CAB203 – Assignment 1

Section 1 Basic Math

1. Show that 79 ≡ 2 mod 11 by finding q so that 79 − 2 = 11q.

$$79 - 2 = 11q$$

$$77 = 11q$$

$$\frac{77}{11} = \frac{11q}{11}$$

$$7 = q$$

2. Using properties of exponents, find integers a, b, c so that

$$\frac{2^3 3^4 (5^{-5})^{-2}}{2^{-1}3^2 5^3} = 2^a 3^b 5^c$$

$$\frac{2^3 3^4 (5^{-5*-2})}{2^{-1}3^2 5^3} = 2^a 3^b 5^c$$

$$\frac{2^3 3^4 (5^{10})}{2^{-1}3^2 5^3} = 2^a 3^b 5^c$$

$$2^{3--1} 3^{4-2} 5^{10-3} = 2^a 3^b 5^c$$

$$2^4 3^2 5^7 = 2^a 3^b 5^c$$

$$\therefore a = 4, \qquad b = 2, \qquad c = 7$$

3. Use properties of exponents to evaluate

$$log_8 \frac{2^{12}}{4^6}$$

$\boldsymbol{Rule}$: $log_b \frac{x}{y} = log_b\, x - log_b y$

$\boldsymbol{Rule}$: $log_b a^n = n\, log_b a$

$$log_8 \frac{2^{12}}{4^6} = log_8 2^{12} - log_8 4^6$$

$$= 12\, log_8 2 - 6\, log_8 4$$

$$= 4 - 4$$

$$= 0$$

4. **Using the Python prompt, calculate ((172833·99345) + 123) mod 7**

>>> 172833 % 77

45

>>> 99345 % 77

15

>>> 123 % 77

46

>>> ((45 * 15) + 46) % 77

28

Section 2 Bit strings and operations

1. **Evaluate ((1010 & ~ (1 << 3)) ^ 0010) | (1 << 2). You may assume bit strings are 4 bits long.**

$= ((1010 \;\&\sim (1000)) \;\text{^}0010) \mid (0100)$

$= ((1010 \;\& \; (0111)) \;\text{^}0010) \mid (0100)$

$= ((0010) \;\text{^}0010) \mid (0100)$

$= (0000) \mid (0100)$

$= 0100$

$= 4$

2. **Write a python function that takes an argument (as a Python Bytes object, so that you can get individual bytes with something like byte9 = packet[9]) and returns True if the ACK flag is set (the bit is 1) or False if the ACK is cleared (the bit is 0)**

```
def checkAck(packet):
    byte13 = packet[13]
    return byte13 & (1 << 4) != 0
```

Section 3 Data Representations

1. **UTF-8 is a variable length encoding, summarised in https://en.wikipedia.org/wiki/UTF-8#Encoding. What is the UTF-8 encoding for code point U+1F600 (grinning face)? Give your answer as a bit string, grouping by 8 bits to make it easier to read and to show the byte boundaries. Note that code points are just numbers written in hexadecimal.**

$= 0xF0\ 0x9F\ 0x98\ 0x80$

= 11110000 10011111 10011000 10000000

2. Using 5-bit 2's compliment encoding:

(a) What is the range of integers that can be represented?

$$Range = [-2^{(n-1)}, \ldots, 2^{(n-1)} - 1]$$

$$Range = [-2^{(5-1)}, \ldots, 2^{(5-1)} - 1]$$

$$Range = [-16, \ldots, 15]$$

5 bits can represent a total of 32 integers as $2^5 = 32$

(b) Give the bit strings for -9 and 14

$-9 = 10111$
$14 = 01110$

(c) By hand, add -9 and 14 as bitstrings to get the bit string for 5, showing your work

$$\begin{array}{rr} & 111 \\ & 10111 \\ + & 1\underline{01110} \\ & 00101 \end{array}$$

Since the bit string has a length of 5, the last bit is cut off and the string is 00101 = 5.

3. Give one disadvantage of signed magnitude compared to 2's complement. You may want to consult online sources, in which case, cite them. It is sufficient to just give the URL

The main disadvantage of signed magnitude is that the total possible numbers that can be represented in the bit string is fewer, since 1 bit is reserved exclusively for the sign (ElectronicsTutorials, 2021).

ElectronicsTutorials. (2021). Signed Binary Numbers. Retrieved 25 March 2021, from https://www.electronics-tutorials.ws/binary/signed-binary-numbers.html

4. Using the toy model, give the bit string corresponding to the integer -42.

-42 in base 16 hexadecimal is -2A. Since it is negative and has 2 digits, the first 4 bits encode the number -2 in 4-bit 2's compliment which is 1110. Then 2 and A converted into a bit string are 0010 1010.

= 1110 0010 1010

Section 4 Sets

1. Write out, in list format, the following set:

$$\{3x + 1 : x \in \{2, -1, 4\}\}$$

$S = \{7, -2, 13\}$

2. Write out, in list format, the following set:

$\{x \in Z: x \,|60\}$

$S = \{-60, -30, -20, -15, -12, -10, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60\}$

3. Write out in list format, the following set:

$S = \{2x: x \in \{5,6,7\}\} \cap \{x \in Z: x|60\}$

$S = \{10, 12, 14\} \cap \{-60, -30, -20, -15, -12, -10, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60\}$

$S = \{10, 12\}$

4. Write a Python function that takes no arguments and returns the set above. It should be non-trivial— you should do the operations and not just return the set as a literal. Hints: you may use the fact that all integers that divide 60 are between -60 and 60 (inclusive) so that you don't have to test the infinite set of integers, and remember not to divide by 0

```
def getSet():
    S1 = {5,6,7}
    S2 = {s for s in range(-60, 61) if s != 0 and 60 % s == 0}
    S1 = {s * 2 for s in S1}
    return S1.intersection(S2)
```

5. Characteristic vectors are a useful representation for sets (see Lecture 3 slide 29). The following properties relate bitwise operations on characteristic vectors to set operations:

- $\chi_S \,|\, \chi_T = \chi_{S\cup T}$. Bitwise OR corresponds to unions
- $\chi_S \,\&\, \chi_T = \chi_{S\cap T}$. Bitwise AND corresponds to intersections
- $\sim \chi_S = \chi_{\overline{S}}$. Bitwise NOT corresponds to complements
- $\chi_S \,\&\, (\sim \chi_T) = \chi_{S\setminus T}$ Bitwise AND with NOT corresponds to difference

Let us illustrate how this works with the following:

- The universe is $U = \{3, 2, 1, 0\}$ so the characteristic vectors will be 4 bits long, and we will have the bits in the order that the elements appear above, so that $\chi_{\{3\}} = 1000$.
- Calculate $\{2, 1\} \cup \{3, 2\}$ by first converting the sets to characteristic vectors, applying an appropriate bitwise operation, and then converting the resulting bit string back into a set.

Show your work.

$x_s|x_t = x\ s \cup t$

$= 0110 \mid 1100$

$= 1110$

$S = \{3, 2, 1\}$

6. Write a Python function that take three sets as arguments, A, B, C, and outputs (A ∩ B) \ C

```
def getSet(a, b, c):

    return a.intersection(b).difference(c)
```