa starts with a zero) [1]
bit after binary point (second bit from left) should be a one [1]

(iii) 00101000 00000011
= 01010000 00000010 [1+1]

(b) (i) 01111111 0111111 [1+1]

(ii) 01000000 1000000 [1+1]

(iii) number will become too large to represent [1]
which will result in overflow [1]

(c) Any point 1 mark

0.1 cannot be represented exactly in binary
0.1 represented here by a value just less than 0.1
the loop keeps adding this approximate value to counter
until all accumulated small differences become significant enough to be seen
[max 3]

Answers
9608/32/O/N/15

1 (a) (i) 01101000 0011
= 0.1101 (or 1/2 + 1/4 + 1/16) × 2↑3 [1+1]
= 110.1
= 6.5 [1]

(ii) +3.5
= 11.1 [1]
= 0.111 × 2↑2 (or indication of moving binary point correctly) [1]
= 01110000 0010 [1]

(iii) 01110000 Allow f.t. from **(ii)**
10001111 One's complement on mantissa [1]
10001111 +1 Two's complement [1]

= 10010000 0010 [1]

(b) (i) Precision/accuracy of numbers represented will increase [1]

(ii) Range of numbers represented will increase [1]

(c) Any point, 1 mark (max. 3)

0.1/0.2 cannot be represented exactly in binary // rounding error [1]
0.1 represented by a value just greater than 0.1 // 0.2 represented by a value just greater than 0.2 [1]
adding two representations together adds the two differences [1]
summed difference significant enough to be seen [1]
[max. 3]

Answers
9608/31/O/N/16

1 (a) +2.5
= 010100000000 0010 [3]
Give full marks for correct answer (normalised or not normalised)

= 10.1 [1]
= 0.101 × 2^2 // evidence of shifting binary point appropriately [1]

[Max 3]

(b) –2.5
101100000000 0010
Give full marks for correct answer

One's complement of 12-bit mantissa of +2.5 101011111111 – allow f.t. [1]
+1 to get two's complement 101100000000 [1]

[Max 3]

(c) 3 [3]
Give full marks for correct answer

= 0.011 X 2^3 // exponent is 3 [1]
= 11.0 // (1/4+1/8) * 8 [1]

[Max 3]

(d) (i) Not normalised [1]

(ii) First two bits should be different for normalised number
// because the number starts with 00 [1]

(e) reduced accuracy [1]
increased range [1]

9608/31/M/J/18

Question	Answer	Marks
1(a)	**1** mark per bullet max **2** • 0101 = 5 (conversion of exponent to denary) • 1.01110011010 = –0.10001100110 (conversion of mantissa to negative binary number) • –10001.100110 (binary value)// –0.54980469 (denary value of mantissa) // –563/1024 Or • Use exponent to denormalise mantissa **1** mark for correct answer • = –17 19/32 // –17.59375	**3**
1(b)	**1** mark per bullet • 5.25 = 101.01 (conversion to binary) • = 0.10101 · 2^3 (evidence of shifting binary point appropriately) • 010101000000 0011 (stored as mantissa and exponent)	**3**
1(c)	**1** mark per bullet • (Size of mantissa decreased means that) precision is reduced • (Size of exponent is increased means that) range is increased	**2**

Answers
9608/32/M/J/18

Question	Answer	Marks
3(a)	1 mark per bullet • 21.75 = 010101.11 (conversion to correct binary) • 0.1010111 × 2^5 (evidence of shifting binary point appropriately) • 01010111 0101 (stored as mantissa and exponent)	**3**

3(b)	1 mark per bullet, max **2**
	
	• 1110 = –2 (conversion of exponent to denary)
	• 1.011000 = –0.101 (conversion of mantissa to negative binary number)// –0.625 (denary value of mantissa)// –5/8
	• –0.00101 (binary value) //
	
	Or
	• Use exponent to denormalise mantissa
	
	1 mark for correct answer
	• –5/32 // –0.15625
		3

Answers
9608/31/M/J/19

Question	Answer	Marks
1(a)	**2 marks** for working shown **1 mark** for the correct answer Working: • Correct calculation of negative value (any method) (= –0.11010001101) • Correctly moving the binary point 7 places (= –01101000.1101) // Exponent 7 Answer: • −104.8125 // $-104\frac{13}{16}$	**3**
1(b)	**2 marks** for working shown **1 mark** for the correct answer Working: • Correct conversion to binary (01.1001) • Correct calculation of exponent (1) Answer: • (Mantissa) 0110 0100 0000 (Exponent) 0001	**3**
1(c)(i)	**1 mark** per bullet point • Mantissa = 0111 1111 1111 • Exponent = 0111	**2**
1(c)(ii)	**1 mark** per bullet point • Mantissa = 0100 0000 0000 • Exponent = 1000	**2**

1(d)	**1 mark** per bullet point to **max 3** - The trade-off is between range and precision - Any increase in the number of bits for the mantissa, means fewer bits available for the exponent // Any decrease in the number of bits for the mantissa, means more bits available for the exponent - More bits used for the mantissa will result in better precision - More bits used for the exponent will result in a larger range of numbers - Fewer bits used for the mantissa will result in worse precision - Fewer bits used for the exponent will result in a smaller range of numbers	**3**