ENEE 446, FALL 2005
Solutions to Home Assignment 3

1. The expressions are evaluated below:
   (a) $2 \times 4 - 4 \times 5 + 3 \pmod{8} = 0 + 4.5 + 3 \pmod{8} = 4 + 3 \pmod{8} = 7$
   (b) $2 \times 9 - 6/2 + 4 \pmod{11} = 7 + 5.2^{-1} + 4 \pmod{11} = 5.6 \pmod{11} = 8$
   (c) $9/3 + 12/4 + 8/2 \pmod{17} = 9.3^{-1} + 12.4^{-1} + 8.2^{-1} \pmod{17}$
       $= 9.6 + 12.13 + 8.9 \pmod{17} = 3 + 3 + 4 \pmod{17} = 10$

2. Proof: Consider the set $A = \{1, 2, \ldots, p-1\}$.
   Let $a \in A$ and $B = aA = \{1 \ast a, 2 \ast a, \ldots, (p-1) \ast a\}$ where all the multiplications are mod $p$.

   Claim: $B = A \iff a$ and $p$ are co-prime.
   Proof: Without loss of generality choose $x, y \in A$ with $x \neq y$ such that $y < x$.

   We compare two elements of $B$, now, $ax = ay \pmod{p}$ iff $a(x - y) = 0 \pmod{p}$. Clearly,
   neither $a$ nor $x - y$ is a multiple of $p$ as $1 \leq a \leq p - 1$ and $1 \leq x - y \leq p - 2$. Therefore,
   $a(x - y) = 0 \pmod{p}$ only when $a$ and $p$ have a common factor, i.e., iff they are not co-prime.

   Sufficiency: Since this is true for all $x \neq y$, $B$ has $p - 1$ distinct elements when $a$ and $p$ are
   co-prime $\implies A = B$ when $a$ and $p$ are co-prime.

   Necessity: For the forward argument, if $B = A$ then no two elements of $B$ can be equal and
   the above reasoning again applies. ■

   The inverse for $a$ can be determined by forming $B$ as $\exists$ a unique $i \in A$ such that $a \ast i = 1$
   (mod $p$). Uniqueness is guaranteed as no two elements in $B$ are equal. Thus, $a$ has an inverse
   and it is unique if $a$ and $p$ are co-prime.

   Now all $a \in A$ are co-prime to $p$ if $p$ is prime. Hence, every non-zero element in $\mathbb{Z}_p$ has a
   multiplicative inverse and it is unique if $p$ is prime. ■

3. The evaluations are shown in the table below:

   \[
   \begin{array}{|c|c|c|}
   \hline
   q & \text{Modulo Subtraction} & \text{Modulo Division} \\
   \hline
   \frac{16}{3} \pmod{17} & 16 - 5 - 5 - 5 = 1 \pmod{17} \Rightarrow q = 3 & 16.5^{-1} = 16.7 \pmod{17} = 10 \\
   \frac{28}{4} \pmod{37} & \underbrace{28 - 4 - \cdots - 4} = 0 \pmod{37} \Rightarrow q = 7 & \underbrace{28.4^{-1} = 28.28} \pmod{37} = 7 \\
   \hline
   \end{array}
   \]

   We see that whenever the remainder is zero, i.e., the divisor exactly divides the dividend then
   the result in both the cases is the same. This can be shown as follows. Let us evaluate $a/b$
   (mod $p$). Let

   \[ a = nb + r \text{ such that } 0 \leq r \leq b - 1 \]
Then,

\[
\frac{a}{b} = a \cdot b^{-1} \pmod{p} \\
= (nb + r) \cdot b^{-1} \pmod{p} \\
= n + r \cdot b^{-1} \pmod{p}
\]

Thus, when \( r = 0 \) we get the same result namely, \( n \), using both subtraction and modulo division.

4. The algorithm for modular addition using 1’s complement and its output is given below:

```java
{switch (operation)
    {case add: sum = x + y; break;
        case sub: sum = x + ~y; break;
    }
    
    // Set the flags
    S = sign(sum);
    if(sum = 0) or (~sum = 0) Z = 1 else Z = 0;
    if((operation = 'add') and sign(x) = sign(y) and sign(x) != sign(sum)
    or (operation = 'sub') and sign(x) != sign(y) and sign(x) != sign(sum))
    F = 1;
    else
    F = 0;
    if((operation = 'add') and sign(x) = 1 and sign(y) = 1
    or (sign(x) != sign(y)) and sign(sum) = 0
    or (operation = 'sub') and sign(x) = 1 and sign(y) = 0
    or (sign(x) != sign(y)) and sign(sum) = 0)
    C = 1;
    else
    C = 0;
}
```

The Output is:

In binary
- \( x = 0000000011110101 \)
- \( y = 0000010011010010 \)
- \( \text{sum} = 0000010111000111 \)

In hexadecimal
- \( x = \text{00F5} \)
- \( y = \text{04D2} \)
- \( \text{sum} = \text{05C7} \)

In decimal
- \( x = 245 \)
- \( y = 1234 \)
- \( \text{sum} = 1479 \)

In unsigned decimal
x = 245
y = 1234
sum = 1479

sign-S = 0
overflow-F = 0
carry-C = 0
zero-Z = 0

In binary
x = 0111010100110000
y = 0011000011010100
sum = 0100010001011100

In hexadecimal
x = 7530
y = 30D4
sum = 445C

In decimal
x = 30000
y = 12500
sum = 17500

In unsigned decimal
x = 30000
y = 12500
sum = 17500

sign-S = 0
overflow-F = 0
carry-C = 1
zero-Z = 0

In binary
x = 1111111111110011
y = 000000010010001
sum = 000000010001001

In hexadecimal
x = FFF3
y = 0091
sum = 0085
In decimal
\[ x = -12 \]
\[ y = 145 \]
\[ \text{sum} = 133 \]

In unsigned decimal
\[ x = 65523 \]
\[ y = 145 \]
\[ \text{sum} = 133 \]

\[ \text{sign-S} = 0 \]
\[ \text{overflow-F} = 0 \]
\[ \text{carry-C} = 1 \]
\[ \text{zero-Z} = 0 \]

The diagram is almost the same as given in the lecture notes. The only difference is that \( \text{sum} = x + \sim y \) when subtracting and that zero flag is set when \( \text{sum} = 2^n - 1 \) as well.

5. The output is given below:

In binary
\[ x = 0000000000100100 \]
\[ y = 0000000000101010 \]
\[ \text{sum} = 0000000001001110 \]

In hexadecimal
\[ x = 0024 \]
\[ y = 002A \]
\[ \text{sum} = 004E \]

In decimal
\[ x = 36 \]
\[ y = 42 \]
\[ \text{sum} = 78 \]

In base 4
\[ x = 00000210 \]
\[ y = 00000222 \]
\[ \text{sum} = 00001032 \]

In unsigned decimal
\[ x = 36 \]
\[ y = 42 \]
\[ \text{sum} = 78 \]

\[ \text{sign-S} = 0 \]
\[ \text{overflow-F} = 0 \]
carry-C = 0
zero-Z = 0

------------------------------------------

In binary
x = 1111111111111011
y = ... = (x + 48)4
when x < 0 and FC(x) = (x)4 otherwise. The 4 subscript implies that the representation is
in base 4.

In hexadecimal
x = FFFB
y = 007D
sum = FF7E

In decimal
x = -5
y = 125
sum = -130

In base 4
x = 33333323
y = 00001331
sum = 33331332

In unsigned decimal
x = 65531
y = 125
sum = 65406

sign-S = 1
overflow-F = 0
carry-C = 1
zero-Z = 0

The algorithm remains the same as for 2’s complement in the binary case except that the
16-bit binary representation is used to get the 8 (quad) digit base-4 numbers by clubbing
groups of 2 bits together. Also the 4’s complement representation of x, FC(x) = (x + 48)4
when x < 0 and FC(x) = (x)4 otherwise. The 4 subscript implies that the representation is
in base 4.