Floating point assignment #1

Zahra Zahid

Real numbers are stored using floating-point representation in a computer system.

This representation uses:

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

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

(a) **(i)** A real number is stored as a 12-bit normalised binary number as follows:

Mantissa

0	1	0	1	0	0	1	0

Exponent

0	0	1	0

Calculate the denary value for this binary number. Show your working.

Working ....

....

....

Denary value .... [3]

(ii) Calculate the normalised binary number for –3.75. Show your working.

Mantissa

Exponent

Working ....

....

....

.... [3]

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

State the effect of increasing the size of the exponent by 4 bits.

....

.... [1]

(c) State why some binary representations can lead to rounding errors.

....

.... [1]

(d) Complete the following descriptions by inserting the **two** missing terms.

.... can occur in the exponent of a floating-point number, when the exponent has become too large to be represented using the number of bits available.

A calculation results in a number so small that it cannot be represented by the number of bits available. This is called ....

[2]

(a) A computer stores real numbers using floating-point representation. The floating-point numbers have:

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

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

(i) Calculate the denary value of the following floating-point number.

Show your working.

Mantissa

0	0	1	1	0	1	1	1

Exponent

0	1	0	1

Working ....

....

....

....

....

Answer ....

[3]

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

....

.... [1]

(iii) Give the floating-point number in **part (a)(i)** in normalised two's complement format.

Mantissa

Exponent

[2]

(b) (i) Convert the denary number +11.625 into a normalised floating-point number.

Show your working.

Working ....

....

....

....

....

....

Mantissa **Exponent**

[3]

(ii) Convert the denary number –11.625 into a normalised floating-point number.

Show your working.

Working ....

....

....

....

....

....

Mantissa **Exponent**

[3]

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

```
OUTPUT(0.2 * 0.4)
```

The student is surprised to see that the interpreter outputs the following:

```
0.08000000000000002
```

Explain why the interpreter outputs this value.

....

....

....

....

....

.... [3]

A normalised floating point representation uses an 8-bit mantissa and a 4-bit exponent, both stored using **two's complement format**.

(a) In binary, write the largest positive number that can be represented using this normalised floating point system in the boxes below:

[2 marks]

•												

Mantissa — Exponent

(b) This is a floating point representation of a number:

0	•1	0	1	1	0	0	0	0	0	1	1

Mantissa — Exponent

Calculate the denary equivalent of the number. Show how you have arrived at your answer.

Working ....

....

....

[1 mark]

Answer ....

[1 mark]

(c) This is a floating point representation of a number:

Mantissa

1 ●	0	1	1	0	0	0	0

Exponent

1	1	0	0

Calculate the denary equivalent of the number. Show how you have arrived at your answer.

Working ....

....

....

[1 mark]

Answer ....

[1 mark]

(d) Write the normalised floating point representation of the **negative** denary value −108 in the boxes below. Show how you have arrived at your answer.

Working ....

....

....

....

[2 marks]

Answer

Mantissa

●							

Exponent

[1 mark]

(e) (i) In the context of floating point representation, explain what overflow is.

[2 marks]

....

....

....

....

(e) (ii) **Table 5** below contains descriptions of operations which may or may not cause an overflow error when they are carried out with a floating point representation.

Place **one** tick next to the operation that may cause overflow.

[1 mark]

Table 5

Operation	**May cause overflow? (Tick one box)**
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.	

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]

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]