# MATH1005/MATH6005 Semester 1 2021

# Assignment 3

**Workshop Details:**

| Number | Day | Time | Demonstrator name |
|---|---|---|---|
| 16 B | **Friday** | 11:00 ~~24~~26/03 | Cai Yang |

**Student Details:**

| ID | Surname | Given name | Preferred name |
|---|---|---|---|
| u7235649 | Zhang | Han | |

## Instructions:

This assignment has four questions. Write your solutions in the spaces provided. Hand-writing is preferable to typesetting unless you are fast and accurate with LaTeX, and even then typesetting will take you longer. Also, be aware that typesetting will not be allowed on exams.

**Except for multiple-choice questions, or where answer boxes are provided** (one question of this type each week), **show all working**.

Question numbers indicate the corresponding questions from the workshop. Since the workshop has six questions, questions on this assignment may not be sequentially numbered.

## Declaration:

*I declare that while I may have discussed some or all of the questions in this assignment with other people, the write-up of my answers herein is entirely my own work. I have not copied or modified the written-out answers of anyone else, nor allowed mine to be so used.*

Signature:...Han..Zhang.............................. Date:..31/03/2021

## This document must be submitted by 11pm on the THURSDAY following your workshop.

Sign, date, then scan this completed document (5 pages) and save as a pdf file with name format uXXXXXXXAssXX.pdf (*e.g.* u6543210Ass01.pdf ).
Upload the file via the link from which you downloaded this document.

***If copying is detected, and/or the document is not signed, no marks will be awarded.***

This document has five pages in total.

**Question 1**$^{+}$ True or false? Justify.

(a) $37 \bmod 7 = 0$ (b) $37 \operatorname{div} 7 = 0$ (c) $500 \bmod 33 = 3$

(d) $84 \equiv 44 \pmod 8$ (e) $44 \equiv 84 \pmod 8$ (f) $84 \equiv -44 \pmod 8$

(a) False. ~~8~~ $37 = 5 \times 7 + 2$. Should be 2.

(b) False. $37 = 5 \times 7 + 2$. Should be 5.

(c) False $500 = 15 \times 33 + 5$. Should be 5.

(d) ~~False~~ True $\cancel{84 = 44 + 8 \times 5}$. $84 \bmod 8 = 4$. $44 \bmod 8 = 4$

(e) True $\cancel{44 = 84 + 8 \times (-5)}$ $44 \bmod 8 = 4$ $84 \bmod 8 = 4$

(f) ~~False~~ True $\cancel{84 = -44 + 8 \times 16}$ $84 \bmod 8 = 4$

$-44 \bmod 8 = 4$.

**Question 2**† Write your answers in the boxes provided.

**For this question, working is not required and will not be marked.**

This question asks you to do translations and arithmetic using numbers expressed in base 3 and base 9. Of course, in practice, bases 2 and 16 are far more commonly used than 3 and 9, and so translations and calculations for binary and hex can be done directly on most calculators. So this question is designed to test your understanding of the principles rather than your proficiency with a calculator.

Hint: The powers of 3 are: 1, 3, 9, 27, 81, 243, 729, 2187, ...

(a) Express $20120_3$ in decimal (i.e. using ordinary base 10) $\boxed{177}$

$20120_3 = 0\times 1 + 2\times 3 + 1\times 9 + 0\times 27 + 2\times 81$
$= 6 + 9 + 162$
$= 177$

(b) Express 357 (i.e. $357_{10}$) using base 3. $\boxed{111020_3}$

$357 = 243\times 1 + 81\times 1 + 27\times 1 + 9\times 0 + 3\times 2 + 1\times 0$
$= 111020_3$

(c) Express $2212012011_3$ using base 9. (Do it directly, not via decimal!) $\boxed{85164_9}$

$2212012011_3 = 2\ 212\ 012\ 011_3$
$= 85164_9$

(d) Express $24680_9$ using base 3. (Do it directly, not via decimal!) $\boxed{211202200_3}$

$24680_9 = 02\ 11\ 20\ 22\ 00_3$

(e) Staying in base 3, calculate the base 3 sum:

$$\begin{array}{r} 2\ 1\ 0\ 1\ 0\ 2 \\ +\ 1\ 0\ 2\ 2\ 1\ 1 \\ \hline \boxed{1\ 0\ 2\ 0\ 0\ 2\ 0_3} \end{array}$$

(f) Staying in base 3, calculate the base 3 difference:

$$\begin{array}{r} 2\ 1\ 0\ 1\ 0\ 2 \\ -\ 1\ 0\ 2\ 2\ 1\ 1 \\ \hline \boxed{\cancel{1\ 0\ 0\ 1\ 2\ 1}} \end{array}$$

$100121_3$

**Question 3*** This question is about the toggle-plus-one method as it relates to the storage of integers in computer words. It also shows how this method avoids the need for separate subtraction circuits.

As demonstrated in lectures, for a binary word $W$, toggle-plus-one means:

| | |
|---|---|
| toggle: | replace every 1 by 0, every 0 by 1, then |
| add one: | treating $W$ as a binary number, add 1. |
| | Ignore any carry beyond the length of the word. |

For $l \in \mathbb{N}$ let $S_l = \{n \in \mathbb{Z} : -2^{l-1} \leq n < 2^{l-1}\}$. Then $S_l$ is the set of all integers that can be stored in words of length $l$ bits, using the standard computer representation. The rules governing the storage of an integer $n$ in a word $W$ may be summarised as:

| | |
|---|---|
| Rule 1: | $n$ is negative if and only the left-most bit of $W$ is 1 |
| Rule 2: | $-n$ is stored as the word obtained from $W$ by toggle-plus-one. |
| | This is true even when $n$ is negative. |
| Rule 3: | If the left-most bit of $W$ is 0 than $n = W_2$. |
| | *i.e.* $n$ is retrieved by treating $W$ as a binary number. |

In computer arithmetic, subtraction uses negation and addition: $x - y = x + (-y)$.
In the addition, any carry beyond the length of the word is ignored.
As an example, take $x = 11100_2$, $y = 10001_2$ and use 8-bit computer arithmetic on:

(a) $x - y$ (b) $y - x$ (c) $-x - y$

Check your results by expressing $x$, $y$ and your answers in decimal.

$x = 2^0 \times 0 + 2^1 \times 0 + 2^2 \times 1 + 2^3 \times 1 + 2^4 \times 1 = 28_{10} = 00011100_2$

$y = 2^0 \times 1 + 2^1 \times 0 + 2^2 \times 0 + 2^3 \times 0 + 2^4 \times 1 = 17_{10} = 00010001_2.$

$-x = 11100011_2 + 1 = 11100100_2$

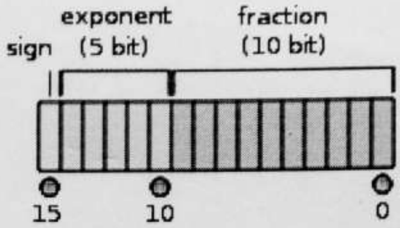
$-y = 11101110_2 + 1 = 11101111_2.$

(a) $x - y = x + (-y) = 00011100 + 11101111 = \underline{00001011} = 11_{10}$

(b) $y - \cancel{x}\, x = y + (-x) = 00010001 + 11100100 = \underline{11110101} = -11_{10}$

(c) $-x - y = -x + (-y) = 11100100 + 11101111 = \underline{11010011} = 45_{10}$

**Question 5**# The shortest IEEE standard for representing rational numbers is called *half-precision floating point.* It uses a 16-bit word partitioned as in the diagram at right. (This diagram is taken from the Wikipedia article on the subject, where more details can be found.)




As described in lectures, to store a rational number $x$ it is first represented as $(-1)^s \times m \times 2^n$ with $1 \le m < 2$. The sign bit $s$ is stored as the left-most bit (bit 15), the mantissa $m$ (called "significand" in IEEE parlance) is stored in the right-most 10 bits (bits 9 to 0), and the exponent $n$ is stored in the 5 bits in between (bits 14 to 10). However:

- *Only the fractional part of $m$ is stored.* Because $1 \le m < 2$, the binary representation of $m$ **always** has the form $1 \cdot \star\star\star\ldots$ where the stars stand for binary digits representing the fractional part of $m$. Hence there is no need to store the $1\cdot$ part.

- *the exponent $n$ is stored with an "offset".* In order to allow for both positive and negative exponents, but to avoid another sign bit, the value stored is $n + 15$. In principle this means that the five exponent bits can store exponents in the range $-15 \le n \le 16$, but 00000 and 11111 are reserved for special purposes so in fact $n$ is restricted to the range $-14 \le n \le 15$.

(a) A rational number $x$ is stored in half-precision floating point as the word $CAFE_{16}$. (That's hex shorthand for the 16-bit binary word.) Write $x$ in ordinary decimal notation.

(b) Given that $\frac{1}{7} = 0.\overline{001}_2$, find the word representing $-\frac{4}{7}$ as accurately as possible in half-precision floating point. Give your answer using hex shorthand, like the word you were given for (a).

(a) $CAFE_{16} = \cancel{1010}\ 1100\ 1010\ 1111\ 1110_2$

Sign = 1

exponent = $10010_2 = 18_{10}$ $\quad 18-15=3.$

fraction = $1011111110_2$

$CAFE_{16} = -1101.1111110 = -[(1+4+8)+2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-5}+2^{-6}] = \cancel{-11.96875}\ \cancel{11.984375}\ -13.984375$

(b) $\frac{4}{7} = 0.\overline{100}_2$

Sign = 1

exponent = $-1+15 = 14_{10} = 01110_2$

fraction = 0010010010

$-\frac{4}{7} = \cancel{100}\ 1011100010010010 = B892_{16}$