NYU Tandon Bridge Winter 2021: Homework 1

Kamel Bassel M Gazzaz

January 16, 2020

Question 1

Part A. Convert the following numbers to their decimal representation

1. $(10011011)_2$

We know that each bit in position k in a binary number represents 2^k. In general, we have:
$$(i...cba)_2 = [(a * 2^0) + (b * 2^1) + (c * 2^2) ... + (i * 2^n)]_{10}$$

Where n is the position of the final digit. So, we get:
$$(10011011)_2 = [(1 * 2^0) + (1 * 2^1) + (0 * 2^2) + (1 * 2^3) + (1 * 2^4) + (0 * 2^5) + (0 * 2^6) + (1 * 2^7)]_{10}$$

$$\boxed{(10011011)_2 = (155)_{10}}$$

2. $(456)_7$

We know that each digit in position k in a base 7 number represents 7^k. In general, we have:
$$(i...cba)_7 = [(a * 7^0) + (b * 7^1) + (c * 7^2) ... + (i * 7^n)]_{10}$$

Where n is the position of the final digit. So, we get:
$$(456)_7 = [(6 * 7^0) + (5 * 7^1) + (4 * 7^2)]_{10}$$

$$\boxed{(456)_7 = (237)_{10}}$$

3. $(38A)_{16}$

We know that each digit in position k in a hexadecimal number represents 16^k. In general, we have:
$$(i...cba)_{16} = [(a * 16^0) + (b * 16^1) + (c * 16^2) ... + (i * 16^n)]_{10}$$

Where n is the position of the final digit. So, we get:

$$(38A)_{16} = [(A * 16^0) + (8 * 16^1) + (3 * 16^2)]$$

We know that hexadecimal representation uses the digits (0,9) and (A,F) to represent (0,15).

So, A = 10

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**

$(38A)_{16} = [(10 * 16^0) + (8 * 16^1) + (3 * 16^2)]$

$(38A)_{16} = (906)_{10}$

4. $(2214)_5$

We know that each digit in position k in a base 5 number represents 5^k. In general, we have:
$(i...cba)_5 = [(a * 5^0) + (b * 5^1) + (c * 5^2) ... + (i * 5^n)]_{10}$

Where n is the position of the final digit. So, we get:
$(2214)_5 = [(4 * 5^0) + (1 * 5^1) + (2 * 5^2) + (2 * 5^3)]_{10}$

$(2214)_5 = (309)_{10}$

Part B. Convert the following numbers to their binary representation

1. $(69)_{10}$

We know that binary numbers group every 2 elements together into a successive digit. Using the successive division method, we can repeatedly divide by 2 where and the remainder will represent that bit. So:

$(69)_{10} = (\_\_\_\_\_\_\_\_\_\_)_2$

$69 \div 2 = 34\ R\ 1$ implies that $(69)_{10} = (\_\_\_\_\_\_\_1)_2$

$34 \div 2 = 17\ R\ 0$ implies that $(69)_{10} = (\_\_\_\_\_\_01)_2$

$17 \div 2 = 8\ R\ 1$ implies that $(69)_{10} = (\_\_\_\_101)_2$

$8 \div 2 = 4\ R\ 0$ implies that $(69)_{10} = (\_\_\_0101)_2$

$4 \div 2 = 2\ R\ 0$ implies that $(69)_{10} = (\_\_00101)_2$

$2 \div 2 = 1\ R\ 0$ implies that $(69)_{10} = (\_000101)_2$

$1 \div 2 = 0\ R\ 1$ implies that $(69)_{10} = (1000101)_2$

$\boxed{(69)_{10} = (1000101)_2}$

2. $(485)_{10}$

We know that binary numbers group every 2 elements together into a successive digit. Using the successive division method, we can repeatedly divide by 2 where and the remainder will represent that bit. So:

$(485)_{10} = (\_\_\_\_\_\_\_\_\_\_)_2$

$485 \div 2 = 242\ R\ 1$ implies that $(485)_{10} = (\_\_\_\_\_\_\_\_1)_2$

$242 \div 2 = 121\ R\ 0$ implies that $(485)_{10} = (\_\_\_\_\_\_\_01)_2$

$121 \div 2 = 60\ R\ 1$ implies that $(485)_{10} = (\_\_\_\_\_\_101)_2$

$60 \div 2 = 30\ R\ 0$ implies that $(485)_{10} = (\_\_\_\_\_0101)_2$

$30 \div 2 = 15\ R\ 0$ implies that $(485)_{10} = (\_\_\_\_00101)_2$

$15 \div 2 = 7\ R\ 1$ implies that $(485)_{10} = (\_\_\_100101)_2$

$7 \div 2 = 3\ R\ 1$ implies that $(485)_{10} = (\_\_1100101)_2$

$3 \div 2 = 1\ R\ 1$ implies that $(485)_{10} = (\_11100101)_2$

$1 \div 2 = 0\ R\ 1$ implies that $(485)_{10} = (111100101)_2$

$(485)_{10} = (111100101)_2$

3. $(6D1A)_{16}$

We know that for converting directly between hexadecimal values and binary, each hexadecimal value is represented by a 4 bit binary value. This is due to the fact that $16 = 2^4$. So we get:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

A = 1010

1 = 0001

D = 1101

6 = 0110

Combining these values, we get:

$(6D1A)_{16}$ = $(0110\ 1101\ 0001\ 1010)_2$

Part C. Convert the following numbers to their hexadecimal representation

1. $(1101011)_2$

We know that for converting directly between hexadecimal values and binary, each hexadecimal value is represented by a 4 bit binary value. This is due to the fact that $16 = 2^4$. So we get:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Since $(1101011)_2$ is made up of 7 bits, we add a 0 to the left to get groups of 4 bits.

So: $(1101011)_2 = (0110\ 1011)_2$

From the table, we get:

- 0110 = 6
- 1011 = B

$\boxed{\text{Therefore, } (1101011)_2 = (6B)_{16}}$

2. $(895)_{10}$

$(895)_{10} = (\_\_\_ \dots \_\_\_\ \_\_\_\ \_\_\_\ \_\_\_)_{16}$

^ ^ ^ ^ ^

16^n 16^3 16^2 16^1 16^0

To calculate the hexadecimal value of $(895)_{10}$, we need to figure out the largest power of 16 that 895 fits into at least once:

- $895 \div 16^0 = 895\ R\ 0$
- $895 \div 16^1 = 55\ R\ 15$
- $895 \div 16^2 = 3\ R\ 127$
- $895 \div 16^3 = 0\ R\ 895$

Therefore, 16^2 is the largest digit that 895 can fill. This makes sense given that $16^2 = 256 < 895$ and $16^3 = 4096 > 895$. So:

$(895)_{10} = (\_\_\_\ \_\_\_\ \_\_\_)_{16}$

^ ^ ^

16^2 16^1 16^0

The first digit is $895 \div 16^2 = 3\ R\ 127$

$(895)_{10} = (\ 3\ \_\_\_\ \_\_\_)_{16}$

^ ^ ^

16^2 16^1 16^0

The second digit is $127 \div 16^1 = 7\ R\ 15$

$(895)_{10} = (\ 3\ \ 7\ \_\_\_)_{16}$

^ ^ ^

16^2 16^1 16^0

The last digit is $15 \div 16^0 = 15$

We know that hexadecimal representation uses the digits (0,9) and (A,F) to represent (0,15).

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**

Therefore, 15 = F.

$\boxed{(895)_{10} = (37F)_{16}}$

Question 2

1. $(7566)_8 + (4515)_8$

Let's take the sum of each octal digit

$$
\begin{array}{rl}
 & (7566)_8 \\
+ & (4515)_8 \\
 & \text{--------}
\end{array}
$$

Single digit:

$(6)_8 + (5)_8 = (13)_8$ So 1 carries over to the tens and a remainder of 3.

$$
\begin{array}{rcccccc}
 & & & & 1 & & \\
 & (& 7 & 5 & 6 & 6 &)_8 \\
+ & (& 4 & 5 & 1 & 5 &)_8 \\
 & \text{-------------} \\
 & (& & & & 3 &)_8
\end{array}
$$

Tens digit:

$(1)_8 + (6)_8 + (1)_8 = (10)_8$ So 1 carries over to the hundreds and a remainder of 0.

$$
\begin{array}{rcccccc}
 & & & 1 & 1 & & \\
 & (& 7 & 5 & 6 & 6 &)_8 \\
+ & (& 4 & 5 & 1 & 5 &)_8 \\
 & \text{-------------} \\
 & (& & & 0 & 3 &)_8
\end{array}
$$

Hundreds digit:

$(1)_8 + (5)_8 + (5)_8 = (13)_8$ So 1 carries over to the thousands and a remainder of 3.

$$\begin{array}{rl} & \ \ {}^{1\ \ 1\ \ 1} \\ & (7\ \ 5\ \ 6\ \ 6)_8 \\ + & (4\ \ 5\ \ 1\ \ 5)_8 \\ & \text{-------------} \\ & (\quad 3\ \ 0\ \ 3)_8 \end{array}$$

Thousands digit:

$(1)_8 + (7)_8 + (4)_8 = (14)_8$ So 1 carries over to the ten thousands and a remainder of 4.

$$\begin{array}{rl} & \ \ {}^{1\ \ 1\ \ 1} \\ & (\quad 7\ \ 5\ \ 6\ \ 6)_8 \\ + & (\quad 4\ \ 5\ \ 1\ \ 5)_8 \\ & \text{-------------} \\ & (\quad 4\ \ 3\ \ 0\ \ 3)_8 \end{array}$$

Ten thousands digit:

$(1)_8 + (0)_8 + (0)_8 = (1)_8$ So a remainder of 1.

$$\begin{array}{rl} & \ \ {}^{1\ \ 1\ \ 1} \\ & (\quad 7\ \ 5\ \ 6\ \ 6)_8 \\ + & (\quad 4\ \ 5\ \ 1\ \ 5)_8 \\ & \text{-------------} \\ & (\ 1\ \ 4\ \ 3\ \ 0\ \ 3)_8 \end{array}$$

$$\boxed{(7566)_8 + (4515)_8 = (14303)_8}$$

2. $(10110011)_2 + (1101)_2$

$$
\begin{array}{rr}
 & (10110011)_2 \\
+ & (1101)_2 \\
 & \text{---------------}
\end{array}
$$

Let’s first represent both numbers with the same number of bits.

$$
\begin{array}{rr}
 & (10110011)_2 \\
+ & (00001101)_2 \\
 & \text{---------------}
\end{array}
$$

Now, let’s take the sum of each bit.

Single digit:

$(1)_2 + (1)_2 = (10)_2$ So 1 carries over to the tens digit and remainder of 0.

$$
\begin{array}{cccccccccc}
 & & & & & & & {}^{1} & & \\
 & (1 & 0 & 1 & 1 & 0 & 0 & 1 & 1)_2 \\
+ & (0 & 0 & 0 & 0 & 1 & 1 & 0 & 1)_2 \\
 & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\
 & (& & & & & & & 0)_2
\end{array}
$$

Tens digit:

$(1)_2 + (1)_2 + (0)_2 = (10)_2$ So 1 carries over to the hundreds digit and remainder of 0.

$$
\begin{array}{ccccccccc}
 & & & & & & 1 & 1 & \\
 & (1 & 0 & 1 & 1 & 0 & 0 & 1 & 1)_2 \\
+ & (0 & 0 & 0 & 0 & 1 & 1 & 0 & 1)_2 \\
\hline
 & (& & & & & & 0 & 0)_2
\end{array}
$$

Hundreds digit:

$(1)_2 + (0)_2 + (1)_2 = (10)_2$ So 1 carries over to the thousands digit and remainder of 0.

$$
\begin{array}{ccccccccc}
 & & & & & 1 & 1 & 1 & \\
 & (1 & 0 & 1 & 1 & 0 & 0 & 1 & 1)_2 \\
+ & (0 & 0 & 0 & 0 & 1 & 1 & 0 & 1)_2 \\
\hline
 & (& & & & & 0 & 0 & 0)_2
\end{array}
$$

Thousands digit:

$(1)_2 + (0)_2 + (1)_2 = (10)_2$ So 1 carries over to the ten thousands digit and remainder of 0.

$$
\begin{array}{ccccccccc}
 & & & & 1 & 1 & 1 & 1 & \\
 & (1 & 0 & 1 & 1 & 0 & 0 & 1 & 1)_2 \\
+ & (0 & 0 & 0 & 0 & 1 & 1 & 0 & 1)_2 \\
\hline
 & (& & & & 0 & 0 & 0 & 0)_2
\end{array}
$$

Ten thousands digit:

$(1)_2 + (1)_2 + (0)_2 = (10)_2$ So 1 carries over to the hundred thousands digit and remainder of 0.

$$\begin{array}{ccccccccc}
 & & & 1 & 1 & 1 & 1 & 1 & \\
 & (1 & 0 & 1 & 1 & 0 & 0 & 1 & 1)_2 \\
+ & (0 & 0 & 0 & 0 & 1 & 1 & 0 & 1)_2 \\
\hline
 & (& & & 0 & 0 & 0 & 0 & 0)_2
\end{array}$$

Hundred thousands digit:

$(1)_2 + (1)_2 + (0)_2 = (10)_2$ So 1 carries over to the millions digit and remainder of 0.

$$\begin{array}{ccccccccc}
 & & 1 & 1 & 1 & 1 & 1 & 1 & \\
 & (1 & 0 & 1 & 1 & 0 & 0 & 1 & 1)_2 \\
+ & (0 & 0 & 0 & 0 & 1 & 1 & 0 & 1)_2 \\
\hline
 & (& & 0 & 0 & 0 & 0 & 0 & 0)_2
\end{array}$$

Millions digit:

$(1)_2 + (0)_2 + (0)_2 = (1)_2$ So nothing carries over to the ten millions digit and remainder of 1.

$$\begin{array}{ccccccccc}
 & & 1 & 1 & 1 & 1 & 1 & 1 & \\
 & (1 & 0 & 1 & 1 & 0 & 0 & 1 & 1)_2 \\
+ & (0 & 0 & 0 & 0 & 1 & 1 & 0 & 1)_2 \\
\hline
 & (& 1 & 0 & 0 & 0 & 0 & 0 & 0)_2
\end{array}$$

Ten millions digit:

$(1)_2 + (0)_2 = (1)_2$ So nothing carries over to the hundred millions digit and remainder of 1.

$$
\begin{array}{rl}
 & {}^{1\ 1\ 1\ 1\ 1\ 1} \\
 & (1\ 0\ 1\ 1\ 0\ 0\ 1\ 1)_2 \\
+ & (0\ 0\ 0\ 0\ 1\ 1\ 0\ 1)_2 \\
 & \text{-------------------------} \\
 & (1\ 1\ 0\ 0\ 0\ 0\ 0\ 0)_2
\end{array}
$$

So $(1011\ 0011)_2 + (1101)_2 = (1100\ 0000)_2$

3. $(7A66)_{16} + (45C5)_{16}$

Let's take the sum of each hexadecimal digit

$$
\begin{array}{rl}
 & (7A66)_{16} \\
+ & (45C5)_{16} \\
 & \text{---------}
\end{array}
$$

Single digit:

$(6)_{16} + (5)_{16} = (11)_{16} = (B)_{16}$

We know that hexadecimal representation uses the digits (0,9) and (A,F) to represent (0,15).

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**

$$
\begin{array}{rl}
 & (7\ \ A\ \ 6\ \ 6)_{16} \\
+ & (4\ \ 5\ \ C\ \ 5)_{16} \\
 & \text{-------------} \\
 & (\qquad\quad B)_{16}
\end{array}
$$

Tens digit:

We know that hexadecimal representation uses the digits (0,9) and (A,F) to represent (0,15).

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**

$(6)_{16} + (C)_{16} = (6)_{16} + (\text{“12”})_{16} = (12)_{16}$ So 1 carries over to the next digit and 2 remainder.

$$\begin{array}{rl} & \quad\;\; {\scriptstyle 1} \\ & (7\;\; A\;\; 6\;\; 6)_{16} \\ + & (4\;\; 5\;\; C\;\; 5)_{16} \\ & \text{-------------} \\ & (\qquad 2\;\; B)_{16} \end{array}$$

Hundreds digit:

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**

$(1)_{16} + (A)_{16} + (5)_{16} = (1)_{16} + (\text{“10”})_{16} + (5)_{16} = (10)_{16}$ So 1 carries over to the next digit and 0 remainder.

$$\begin{array}{rl} & \;\; {\scriptstyle 1}\;\;\; {\scriptstyle 1} \\ & (7\;\; \cancel{A}\;\; 6\;\; 6)_{16} \\ + & (4\;\; 5\;\; C\;\; 5)_{16} \\ & \text{-------------} \\ & (\quad 0\;\; 2\;\; B)_{16} \end{array}$$

Thousands digit:

$(1)_{16} + (7)_{16} + (4)_{16} = (\text{“12”})_{16} = (C)_{16}$

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**

$$
\begin{array}{rl}
 & (\overset{1}{\cancel{7}}\ \overset{1}{\cancel{A}}\ 6\ 6)_{16} \\
+ & (4\ 5\ C\ 5)_{16} \\
\hline
 & (C\ 0\ 2\ B)_{16}
\end{array}
$$

$\boxed{\text{So, } (7A66)_{16} + (45C5)_{16} = (C02B)_{16}}$

4. $(3022)_5 - (2433)_5$

Let's subtract each digit:

$$
\begin{array}{rl}
 & (3022)_5 \\
- & (2433)_5 \\
\hline
\end{array}
$$

Single digit:

$(2)_5 - (3)_5 < 0$ Therefore, we borrow a group of 5 (base 5 ten) from the next digit.

$$
\begin{array}{rl}
 & (3\ 0\ \overset{1}{\cancel{2}}\ \overset{10}{2})_5 \\
- & (2\ 4\ 3\ 3)_5 \\
\hline
\end{array}
$$

$(12)_5 - (3)_5 = (4)_5$

$$
\begin{array}{rl}
 & (3\ 0\ \overset{1}{\cancel{2}}\ \overset{10}{2})_5 \\
- & (2\ 4\ 3\ 3)_5 \\
\hline
 & (\qquad\ \ 4)_5
\end{array}
$$

Tens digit:

$(1)_5 - (3)_5 < 0$, so we borrow $(10)_5$ from the hundreds digit.

But the hundreds digit is 0 so we borrow $(10)_5$ from the thousands digit for that digit first to give to the tens digit.

$$\begin{array}{cccccc} & & 2 & 10 & 1 & 10 \\ & (& \cancel{3} & \cancel{0} & \cancel{2} & 2)_5 \\ - & (& 2 & 4 & 3 & 3)_5 \\ \hline & (& & & 3 & 4)_5 \end{array}$$

Then we give $(10)_5$ from the hundreds digit to the tens digit.

$$\begin{array}{cccccc} & & & 4 & & \\ & & 2 & \cancel{10} & 11 & 10 \\ & (& \cancel{3} & \cancel{0} & \cancel{2} & 2)_5 \\ - & (& 2 & 4 & 3 & 3)_5 \\ \hline & (& & & 3 & 4)_5 \end{array}$$

Hundreds digit:

$(4)_5 - (4)_5 = (0)_5$

$$\begin{array}{cccccc} & & & 4 & & \\ & & 2 & \cancel{10} & 11 & 10 \\ & (& \cancel{3} & \cancel{0} & \cancel{2} & 2)_5 \\ - & (& 2 & 4 & 3 & 3)_5 \\ \hline & (& & 0 & 3 & 4)_5 \end{array}$$

Thousands digit:

$(2)_5 - (2)_5 = (0)_5$

$$\begin{array}{cccccc} & & & 4 & & \\ & & 2 & \cancel{10} & 11 & 10 \\ & (& \cancel{3} & \cancel{0} & \cancel{2} & 2)_5 \\ - & (& 2 & 4 & 3 & 3)_5 \\ \hline & (& 0 & 0 & 3 & 4)_5 \end{array}$$

$$\boxed{\text{So, } (3022)_5 - (2433)_5 = (34)_5}$$

Question 3

Part A. Convert the following numbers to their 8-bits two's complement representation.

1. $(124)_{10}$

Step 1: Determine if the number is positive or negative.

$(124)_{10} > 0$

Since the decimal number is positive, we do not need to take its inverse. We can simply convert it to binary.

Step 2: We will use the successive division method to convert to binary.

$124 \div 2 = 62\,R\,0$

$62 \div 2 = 31\,R\,0$

$31 \div 2 = 15\,R\,1$

$15 \div 2 = 7\,R\,1$

$7 \div 2 = 3\,R\,1$

$3 \div 2 = 1\,R\,1$

$1 \div 2 = 0\,R\,1$

So $(124)_{10} = (1111100)_2$

Step 3: Convert binary number into 8 bit 2's complement.

As is, the binary number is 7 bits long. So we pad the number with a leftmost 0 to indicate that it is a positive number and complete 8 bits. We get:

$(111\ 1100)_2 = (0111\ 1100)_{8\text{ bit 2's complement}}$

$\boxed{\text{Therefore, } (124)_{10} = (0111\ 1100)_{8\text{ bit 2's complement}}}$

2. $(-124)_{10}$

Step 1: Determine if positive or negative number.

$(-124)_{10} < 0$

So we take its positive inverse $(124)_{10}$ and convert it to binary then take its complement.

Step two: Convert $(124)_{10}$ to binary

In the previous question, we established that $(124)_{10} = (0111\ 1100)_{8 \text{ bit 2's complement}}$

Step 3: Take the inverse 8 bit 2's complement to represent the negative number.

We know that an 8 bit 2's complement and its inverse must sum to 2^k where k is the bit size.

So $2^k = 2^8 = (1\ 0000\ 0000)_2$

So we set up the equation:

$$
\begin{array}{rl}
 & (0111\ 1100)_{8 \text{ bit 2's complement}} \\
+ & (\qquad\qquad)_{8 \text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Singles digit:

We need $(0)_2 + (x)_2 = (0)_2$, so x = 0.

$$
\begin{array}{rl}
 & (0111\ 1100)_{8 \text{ bit 2's complement}} \\
+ & (\qquad\quad 0)_{8 \text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Tens digit:

We need $(0)_2 + (x)_2 = (0)_2$, so x = 0.

$$
\begin{array}{rl}
 & (0111\ 1100)_{8 \text{ bit 2's complement}} \\
+ & (\qquad\quad 00)_{8 \text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Hundreds digit:

We need $(1)_2 + (x)_2 = (0)_2$, so x = 1 and we get $(1)_2 + (1)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rr}
 & {\scriptstyle 1} \\
 & (0111\ 1100)_{8\text{ bit 2's complement}} \\
+ & (\qquad 100)_{8\text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Thousands digit:

We need $(1)_2 + (1)_2 + (x)_2 = (0)_2$, so x = 0 and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rr}
 & {\scriptstyle 1\ 1} \\
 & (0111\ 1100)_{8\text{ bit 2's complement}} \\
+ & (\qquad 0100)_{8\text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Ten thousands digit:

We need $(1)_2 + (1)_2 + (x)_2 = (0)_2$, so x = 0 and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rr}
 & {\scriptstyle 1\,1\ \ 1} \\
 & (0111\ 1100)_{8\text{ bit 2's complement}} \\
+ & (\quad 0\ 0100)_{8\text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Hundred thousands digit:

We need $(1)_2 + (1)_2 + (x)_2 = (0)_2$, so x = 0 and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rr}
 & {\scriptstyle 1\,1\,1\ \ 1} \\
 & (0111\ 1100)_{8\text{ bit 2's complement}} \\
+ & (\ 00\ 0100)_{8\text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Millions digit:

We need $(1)_2 + (1)_2 + (x)_2 = (0)_2$, so x = 0 and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rl}
 & 1\,1\,1\,1\;\;1 \\
 & (0111\ 1100)_{8\text{ bit 2's complement}} \\
+ & (\ 000\ 0100)_{8\text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Ten millions digit:

We need $(1)_2 + (0)_2 + (x)_2 = (0)_2$, so x = 1 and we get $(1)_2 + (0)_2 + (1)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rl}
 & 1\,1\,1\,1\,1\;\;1 \\
 & (\ 0111\ 1100)_{8\text{ bit 2's complement}} \\
+ & (\ 1000\ 0100)_{8\text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Hundred millions digit:

We need $(1)_2 + (0)_2 + (x)_2 = (1)_2$, so x = 0 and we get $(1)_2 + (0)_2 + (0)_2 = (1)_2$.

So, the inverse of $(0111\ 1100)_{8\text{ bit 2's complement}}$ is $(1000\ 0100)_{8\text{ bit 2's complement}}$.

$\boxed{\text{Therefore, } (-124)_{10} = (1000\ 0100)_{8\text{ bit 2's complement}}}$

3. $(109)_{10}$

Step 1: Determine if positive or negative decimal value

$(109)_{10} > 0$, so we can convert the number as is into binary without changing the sign.

Step 2: Convert the positive decimal number to binary

We will use the successive divisions method:

- $109 \div 2 = 54\ R\ 1$
- $54 \div 2 = 27\ R\ 0$
- $27 \div 2 = 13\ R\ 1$

- $13 \div 2 = 6\,R\,1$
- $6 \div 2 = 3\,R\,0$
- $3 \div 2 = 1\,R\,1$
- $1 \div 2 = 0\,R\,1$

Therefore, $(109)_{10} = (1101101)_2$

Step 3: Convert the binary number to 8 bit 2's complement.

As is, the binary number is 7 bits long. So we pad the number with a leftmost 0 to indicate that it is a positive number and complete 8 bits. We get:

$(110\ 1101)_2 = (0110\ 1101)_{8\text{ bit 2's complement}}$

$\boxed{\text{Therefore, } (109)_{10} = (0110\ 1101)_{8\text{ bit 2's complement}}}$

4. $(-79)_{10}$

Step 1: Determine if positive or negative decimal value.

$(-79)_{10} < 0$, so we take its inverse value $(79)_{10}$ and convert it into binary.

Step 2: Convert the positive value $(79)_{10}$ to binary.

We will use the successive divisions method:

- $79 \div 2 = 39\,R\,1$
- $39 \div 2 = 19\,R\,1$
- $19 \div 2 = 9\,R\,1$
- $9 \div 2 = 4\,R\,1$
- $4 \div 2 = 2\,R\,0$
- $2 \div 2 = 1\,R\,0$
- $1 \div 2 = 0\,R\,1$

So, $(79)_{10} = (1001111)_2$

Step 3: Convert the binary number to 8 bit 2’s complement.

As is, the binary number is 7 bits long. So we pad the number with a leftmost 0 to indicate that it is a positive number and complete 8 bits. We get:

So, $(79)_{10} = (0100\ 1111)_2$

Step 4: Take the 8 bit 2’s complement’s inverse

We know that an 8 bit 2’s complement and its inverse must sum to 2^k where k is the bit size.

So $2^k = 2^8 = (1\ 0000\ 0000)_2$

So we set up the equation:

$$
\begin{array}{rl}
 & (0100\ 1111)_{8 \text{ bit 2's complement}} \\
+ & (\qquad\qquad)_{8 \text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Singles digit:

We need $(1)_2 + (x)_2 = (0)_2$, so x = 1 and we get $(1)_2 + (1)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rl}
 & (0100\ 11\overset{1}{1}1)_{8 \text{ bit 2's complement}} \\
+ & (\qquad\qquad 1)_{8 \text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Tens digit:

We need $(1)_2 + (1)_2 + (x)_2 = (0)_2$, so x = 0 and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rl}
 & (0100\ 1\overset{1}{1}\overset{1}{1}1)_{8 \text{ bit 2's complement}} \\
+ & (\qquad\qquad 01)_{8 \text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Hundreds digit:

We need $(1)_2 + (1)_2 + (x)_2 = (0)_2$, so x = 0 and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$\begin{array}{rl} & {}^{1\,1\,1} \\ & (0100\ 1111)_{\text{8 bit 2's complement}} \\ + & (\qquad 001)_{\text{8 bit 2's complement}} \\ \hline & (1\ 0000\ 0000)_2 \end{array}$$

Thousands digit:

We need $(1)_2 + (1)_2 + (x)_2 = (0)_2$, so x = 0 and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$\begin{array}{rl} & {}^{1\ \ 1\,1\,1} \\ & (0100\ 1111)_{\text{8 bit 2's complement}} \\ + & (\qquad 0001)_{\text{8 bit 2's complement}} \\ \hline & (1\ 0000\ 0000)_2 \end{array}$$

Ten thousands digit:

We need $(1)_2 + (0)_2 + (x)_2 = (0)_2$, so x = 1 and we get $(1)_2 + (0)_2 + (1)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$\begin{array}{rl} & {}^{1\,1\ \ 1} \\ & (0100\ 1111)_{\text{8 bit 2's complement}} \\ + & (\quad 1\ 0001)_{\text{8 bit 2's complement}} \\ \hline & (1\ 0000\ 0000)_2 \end{array}$$

Hundred thousands digit:

We need $(1)_2 + (0)_2 + (x)_2 = (0)_2$, so x = 1 and we get $(1)_2 + (0)_2 + (1)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$\begin{array}{rl} & {}^{1\,1\,1\ \ 1} \\ & (0100\ 1111)_{\text{8 bit 2's complement}} \\ + & (\ \ 11\ 0001)_{\text{8 bit 2's complement}} \\ \hline & (1\ 0000\ 0000)_2 \end{array}$$

Millions digit:

We need $(1)_2 + (1)_2 + (x)_2 = (0)_2$, so x = 0 and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rl}
 & {}^{1\,1\,1\,1\ \ 1} \\
 & (0100\ 1111)_{8\text{ bit 2's complement}} \\
+ & (\ 011\ 0001)_{8\text{ bit 2's complement}} \\
 & \text{---------------} \\
 & (1\ 0000\ 0000)_2
\end{array}
$$

Ten millions digit:

We need $(1)_2 + (0)_2 + (x)_2 = (0)_2$, so x = 1 and we get $(1)_2 + (0)_2 + (1)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rl}
 & {}^{1\,1\,1\,1\,1\ \ 1} \\
 & (\ 0100\ 1111)_{8\text{ bit 2's complement}} \\
+ & (\ 1011\ 0001)_{8\text{ bit 2's complement}} \\
 & \text{---------------} \\
 & (1\ 0000\ 0000)_2
\end{array}
$$

Hundred millions digit:

We need $(1)_2 + (0)_2 + (x)_2 = (1)_2$, so x = 0 and we get $(1)_2 + (0)_2 + (0)_2 = (1)_2$.

So, the inverse of $(0100\ 1111)_{8\text{ bit 2's complement}}$ is $(1011\ 0001)_{8\text{ bit 2's complement}}$.

Therefore, $(-79)_{10} = (1011\ 0001)_{8\text{ bit 2's complement}}$

Part B. Convert the following numbers (represented as 8-bit two's complement) to their decimal representation.

1. $(00011110)_{8\text{ bit 2's complement}}$

Step 1: Determine if positive or negative 8 bit 2's complement value.

$(00011110)_{8\text{ bit 2's complement}}$ is padded with a leftmost 0, so it's a positive number. As such, we can convert it directly (as a binary) to decimal.

Step 2: Convert the binary number to decimal

$$(00011110)_{8\ bit\ 2's\ complement} = (00011110)_2$$

$(00011110)_{8\ bit\ 2's\ complement} = [(0 * 2^0) + (1 * 2^1) + (1 * 2^2) + (1 * 2^3) + (1 * 2^4) + (0 * 2^5) + (0 * 2^6) + (0 * 2^7)]$

$\boxed{(00011110)_{8\ bit\ 2's\ complement} = (30)_{10}}$

2. $(1110\ 0110)_{8\text{ bit 2's complement}}$

Step 1: Determine if positive or negative 8 bit 2's complement value.

$(1110\ 0110)_{8\text{ bit 2's complement}}$ is not padded with a leftmost 0, so it's a negative number. As such, we need to get its inverse complement, and convert that to a decimal value, then add the negative sign.

Step 2: Get the complement's inverse.

We know that an 8 bit 2's complement and its inverse must sum to 2^k where k is the bit size.

So $2^k = 2^8 = (1\ 0000\ 0000)_2$

So we set up the equation:

$$\begin{array}{rl} & (1110\ 0110)_{8\text{ bit 2's complement}} \\ + & (\qquad\qquad)_{8\text{ bit 2's complement}} \\ \hline & (1\ 0000\ 0000)_2 \end{array}$$

Singles digit:

We need $(0)_2 + (x)_2 = (0)_2$, so x = 0 and we get $(0)_2 + (0)_2 = (0)_2$.

$$\begin{array}{rl} & (1110\ 0110)_{8\text{ bit 2's complement}} \\ + & (\qquad\qquad 0)_{8\text{ bit 2's complement}} \\ \hline & (1\ 0000\ 0000)_2 \end{array}$$

Tens digit:

We need $(1)_2 + (x)_2 = (0)_2$, so x = 1 and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$\begin{array}{rr} & \overset{}{(1110\ 0\overset{1}{1}10)}_{8\text{ bit 2's complement}} \\ + & (\qquad\qquad 10)_{8\text{ bit 2's complement}} \\ \hline & (1\ 0000\ 0000)_2 \end{array}$$

Hundreds digit:

We need $(1)_2 + (1)_2 + (x)_2 = (0)_2$, so x = 0 and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$\begin{array}{rr} & (1110\ \overset{1}{0}\overset{1}{1}10)_{8\text{ bit 2's complement}} \\ + & (\qquad\qquad 010)_{8\text{ bit 2's complement}} \\ \hline & (1\ 0000\ 0000)_2 \end{array}$$

Thousands digit:

We need $(1)_2 + (0)_2 + (x)_2 = (0)_2$, so x = 1 and we get $(1)_2 + (0)_2 + (1)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$\begin{array}{rr} & (111\overset{1}{0}\ \overset{1}{0}\overset{1}{1}10)_{8\text{ bit 2's complement}} \\ + & (\qquad\qquad 1010)_{8\text{ bit 2's complement}} \\ \hline & (1\ 0000\ 0000)_2 \end{array}$$

Ten thousands digit:

We need $(1)_2 + (0)_2 + (x)_2 = (0)_2$, so x = 1 and we get $(1)_2 + (0)_2 + (1)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$\begin{array}{rr} & (11\overset{1}{1}\overset{1}{0}\ \overset{1}{0}\overset{1}{1}10)_{8\text{ bit 2's complement}} \\ + & (\quad 1\ 1010)_{8\text{ bit 2's complement}} \\ \hline & (1\ 0000\ 0000)_2 \end{array}$$

Hundred thousands digit:

We need $(1)_2 + (1)_2 + (x)_2 = (0)_2$, so x = 0 and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rl}
 & \overset{111\ \ 11}{(1110\ 0110)}_{8\text{ bit 2's complement}} \\
+ & (\quad 01\ 1010)_{8\text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Millions digit:

We need $(1)_2 + (1)_2 + (x)_2 = (0)_2$, so x = 0 and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rl}
 & \overset{1111\ \ 11}{(1110\ 0110)}_{8\text{ bit 2's complement}} \\
+ & (\ 001\ 1010)_{8\text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Ten millions digit:

We need $(1)_2 + (1)_2 + (x)_2 = (0)_2$, so x = 0 and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rl}
 & (\overset{11111\ \ 11}{1110\ 0110})_{8\text{ bit 2's complement}} \\
+ & (\ \ 0001\ 1010)_{8\text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Hundred millions digit:

We need $(1)_2 + (0)_2 + (x)_2 = (1)_2$, so x = 0 and we get $(1)_2 + (0)_2 + (0)_2 = (1)_2$.

So, the inverse of $(1110\ 0110)_{8\text{ bit 2's complement}}$ is $(0001\ 1010)_{8\text{ bit 2's complement}}$

Step 3: Get the decimal value of $(0001\ 1010)_{8\text{ bit 2's complement}}$

$$
\begin{aligned}
(0001\ 1010)_{8\ bit\ 2's\ complement} &= (0001\ 1010)_2 \\
(0001\ 1010)_{8\ bit\ 2's\ complement} &= [(0 * 2^0) + (1 * 2^1) + (0 * 2^2) + (1 * 2^3) + (1 * 2^4) + (0 * 2^5) + (0 * 2^6) + (0 * 2^7)] \\
(0001\ 1010)_{8\ bit\ 2's\ complement} &= (26)_{10}
\end{aligned}
$$

Step 4: Add the negative sign.

$\boxed{(1110\ 0110)_{8\text{ bit 2's complement}} = (-26)_{10}}$

3. $(00101101)_{8\text{ bit 2's comp}}$

Step 1: Determine if positive or negative 8 bit 2's complement value.

$(00101101)_{8\text{ bit 2's complement}}$ is padded with a leftmost 0, so it's a positive number. As such, we can convert it directly (as a binary) to decimal.

Step 2: Convert the binary number to decimal

$$(0010\ 1101)_{8\ bit\ 2's\ complement} = (0010\ 1101)_2$$

$$(0010\ 1101)_{8\ bit\ 2's\ complement} = [(1 * 2^0) + (0 * 2^1) + (1 * 2^2) + (1 * 2^3) + (0 * 2^4) + (1 * 2^5) + (0 * 2^6) + (0 * 2^7)]$$

$$\boxed{(0010\ 1101)_{8\ bit\ 2's\ complement} = (45)_{10}}$$

4. $(1001\ 1110)_{8\text{ bit 2's comp}}$

Step 1: Determine if positive or negative 8 bit 2's complement value.

$(1001\ 1110)_{8\text{ bit 2's complement}}$ is not padded with a leftmost 0, so it's a negative number. As such, we need to get its inverse complement, and convert that to a decimal value, then add the negative sign.

Step 2: Get the complement's inverse.

We know that an 8 bit 2's complement and its inverse must sum to 2^k where k is the bit size.

So $2^k = 2^8 = (1\ 0000\ 0000)_2$

So we set up the equation:

$$\begin{array}{rl} & (1001\ 1110)_{8\text{ bit 2's complement}} \\ + & (\qquad\qquad)_{8\text{ bit 2's complement}} \\ \hline & (1\ 0000\ 0000)_2 \end{array}$$

Singles digit:

We need $(0)_2 + (x)_2 = (0)_2$, so x = 0 and we get $(0)_2 + (0)_2 = (0)_2$.

$$
\begin{array}{rr}
 & (1001\ 1110)_{8\text{ bit 2's complement}} \\
+ & (\qquad\quad 0)_{8\text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Tens digit:

We need $(1)_2 + (x)_2 = (0)_2$, so x = 1 and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rr}
 & {}^{1} \\
 & (1001\ 1110)_{8\text{ bit 2's complement}} \\
+ & (\qquad\quad 10)_{8\text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Hundreds digit:

We need $(1)_2 + (1)_2 + (x)_2 = (0)_2$, so x = 0 and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rr}
 & {}^{1\,1} \\
 & (1001\ 1110)_{8\text{ bit 2's complement}} \\
+ & (\qquad\quad 010)_{8\text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Thousands digit:

We need $(1)_2 + (1)_2 + (x)_2 = (0)_2$, so x = 0 and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rr}
 & {}^{1}{}^{1\,1} \\
 & (1001\ 1110)_{8\text{ bit 2's complement}} \\
+ & (\qquad 0010)_{8\text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Ten thousands digit:

We need $(1)_2 + (1)_2 + (x)_2 = (0)_2$, so x = 0 and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rl}
 & {\scriptstyle 1\,1\ \ 1\,1} \\
 & (1001\ 1110)_{8\text{ bit 2's complement}} \\
+ & (\quad 0\ 0010)_{8\text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Hundred thousands digit:

We need $(1)_2 + (0)_2 + (x)_2 = (0)_2$, so $x = 1$ and we get $(1)_2 + (0)_2 + (1)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rl}
 & {\scriptstyle 1\,1\,1\ \ 1\,1} \\
 & (1001\ 1110)_{8\text{ bit 2's complement}} \\
+ & (\quad 10\ 0010)_{8\text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Millions digit:

We need $(1)_2 + (0)_2 + (x)_2 = (0)_2$, so $x = 1$ and we get $(1)_2 + (0)_2 + (1)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rl}
 & {\scriptstyle 1\,1\,1\,1\ \ 1\,1} \\
 & (1001\ 1110)_{8\text{ bit 2's complement}} \\
+ & (\ 110\ 0010)_{8\text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Ten millions digit:

We need $(1)_2 + (1)_2 + (x)_2 = (0)_2$, so $x = 0$ and we get $(1)_2 + (1)_2 + (0)_2 = (10)_2$. 1 carries over to the next digit and remainder of 0.

$$
\begin{array}{rl}
 & {\scriptstyle 1\,1\,1\,1\,1\ \ 1\,1} \\
 & (\ 1001\ 1110)_{8\text{ bit 2's complement}} \\
+ & (\ 0110\ 0010)_{8\text{ bit 2's complement}} \\
\hline
 & (1\ 0000\ 0000)_2
\end{array}
$$

Hundred millions digit:

We need $(1)_2 + (0)_2 + (x)_2 = (1)_2$, so $x = 0$ and we get $(1)_2 + (0)_2 + (0)_2 = (1)_2$.

So, the inverse of $(1001\ 1110)_{8\text{ bit 2's complement}}$ is $(0110\ 0010)_{8\text{ bit 2's complement}}$

Step 3: Get the decimal value of $(0110\ 0010)_{8\text{ bit 2's complement}}$

$$(0110\ 0010)_{8\ bit\ 2's\ complement} = (0110\ 0010)_2$$
$$(0110\ 0010)_{8\ bit\ 2's\ complement} = [(0 * 2^0) + (1 * 2^1) + (0 * 2^2) + (0 * 2^3) + (0 * 2^4) + (1 * 2^5) + (1 * 2^6) + (0 * 2^7)]$$
$$(0110\ 0010)_{8\ bit\ 2's\ complement} = (98)_{10}$$

Step 4: Add the negative sign.

$$\boxed{(1001\ 1110)_{8\text{ bit 2's complement}} = (-98)_{10}}$$

Question 4

1. **Exercise 1.2.4**

B. Write a truth table for ¬(p ∨ q)

Steps:
- First fill in the table with all possible values for p and q
- Add in an intermediate column for $p \vee q$. If either p or q are True, then $p \vee q$ is also True
- Finally, take the opposite of $p \vee q$ for $\neg(p \vee q)$

Truth table:

p	q	$p \vee q$	$\neg(p \vee q)$
T	T	T	F
T	F	T	F
F	T	T	F
F	F	F	T

Justification table:

p	q	$p \vee q$	$\neg(p \vee q)$
All possible combinations of p and q	All possible combinations of p and q	Both p and q are True, so it's True	$p \vee q$ is True, so it's False
All possible combinations of p and q	All possible combinations of p and q	p is True, so it's True	$p \vee q$ is True, so it's False
All possible combinations of p and q	All possible combinations of p and q	q is True, so it's True	$p \vee q$ is True, so it's False
All possible combinations of p and q	All possible combinations of p and q	Both p and q are False, so it's False	$p \vee q$ is False, so it's True

C. Write a truth table for r ∨ (p ∧ ¬q)

Steps:

- Fill in the table with all possible values for p, q and r.
- Add in an intermediate column for $\neg q$. If q is True, then $\neg q$ is False and vice versa.
- Add in an intermediate column for $(p \land \neg q)$. If both p and $\neg q$ are True, then $(p \land \neg q)$ is True. Otherwise, it is False.
- Finally, determine $r \lor (p \land \neg q)$. If both r and $(p \land \neg q)$ are False, then it's False. Otherwise it's True.

Truth table:

p	q	r	$\neg q$	$(p \land \neg q)$	$r \lor (p \land \neg q)$
T	T	T	F	F	T
T	T	F	F	F	F
T	F	T	T	T	T
T	F	F	T	T	T
F	T	T	F	F	T
F	T	F	F	F	F
F	F	T	T	F	T
F	F	F	T	F	F

Justification table:

p	q	r	$\neg q$	$(p \wedge \neg q)$	$r \vee (p \wedge \neg q)$
All possible combinations of p, q, and r	All possible combinations of p, q, and r	All possible combinations of p, q, and r	q is True, so it's False	$\neg q$ is False, so it's False	r is True, so it's True
All possible combinations of p, q, and r	All possible combinations of p, q, and r	All possible combinations of p, q, and r	q is True, so it's False	$\neg q$ is False, so it's False	Both r and $(p \wedge \neg q)$ are False, so it's False
All possible combinations of p, q, and r	All possible combinations of p, q, and r	All possible combinations of p, q, and r	q is False, so it's True	Both p and $\neg q$ are True, so it's True	Both r and $(p \wedge \neg q)$ are True, so it's True
All possible combinations of p, q, and r	All possible combinations of p, q, and r	All possible combinations of p, q, and r	q is False, so it's True	Both p and $\neg q$ are True, so it's True	$(p \wedge \neg q)$ is True, so it's True
All possible combinations of p, q, and r	All possible combinations of p, q, and r	All possible combinations of p, q, and r	q is True, so it's False	Both p and $\neg q$ are False, so it's False	r is True, so it's True
All possible combinations of p, q, and r	All possible combinations of p, q, and r	All possible combinations of p, q, and r	q is True, so it's False	Both p and $\neg q$ are False, so it's False	Both r and $(p \wedge \neg q)$ are False, so it's False
All possible combinations of p, q, and r	All possible combinations of p, q, and r	All possible combinations of p, q, and r	q is False, so it's True	p is False, so it's False	r is True, so it's True
All possible combinations of p, q, and r	All possible combinations of p, q, and r	All possible combinations of p, q, and r	q is False, so it's True	p is False, so it's False	Both r and $(p \wedge \neg q)$ are False, so it's False

2. **Exercise 1.3.4**

B. Write a truth table for (p → q) → (q → p)

Steps:

- Fill in the table with all possible values for p and q.
- Add in an intermediate column for (p → q). If p is True and q is False then (p → q) is False. Otherwise, it is True.
- Add in an intermediate column for (q → p). If q is True and p is False then (q → p) is False. Otherwise, it is True.
- Finally, determine (p → q) → (q → p). If (p → q) is True and (q → p) is False then (p → q) → (q → p) is False. Otherwise, it is True.

Truth table:

p	q	(p → q)	(q → p)	(p → q) → (q → p)
T	T	T	T	T
T	F	F	T	T
F	T	T	F	F
F	F	T	T	T

Justification table:

p	q	(p → q)	(q → p)	(p → q) → (q → p)
Combination of p and q	Combination of p and q	Both p and q are True, so it's True	Both p and q are True, so it's True	Both (p → q) and (q → p) are True, so it's True
Combination of p and q	Combination of p and q	p is True, but q is False, so it's False	q is False, so it's True	(p → q) is False, so it's True
Combination of p and q	Combination of p and q	p is False, so it's True	q is True, but p is False, so it's False	(p → q) is True, but (q → p) is False, so it's False
Combination of p and q	Combination of p and q	Both p and q are False, so it's True	Both p and q are False, so it's True	Both (p → q) and (q → p) are True, so it's True

D. Write a truth table for (p ↔ q) ⊕ (p ↔ ¬q)

Steps:

- Fill in the table with all possible values for p and q.
- Add an intermediate column for $\neg q$. If q is True, then $\neg q$ is False and vice versa.
- Add in an intermediate column for (p ↔ q). If p and q have the same truth value, then (p ↔ q) is True. Otherwise, it is False.
- Add in an intermediate column for (p ↔ ¬q). If p and ¬q have the same truth value, then (p ↔ ¬q) is True. Otherwise, it is False.
- Finally, determine (p ↔ q) ⊕ (p ↔ ¬q). If (p ↔ q) is True or (p ↔ ¬q) is True, but not both, then (p ↔ q) ⊕ (p ↔ ¬q) is True. Otherwise, it is False.

Truth table:

p	q	¬q	(p ↔ q)	(p ↔ ¬q)	(p ↔ q) ⊕ (p ↔ ¬q)
T	T	F	T	F	T
T	F	T	F	T	T
F	T	F	F	T	T
F	F	T	T	F	T

Justification table:

p	q	¬q	(p ↔ q)	(p ↔ ¬q)	(p ↔ q) ⊕ (p ↔ ¬q)
Combination of p and q	Combination of p and q	q is True, so it's False	Both p and q are True, so it's True	p is True but ¬q is False, so False	(p ↔ q) is True, but (p ↔ ¬q) is False, so it's True
Combination of p and q	Combination of p and q	q is False, so it's True	p is True but q is False, so False	Both p and ¬q are True, so True	(p ↔ q) is False, but (p ↔ ¬q) is True, so it's True
Combination of p and q	Combination of p and q	q is True, so it's False	p is False but q is True, so False	Both p and ¬q are False, so True	(p ↔ q) is False, but (p ↔ ¬q) is True, so it's True
Combination of p and q	Combination of p and q	q is False, so it's True	Both p and q are False, so it's True	p is False but ¬q is True, so False	(p ↔ q) is True, but (p ↔ ¬q) is False, so it's True

Question 5

1. **Exercise 1.2.7**

In order to apply for a credit card:

B: Applicant presents a birth certificate.
D: Applicant presents a driver's license.
M: Applicant presents a marriage license.

b) The applicant must present at least two of the following forms of identification: birth certificate, driver's license, marriage license.

Step 1: Determine how to satisfy the condition in English.

In order to satisfy the condition, the applicant can present any two forms of identification from the list. The possible combinations are:

- Birth certificate and driver's license
- Driver's license and marriage license
- Birth certificate and marriage license

Step 2: Translate the possibilities to logical language.

- Birth certificate and driver's license: $(B \wedge D)$
- Driver's license and marriage license $(D \wedge M)$
- Birth certificate and marriage license $(B \wedge M)$

Step 3: Formulate the coherent expression.

In order to satisfy the condition, any of the 3 combinations must be True. In other words:

- If $(B \wedge D)$ is True, then the condition is satisfied (regardless of the truth value of the other combinations).
- If $(D \wedge M)$ is True, then the condition is satisfied (regardless of the truth value of the other combinations).
- If $(B \wedge M)$ is True, then the condition is satisfied (regardless of the truth value of the other combinations).

Justification Truth Table:

B	D	M	(B^D)	(D^M)	(B^M)	2 forms of identification condition satisfied
T	T	T	**T**	**T**	**T**	Yes
T	T	F	**T**	F	F	Yes
T	F	T	F	F	**T**	Yes
T	F	F	F	F	F	No
F	T	T	F	**T**	F	Yes
F	T	F	F	F	F	No
F	F	T	F	F	F	No
F	F	F	F	F	F	No

This means that $(B \wedge D)$ or $(D \wedge M)$ or $(B \wedge M)$ should be True for the condition to be satisfied.

Therefore, the formula is $(B \wedge D) \vee (D \wedge M) \vee (B \wedge M)$

c) Applicant must present either a birth certificate or both a driver's license and a marriage license.

Step 1: Determine how to satisfy the condition in English.

In order to satisfy the condition, the applicant can:

- Present a birth certificate.
- Present both a driver's license and a marriage license.

Step 2: Translate satisfying cases to logic language.

English	Logic
Present a birth certificate:	B is True
Present both a driver's license and a marriage license	$D \wedge M$ is True

Step 3: Formulate a coherent expression in logic language

In order to satisfy the condition:

- B is True (regardless of the values of D and M).
- $D \wedge M$ is True (regardless of the value of B).

Justification truth table:

B	D	M	(D^M)	condition satisfied
T	T	T	**T**	Yes
T	T	F	F	Yes
T	F	T	F	Yes
T	F	F	F	Yes
F	T	T	**T**	Yes
F	T	F	F	No
F	F	T	F	No
F	F	F	F	No

This means that B or $(D \wedge M)$ should be True for the condition to be satisfied.

Therefore, the formula is $B \vee (D \wedge M)$

2. **Exercise 1.3.7**

Define the following propositions:

s: a person is a senior
y: a person is at least 17 years of age
p: a person is allowed to park in the school parking lot

b) A person can park in the school parking lot if they are a senior or at least seventeen years of age.

Step 1: Get the general form of the sentence.

This sentence is in the form B if A. Sentences in this form amount to A→B where A is what comes after the if statement and B is what comes before. In this case:

A person can park in the school parking lot if they are a senior or at least seventeen years of age.
B **A**

So we get A → B.

Step 2: Break down what constitutes B.

In our case:
- B: A person can park in the school parking lot
- A: They are a senior or at least seventeen years of age.

We notice that B is equivalent to proposition p in this exercise.

So our expression is A → p.

Step 3: Break down what constitutes A.

We notice that A is made up of the two propositions s and y separated by an or.

A: They are a senior or at least seventeen years of age.
s **y**

Therefore, A can be expressed as $(s \vee y)$.

Step 4: Combine the entire expression.

We've seen that the sentence has the form A → B where B is p and A is $(s \vee y)$.

Therefore, the entire formula is $(s \vee y) \rightarrow p$

c) Being 17 years of age is a necessary condition for being able to park in the school parking lot.

Step 1: Determine the general format of the English sentence.

The sentence is of the general format B is necessary for A, where B represents what comes before the delimiter and A is what comes after.

Being 17 years of age is a necessary condition for being able to park in the school parking lot.
B **A**

Sentences in this format express A → B. With, in our case:

- B: Being 17 years of age.
- A: Being able to park in the school parking lot.

Step 2: Break down what constitutes A and B.

- We notice that A is equivalent to p in this exercise
- We notice that B is equivalent to y in this exercise

Step 3: Complete the formula.

We know that:

- The sentence has the form A → B
- A is p
- B is y

Therefore, the expression (p → y) is the logical representation of the English sentence.

d) A person can park in the school parking lot if and only if the person is a senior and at least 17 years of age.

Step 1: Determine the general form of the sentence.

The sentence is of the form A if and only if B. Which represents the logical expression A↔B.

In our case:

A ↔ **B**
A person can park in the school parking lot if and only if the person is a senior and at least 17 years of age.

Step 2: Break down what constitutes A.

A: The person can park in the school parking lot.

This is equivalent to statement p in this exercise.

Step 3: Break down what constitutes B.

We notice that B is made up of the propositions s and y separated by an “and”.

B: The person is a senior and at least 17 years of age.

s $\wedge$ **y**

So B can be expressed as $(s \wedge y)$.

Step 4: Combine the sides of the formula.

We know that:

- The sentence can be expressed as A $\leftrightarrow$ B
- A: p
- B: $(s \wedge y)$

Therefore the entire expression is $p \leftrightarrow (s \wedge y)$.

e) Being able to park in the school parking lot implies that the person is either a senior or at least 17 years old.

Step 1: Determine the general form of the sentence.

The sentence is of the form A implies B. This form represents A $\rightarrow$ B, where A is what comes before the delimiter “implies that” B is what comes after. In our case:

A $\rightarrow$ **B**

Being able to park in the school parking lot implies that the person is either a senior or at least 17 years old

Step 2: Break down what constitutes A.

A: Being able to park in the school parking lot

This is equivalent to proposition p in this exercise.

Step 3: Break down what constitutes B.

We notice that B is made up of s and y separated by an “or”.

B: The person is either a senior or at least 17 years old

s $\vee$ **y**

Therefore B can be expressed as $(s \vee y)$.

Step 4: Combine the sides of the full expression.

We know that:

- The sentence has the form A → B
- A: p
- B: $(s \vee y)$

Therefore, the expression is $p \rightarrow (s \vee y)$.

3. **Exercise 1.3.9**

Use the definitions of the variables below to translate each English statement into an equivalent logical expression.

- y: the applicant is at least eighteen years old
- p: the applicant has parental permission
- c: the applicant can enroll in the course

c) The applicant can enroll in the course only if the applicant has parental permission.

Step 1: Determine the form of the sentence.

The sentence is of the form A only if B. Which represents A→B. In our case:

The applicant can enroll in the course only if the applicant has parental permission.

A → **B**

Step 2: Break down what constitutes A and B.

We notice that:

- A is equivalent to c in this exercise.
- B is equivalent to p in this exercise.

Step 3: Express the logical statement.

We know that:

- The sentence is of the form A → B

- A: c
- B: p

Therefore, the statement is $(c \to p)$.

d) Having parental permission is a necessary condition for enrolling in the course.

Step 1: Determine the form of the sentence.

The sentence is of the form B is necessary for B. Which represents A→B. In our case:

Having parental permission is a necessary condition for enrolling in the course.

B ← **A**

Step 2: Break down what constitutes A and B.

We notice that:

- A is equivalent to c in this exercise.
- B is equivalent to p in this exercise.

Step 3: Express the logical statement.

We know that:

- The sentence is of the form A → B
- A: c
- B: p

Therefore, the statement is $(c \to p)$.

Question 6

1. **Exercise 1.3.6**

Give an English sentence in the form "If...then...." that is equivalent to each sentence.

b) Maintaining a B average is necessary for Joe to be eligible for the honors program.

Step 1: Determine the form of the sentence.

The sentence is of the form B is necessary for A. Which represents A → B.

Maintaining a B average is necessary for Joe to be eligible for the honors program.
B ← **A**

Step 2: Express the same sentence in if A then B form.

If Joe is eligible for the honors program then he maintained a B average.

c) Rajiv can go on the roller coaster only if he is at least four feet tall.

Step 1: Determine the form of the sentence.

The sentence is of the form A only if B. Which represents A → B.

Rajiv can go on the roller coaster only if he is at least four feet tall.
A → **B**

Step 2: Express the sentence in if A, then B form.

If Rajiv can go on the roller coaster, then he is at least four feet tall.

d) Rajiv can go on the roller coaster if he is at least four feet tall.

Step 1: Determine the form of the sentence.

The sentence is of the form B if A. Which represents A → B.

Rajiv can go on the roller coaster if he is at least four feet tall.
B ← **A**

Step 2: Express the sentence in if A, then B form.

If he is at least four feet tall, then Rajiv can go on the roller coaster.

2. **Exercise 1.3.10**

The variable p is true, q is false, and the truth value for variable r is unknown. Indicate whether the truth value of each logical expression is true, false, or unknown.

c) $(p \lor r) \leftrightarrow (q \land r)$

In English:

- We know that p is true, so $(p \lor r)$ must be true regardless of the value of r.
- We know that q is False, so $(q \land r)$ must be false regardless of the value of r.
- Since $(p \lor r)$ is true and $(q \land r)$ is false, then $(p \lor r) \leftrightarrow (q \land r)$ is False

Truth table:

p	q	r	$(p \lor r)$	$(q \land r)$	$(p \lor r) \leftrightarrow (q \land r)$
T	F	unknown	T	F	F

Therefore, $(p \lor r) \leftrightarrow (q \land r)$ is False

d) $(p \land r) \leftrightarrow (q \land r)$

In English:

- p is true, so the truth value of $(p \land r)$ depends on the value of r. If r is true, then $(p \land r)$ is true. If r is false then $(p \land r)$ is false.
- q is false, so $(q \land r)$ is false regardless of the value of r.
- So the truth value of $(p \land r) \leftrightarrow (q \land r)$ is unknown. If r is true then $(p \land r)$ is true, and $(p \land r) \leftrightarrow (q \land r)$ is false. If r is false then $(p \land r)$ is false, and $(p \land r) \leftrightarrow (q \land r)$ is true.

Truth table:

p	q	r	$(p \land r)$	$(q \land r)$	$(p \land r) \leftrightarrow (q \land r)$
T	F	unknown	unknown	F	unknown

Therefore, the truth value of $(p \wedge r) \leftrightarrow (q \wedge r)$ is unknown.

e) $p \rightarrow (r \vee q)$

In English:

- p is true. So for $p \rightarrow (r \vee q)$ to be true, we need $(r \vee q)$ to be true.
- q is false. So the truth value of $(r \vee q)$ depends on r. If r is true then $(r \vee q)$ is true and $p \rightarrow (r \vee q)$ is true. If If r is false then $(r \vee q)$ is false and $p \rightarrow (r \vee q)$ is false.

Truth table:

p	q	r	$(q \vee r)$	$p \rightarrow (q \vee r)$
T	F	unknown	unknown	unknown

Therefore, the truth value of $p \rightarrow (r \vee q)$ is unknown.

f) $(p \wedge q) \rightarrow r$

In English:

- q is false, so $(p \wedge q)$ is false.
- Since the condition $(p \wedge q)$ is false, then $(p \wedge q) \rightarrow r$ must be true regardless of the value of r.

Truth table:

p	q	r	$(p \wedge q)$	$(p \wedge q) \rightarrow r$
T	F	unknown	F	T

Therefore, $(p \wedge q) \rightarrow r$ is true.

Question 7

1. **Exercise 1.4.5**

Define the following propositions:

- j: Sally got the job.
- l: Sally was late for her interview
- r: Sally updated her resume.

Express each pair of sentences using a logical expression. Then prove whether the two expressions are logically equivalent.

b)

Sentence 1: If Sally did not get the job, then she was late for interview or did not update her resume.

Sentence 2: If Sally updated her resume and was not late for her interview, then she got the job.

Step 1: Express sentence 1 as a logical expression

Sentence is in an if A then B form. This form expresses A → B. So:

$\underbrace{\text{If Sally did not get the job,}}_{\textbf{A}}$ then $\underbrace{\text{she was late for her interview or did not update her resume.}}_{\textbf{B}}$

Let's break down A and B. Let's start with A.

We notice that A is the proposition j preceded by "not". In logical terms we express:

A: Sally did not get the job.

A: $\neg$ j

So A is equivalent to $\neg$j.

We notice that B is the propositions l and "not" r separated by an "or". In logical terms:

B: she was late for her interview or did not update her resume.

l $\vee$ $\neg$ **r**

So B is equivalent to $(l \vee \neg r)$.

$\boxed{\text{Therefore, sentence 1 can be expressed as } \neg j \to (l \vee \neg r).}$

Step 2: Express sentence 2 as a logical expression.

Sentence 2 is in if A then B form, which expresses A → B.

If Sally updated her resume and was not late for her interview, then she got the job.

A → **B**

Let's break down what constitutes A and B.

- We notice that B is equivalent to the proposition j.
- We notice that A contains the propositions r and "not" l separated by an "and".

A: Sally updated her resume and was not late for her interview.

r ∧ ¬ l

So B is equivalent to j and A is equivalent to $(r \wedge \neg l)$.

Therefore, sentence 2 can be expressed as $(r \wedge \neg l) \rightarrow j$.

Step 3: Determine if sentence 1 and sentence 2 are logically equivalent.

Truth table:

j	l	r	¬j	¬l	¬r	$(l \vee \neg r)$	$(r \wedge \neg l)$	$\neg j \rightarrow (l \vee \neg r)$	$(r \wedge \neg l) \rightarrow j$
T	T	T	F	F	F	T	F	**T**	**T**
T	T	F	F	F	T	T	F	**T**	**T**
T	F	T	F	T	F	F	T	**T**	**T**
T	F	F	F	T	T	T	F	**T**	**T**
F	T	T	T	F	F	T	F	**T**	**T**
F	T	F	T	F	T	T	F	**T**	**T**
F	F	T	T	T	F	F	T	**F**	**F**
F	F	F	T	T	T	T	F	**T**	**T**

Both statements produce the same truth value for every possible case. Therefore, $\neg j \rightarrow (l \vee \neg r)$ and $(r \wedge \neg l) \rightarrow j$ (and the English sentences they represent) **are logically equivalent**.

c)

Sentence 1: If Sally got the job then she was not late for her interview.

Sentence 2: If Sally did not get the job, then she was late for her interview.

Step 1: Express sentence 1 as a logical statement

The sentence has the form "if A then B", which represents A → B.

If Sally got the job then she was not late for her interview.

A → **B**

We notice that:

- A is equivalent to proposition j.
- B is composed of proposition l preceded by a "not".

B: she was not late for her interview.

¬ l

$\boxed{\text{Therefore, sentence 1 can be expressed as } j \to \neg l.}$

Step 2: Express sentence 2 as a logical statement.

The sentence is of the form if A then B, which represents A → B.

If Sally did not get the job, then she was late for her interview.

A → **B**

We notice that:

- B is equivalent to l.
- A is composed of j preceded by a "not".

A: Sally did not get the job

¬ **j**

Therefore, sentence 2 can be expressed as $\neg j \to l$.

Step 3: Determine if the statements are logically equivalent.

Truth table:

j	l	$\neg j$	$\neg l$	$j \rightarrow \neg l$	$\neg j \rightarrow l$
T	T	F	F	**F**	**T**
T	F	F	T	**T**	**T**
F	T	T	F	**T**	**T**
F	F	T	T	**T**	**F**

The statements do not have the same truth values for all possible cases. Therefore, $j \rightarrow \neg l$ and $\neg j \rightarrow l$ (and by extension the English sentences they represent) **are not logically equivalent**.

d)

Sentence 1: If Sally updated her resume or she was not late for her interview, then she got the job.

Sentence 2: If Sally got the job, then she updated her resume and was not late for her interview.

Step 1: Express sentence 1 as a logical statement.

The sentence is of the form if A then B, which represents A → B.

If Sally updated her resume or she was not late for her interview, then she got the job.

A → **B**

We notice that:

- B is equivalent to j
- A is made of r and "not" l separated by an "or"

A: Sally updated her resume or she was not late for her interview

r ∨ ¬ l

Therefore, sentence 1 can be expressed as $(r \vee \neg l) \rightarrow j$.

Step 2: Express sentence 2 as a logical statement.

The sentence is of the form if A then B, which represents A → B.

If Sally got the job, then she updated her resume and was not late for her interview.

A → **B**

We notice that:

- A is equivalent to j.
- B is composed of r and "not" l separated by an "and".

B: she updated her resume and was not late for her interview.

r $\wedge$ ¬ l

Therefore, sentence 2 can be expressed as $j \to (r \wedge \neg l)$.

Step 3: Determine if the statements are logically equivalent.

Truth table:

j	r	l	$\neg l$	$(r \vee \neg l)$	$(r \wedge \neg l)$	$(r \vee \neg l) \to j$	$j \to (r \wedge \neg l)$
T	T	T	F	T	F	**T**	**F**
T	T	F	T	T	T	**T**	**T**
T	F	T	F	F	F	**T**	**F**
T	F	F	T	T	F	**T**	**F**
F	T	T	F	T	F	**F**	**T**
F	T	F	T	T	T	**F**	**T**
F	F	T	F	F	F	**T**	**T**
F	F	F	T	T	F	**F**	**T**

The statements do not have the same truth values for all possible cases. Therefore, $(r \vee \neg l) \to j$ and $j \to (r \wedge \neg l)$ (and by extension the sentences they represent) **are not logically equivalent**.

Question 8

1. Exercise 1.5.2

c) $(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$

Equation	Law
$(p \rightarrow q) \wedge (p \rightarrow r)$	N/A
$(p \rightarrow q) \wedge (\neg p \vee r)$	Conditional identity
$(\neg p \vee q) \wedge (\neg p \vee r)$	Conditional identity
$\neg p \vee (q \wedge r)$	Distributive Law
$p \rightarrow (q \wedge r)$	Conditional identity

Therefore, $(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$

f) $\neg(p \vee (\neg p \wedge q)) \equiv \neg p \wedge \neg q$

Equation	Law
$\neg(p \vee (\neg p \wedge q))$	N/A
$\neg p \wedge \neg(\neg p \wedge q)$	De Morgan's Law
$\neg p \wedge (\neg\neg p \vee \neg q)$	De Morgan's Law
$\neg p \wedge (p \vee \neg q)$	Double negation Law
$(\neg p \wedge p) \vee (\neg p \wedge \neg q)$	Distributive Law
$(p \wedge \neg p) \vee (\neg p \wedge \neg q)$	Commutative Law
$F \vee (\neg p \wedge \neg q)$	Complement Law
$(\neg p \wedge \neg q) \vee F$	Commutative Law
$\neg p \wedge \neg q$	Identity Law

Therefore, $\neg(p \vee (\neg p \wedge q)) \equiv \neg p \wedge \neg q$

i) $(p \land q) \rightarrow r \equiv (p \land \neg r) \rightarrow \neg q$

Equation	Law
$(p \land q) \rightarrow r$	N/A
$\neg(p \land q) \lor r$	Conditional Identity
$(\neg p \lor \neg q) \lor r$	De Morgan's Law
$\neg p \lor (\neg q \lor r)$	Associative Law
$\neg p \lor (r \lor \neg q)$	Commutative Law
$(\neg p \lor r) \lor \neg q$	Associative Law
$\neg(\neg p \lor r) \rightarrow \neg q$	Conditional identity
$(\neg\neg p \land \neg r) \rightarrow \neg q$	De Morgan's Law
$(p \land \neg r) \rightarrow \neg q$	Double Negation Law

Therefore, $(p \land q) \rightarrow r \equiv (p \land \neg r) \rightarrow \neg q$

2. Exercise 1.5.3

c) $\neg r \lor (\neg r \rightarrow p)$

Equation	Law
$\neg r \lor (\neg r \rightarrow p)$	N/A
$\neg r \lor (\neg\neg r \lor p)$	Conditional identity
$\neg r \lor (r \lor p)$	Double negation
$(\neg r \lor r) \lor p$	Associative Law
$(r \lor \neg r) \lor p$	Commutative Law
$T \lor p$	Complement Law
$p \lor T$	Commutative Law
T	Domination Law

Since $\neg r \lor (\neg r \rightarrow p) \equiv T$, it is a tautology by definition.

d) $\neg(p \rightarrow q) \rightarrow \neg q$

Equation	Law
$\neg(p \rightarrow q) \rightarrow \neg q$	N/A
$\neg\neg(p \rightarrow q) \lor \neg q$	Conditional identity
$(p \rightarrow q) \lor \neg q$	Double negation Law
$(\neg p \lor q) \lor \neg q$	Conditional identity
$\neg p \lor (q \lor \neg q)$	Associative Law
$\neg p \lor T$	Complement Law
T	Domination Law

Since $\neg(p \rightarrow q) \rightarrow \neg q \equiv T$, it is a tautology by definition.

Question 9

1. **Exercise 1.6.3**

c)

Sentence breakdown: There is a number that is equal to its square.

$\exists_x \quad x \quad = \quad x^2$

Therefore, the English statement can be expressed as $\exists_x\,(x = x^2)$ in the domain of real numbers.

d)

Sentence breakdown: Every number is less than or equal to its square.

$\forall_x \quad x \quad \leq \quad x^2$

Therefore, the English statement can be expressed as $\forall_x\,(x \leq x^2)$ in the domain of real numbers.

2. **Exercise 1.7.4**

In the following question, the domain of discourse is a set of employees who work at a company. Ingrid is one of the employees at the company. Define the following predicates:

S(x): x was sick yesterday
W(x): x went to work yesterday
V(x): x was on vacation yesterday

b)

Sentence breakdown: Everyone was well and went to work yesterday.

$\forall_x \quad \neg S(x) \quad \wedge \quad W(x)$

Therefore, the sentence can be expressed as $\forall_x(\neg S(x) \wedge W(x))$.

c)

Sentence breakdown: Everyone who was sick yesterday did not go to work.

$\forall_x$ $\rightarrow$ $S(x)$ $\neg W(x)$

Therefore, the sentence can be expressed as $\forall_x(S(x) \rightarrow \neg W(x))$.

d)

Sentence breakdown: Yesterday someone was sick and went to work.

$\exists_x$ $S(x)$ $\wedge$ $W(x)$

Therefore, the sentence can be expressed as $\exists_x(S(x) \wedge W(x))$.

Question 10

1. **Exercise 1.7.9**

c) ∃x((x = c) → P(x))

	P(x)	Q(x)	R(x)
a	T	T	F
b	T	F	F
c	F	T	F
d	T	T	F
e	T	T	T

If x=a, then (x = c) is False. In that case, since the hypothesis is false then, (x = c) → P(x) must be True.

Therefore, ∃x((x = c) → P(x)) is True.

d) ∃x(Q(x) ∧ R(x))

	P(x)	Q(x)	R(x)
a	T	T	F
b	T	F	F
c	F	T	F
d	T	T	F
e	T	T	T

If x = e, then Q(x) = R(x) = True.

Therefore, ∃x(Q(x) ∧ R(x)) is True.

e) $Q(a) \wedge P(d)$

	P(x)	Q(x)	R(x)
a	T	T	F
b	T	F	F
c	F	T	F
d	T	T	F
e	T	T	T

From the table:

- $Q(a)$ is True
- $P(d)$ is True

Therefore, $Q(a) \wedge P(d)$ is True.

f) $\forall x\ ((x \neq b) \to Q(x))$

	P(x)	Q(x)	R(x)
a	T	T	F
b	T	F	F
c	F	T	F
d	T	T	F
e	T	T	T

If x is b:

- $(x \neq b)$ is False, so $((x \neq b) \to Q(x))$ is True.

If x is not b:

- $(x \neq b)$ is True and $Q(a)$ is True
- $(x \neq b)$ is True and $Q(c)$ is True
- $(x \neq b)$ is True and $Q(d)$ is True
- $(x \neq b)$ is True and $Q(e)$ is True

Therefore, $\forall x\ ((x \neq b) \to Q(x))$ is True.

g) $\forall x\,(P(x) \vee R(x))$

	P(x)	Q(x)	R(x)
a	T	T	F
b	T	F	F
c	F	T	F
d	T	T	F
e	T	T	T

If x= c, then P(x) = R(x) = False.

Therefore, $\forall x\,(P(x) \vee R(x))$ is False.

h) $\forall x\,(R(x) \rightarrow P(x))$

	P(x)	Q(x)	R(x)
a	T	T	F
b	T	F	F
c	F	T	F
d	T	T	F
e	T	T	T

For $x \neq e$:

- R(a) is False, so $\forall x\,(R(x) \rightarrow P(x))$ is True.
- R(b) is False, so $\forall x\,(R(x) \rightarrow P(x))$ is True.
- R(c) is False, so $\forall x\,(R(x) \rightarrow P(x))$ is True.
- R(d) is False, so $\forall x\,(R(x) \rightarrow P(x))$ is True.

For $x = e$:

- R(e) is True and P(e) is True, so $\forall x\,(R(x) \rightarrow P(x))$ is True.

Therefore, $\forall x\,(R(x) \rightarrow P(x))$ is True.

i) ∃x(Q(x) ∨ R(x))

	P(x)	Q(x)	R(x)
a	T	T	F
b	T	F	F
c	F	T	F
d	T	T	F
e	T	T	T

If x = a. Q(x) is True, so (Q(x) ∨ R(x)) is True. Same for c, d, and e.

Therefore, ∃x(Q(x) ∨ R(x)) is True.

2. **Exercise 1.9.2**

P	1	2	3
1	T	F	T
2	T	F	T
3	T	T	F

Q	1	2	3
1	F	F	F
2	T	T	T
3	T	F	F

S	1	2	3
1	F	F	F
2	F	F	F
3	F	F	F

b) ∃x ∀y Q(x, y)

Q	1	2	3
1	F	F	F
2	T	T	T
3	T	F	F

If x= 2:

- Q(x,1) is True
- Q(x,2) is True
- Q(x,3) is True

Therefore, ∃x ∀y Q(x, y) is True.

c) ∃x ∀y P(y, x)

P	1	2	3
1	T	F	T
2	T	F	T
3	T	T	F

If x = 1:

- P(1,x) is True
- P(2,x) is True
- P(3,x) is True

Therefore, ∃x ∀y P(y, x) is True.

d) ∃x ∃y S(x, y)

S	1	2	3
1	F	F	F
2	F	F	F
3	F	F	F

For all combinations of x and y, S(x,y) is False.

Therefore, ∃x ∃y S(x, y) is False.

e) ∀x ∃y Q(x, y)

Q	1	2	3
1	F	F	F
2	T	T	T
3	T	F	F

If x = 1:

- Q(x, 1) is False.
- Q(x, 2) is False.
- Q(x, 3) is False.

Therefore, ∀x ∃y Q(x, y) is False.

f) ∀x ∃y P(x, y)

P	1	2	3
1	T	F	T
2	T	F	T
3	T	T	F

If x = 1, P(1,1) is True.
If x = 2, P(2,1) is True.
If x = 3, P(3,1) is True.

Therefore, ∀x ∃y P(x, y) is True.

g) ∀x ∀y P(x, y)

P	1	2	3
1	T	F	T
2	T	F	T
3	T	T	F

If x = 1 and y = 2, P(x, y) is False.

Therefore, ∀x ∀y P(x, y) is False.

h) ∃x ∃y Q(x, y)

Q	1	2	3
1	F	F	F
2	T	T	T
3	T	F	F

If x = 3 and y = 1, then Q(x, y) is True.

Therefore, ∃x ∃y Q(x, y) is True.

i) ∀x ∀y ¬S(x, y)

S	1	2	3
1	F	F	F
2	F	F	F
3	F	F	F

For all possible combinations of x and y, S(x, y) is False, so ¬S(x, y) is True.

Therefore, ∀x ∀y ¬S(x, y) is True.

Question 11

1. **Exercise 1.10.4**

c)

Sentence breakdown: There are two numbers whose sum is equal to their product.

$\exists_x \exists_y$ $x + y$ $=$ xy

Therefore, the sentence can be expressed $\exists_x \exists_y (x + y = xy)$.

d)

Sentence breakdown: The ratio of every two positive numbers is also positive.

x/y $\forall_x \forall_y$ x > 0, y > 0 >0

Therefore, the sentence can be expressed as $\forall_x \forall_y (((x > 0) \wedge (y > 0)) \rightarrow (x/y > 0))$.

e)

Sentence breakdown: The reciprocal of every positive number less than one is greater than one.

1/x $\forall_x$ x > 0 x < 1 > 1

Therefore, the sentence can be expressed as $\forall_x (((x > 0) \wedge (x < 1)) \rightarrow (1/x > 1))$.

f)

Sentence breakdown: There is no smallest number.

$\exists_x$ ¬ x ≤ all other numbers.

To express all numbers other than x, we need another variable y.

Therefore, the sentence can be expressed as $\neg \exists_x \forall_y (x \leq y)$.

**Note: Technically, the sentence could also be expressed $\neg \exists_x \forall_y ((y \neq x) \rightarrow (x < y))$.

g)

Sentence breakdown: Every number besides 0 has a multiplicative inverse.

$\forall_x$ $x \neq 0$ $\exists_y$ $xy = 1$

Therefore, the sentence can be expressed as $\forall_x \exists_y ((x \neq 0) \rightarrow (xy = 1))$.

2. **Exercise 1.10.7**

The domain of discourse is a group working on a project at a company. One of the members of the group is named Sam. Define the following predicates.

- P(x, y): x knows y's phone number. (A person may or may not know their own phone number.)
- D(x): x missed the deadline.
- N(x): x is a new employee.

c)

Sentence breakdown: There is at least one new employee who missed the deadline.

$\exists_x$ N(x) $\wedge$ D(x)

Therefore, the sentence can be expressed as $\exists_x (N(x) \wedge D(x))$.

d)

Sentence breakdown: Sam knows the phone number of everyone who missed the deadline.

P(Sam, x) $\forall_x$ $\rightarrow$ D(x)

Therefore, the sentence can be expressed as $\forall_x (D(x) \rightarrow P(Sam, x))$.

Note: Sam can be included in $\forall_x$ (so he would know his own number if he missed the deadline).

e)

Sentence breakdown: There is a new employee who knows everyone's phone number.

$\exists_x$ N(x) $\wedge$ $\forall_y$ P(x, y)

Therefore, the sentence can be expressed as $\exists_x \forall_y (N(x) \wedge P(x, y))$.

f)

Sentence breakdown: Exactly one new employee missed the deadline.

$\exists_x \forall_y (x \neq y)$ N(x) D(x), $\neg D(y)$

Therefore, the sentence can be expressed as $\exists_x \forall_y (N(x) \wedge D(x) \wedge (((x \neq y) \wedge N(y)) \rightarrow \neg D(y)))$.

3. **Exercise 1.10.10**

The domain for the first input variable to predicate T is a set of students at a university. The domain for the second input variable to predicate T is the set of Math classes offered at that university. The predicate T(x, y) indicates that student x has taken class y. Sam is a student at the university and Math 101 is one of the courses offered at the university. Give a logical expression for each sentence.

c)
x: students
y: class
Sentence breakdown: Every student has taken at least one class besides Math 101.

$\forall_x$ $T(x,y)$ $\exists_y$ $y \neq math\ 101$

Therefore, the sentence can be expressed as $\forall_x \exists_y ((y \neq math\ 101) \wedge T(x,y))$.

d)
x: student
y: classes
Sentence breakdown: There is a student who has taken every math class besides Math 101.

$\exists_x$ $T(x,y)$ $\forall_y$ $y \neq math\ 101$

Therefore, the sentence can be expressed as $\exists_x \forall_y ((y \neq math\ 101) \rightarrow T(x,y))$.

e)
x: students
y: class 1
z: class 2
Sentence breakdown: Everyone besides Sam has taken at least two different math classes.

$\forall_x$ $x \neq Sam$ T(x,y)^T(x,z) $\exists_y \exists_z (y \neq z)$

Therefore, the sentence can be expressed $\forall_x \exists_y \exists_z ((x \neq Sam) \rightarrow ((y \neq z) \wedge (T(x,y) \wedge T(x,z)))$.

f)

a: class 1

b: class 2

c: all other classes

Sentence breakdown: Sam has taken exactly two math classes.

$$T(Sam,a) \quad \exists_a \exists_b \forall_c \quad a \neq b$$
$$T(Sam,b) \quad b \neq c$$
$$\neg T(Sam,c) \quad a \neq c$$

Therefore, the sentence can be expressed

$\exists_a \exists_b \forall_c ((a \neq b) \wedge T(Sam,a) \wedge T(Sam,b) \wedge (((a \neq c) \wedge (b \neq c)) \rightarrow \neg T(Sam,c)))$

Question 12

1. Exercise 1.8.2

In the following question, the domain of discourse is a set of male patients in a clinical study. Define the following predicates:

P(x): x was given the placebo
D(x): x was given the medication
M(x): x had migraines

Translate each statement into a logical expression. Then negate the expression by adding a negation operation to the beginning of the expression. Apply De Morgan's law until each negation operation applies directly to a predicate and then translate the logical expression back into English.

b) Every patient was given the medication or the placebo or both.

Step 1: Translate sentence to logical statement

Breakdown: Every patient was given the medication or the placebo or both.

$\forall_x$ $D(x)$ $\vee$ $P(x)$ $\vee$ $(M(x) \wedge P(x))$

So, we can express the sentence as $\forall_x(D(x) \vee P(x))$.

Step 2: Negate the logical statement

$\neg\forall_x(D(x) \vee P(x))$

Step 3: Apply De Morgan's Law

Statement	Law
$\neg\forall_x(D(x) \vee P(x))$	N/A
$\exists_x\neg(D(x) \vee P(x))$	De Morgan's Law
$\exists_x(\neg D(x) \wedge \neg P(x))$	De Morgan's Law

So the negated statement can be written as $\exists_x(\neg D(x) \wedge \neg P(x))$

Step 4: Express the sentence in English

There is a patient who did not receive the medication and did not receive the placebo.

c) There is a patient who took the medication and had migraines.

Step 1: Translate sentence to logical statement

Breakdown: There is a patient who took the medication and had migraines.

$\exists_x$ $D(x)$ $\wedge$ $M(x)$

Therefore, the sentence can be expressed as $\exists_x(D(x) \wedge M(x))$.

Step 2: Negate the logical expression

$\neg\exists_x(D(x) \wedge M(x))$.

Step 3: Apply De Morgan's Law

Statement	Law
$\neg\exists_x(D(x) \wedge M(x))$	N/A
$\forall_x\neg(D(x) \wedge M(x))$	De Morgan's Law
$\forall_x(\neg D(x) \vee \neg M(x))$	De Morgan's Law

So, the negated statement can be expressed $\forall_x(\neg D(x) \vee \neg M(x))$.

Step 4: Translate the logical statement to English

All patients either did not take the medication or did not have migraines or did not have both.

d) Every patient who took the placebo had migraines. (Hint: you will need to apply the conditional identity, $p \rightarrow q \equiv \neg p \vee q$.)

Step 1: Express the sentence as a logical statement

Breakdown: Every patient who took the placebo had migraines.

$\forall_x$ $\rightarrow$ $P(x)$ $M(x)$

So the sentence can be expressed as $\forall_x(P(x) \rightarrow M(x))$.

Step 2: Negate the statement

$\neg\forall_x(P(x) \rightarrow M(x))$

Step 3: Apply De Morgan's Law

Equation	Law
$\neg\forall_x(P(x) \rightarrow M(x))$	
$\exists_x\neg(P(x) \rightarrow M(x))$	De Morgan's Law
$\exists_x\neg(\neg P(x) \vee M(x))$	Conditional Identity
$\exists_x(\neg\neg P(x) \wedge \neg M(x))$	De Morgan's Law
$\exists_x(P(x) \wedge \neg M(x))$	Double Negation

So the negated statement can be written as $\exists_x(P(x) \wedge \neg M(x))$

Step 4: Translate the logical statement into English

There is a patient who took the placebo and did not have migraines.

e) There is a patient who had migraines and was given the placebo.

Step 1: Express the sentence as a logical statement

Breakdown: There is a patient who had migraines and was given the placebo.

$\exists_x$ $M(x)$ $\wedge$ P(x)

Therefore, the sentence can be expressed as $\exists_x(M(x) \wedge P(x))$.

Step 2: Negate the statement

$\neg\exists_x(M(x) \wedge P(x))$.

Step 3: Apply De Morgan's Law

Equation	Law
$\neg\exists_x(M(x) \land P(x))$	N/A
$\forall_x\neg(M(x) \land P(x))$	De Morgan's Law
$\forall_x(\neg M(x) \lor \neg P(x))$	De Morgan's Law

So the negated statement can be expressed as $\forall_x(\neg M(x) \lor \neg P(x))$.

Step 4: Translate the logical statement to English.

All patients either did not have migraines or did not receive the placebo (or both).

2. Exercise 1.9.4

Write the negation of each of the following logical expressions so that all negations immediately precede predicates. In some cases, it may be necessary to apply one or more laws of propositional logic.

c) $\exists x\ \forall y\ (P(x, y) \rightarrow Q(x, y))$

Equation	Law
$\neg\exists x\ \forall y\ (P(x, y) \rightarrow Q(x, y))$	N/A
$\forall x\neg\ \forall y\ (P(x, y) \rightarrow Q(x, y))$	De Morgan's Law
$\forall x\exists y\neg(P(x, y) \rightarrow Q(x, y))$	De Morgan's Law
$\forall x\exists y\neg(\neg P(x, y)\ \lor\ Q(x, y))$	Conditional Identity
$\forall x\exists y(\neg\neg P(x, y)\ \land\ \neg Q(x, y))$	De Morgan's Law
$\forall x\exists y(P(x, y)\ \land\ \neg Q(x, y))$	Double Negation Law

Therefore, the negated statement can be expressed as $\forall x\ \exists y\ (P(x, y) \land \neg Q(x, y))$.

d) $\exists x\ \forall y\ (P(x, y) \leftrightarrow P(y, x))$

Equation	Law
$\neg \exists x\ \forall y\ (P(x, y) \Leftrightarrow P(y, x))$	N/A
$\forall x \neg \forall y\ (P(x, y) \Leftrightarrow P(y, x))$	De Morgan's Law
$\forall x \exists y\ \neg(P(x, y) \Leftrightarrow P(y, x))$	De Morgan's Law
$\forall x \exists y\ \neg((P(x, y) \rightarrow P(y, x)) \land (P(y, x) \rightarrow P(x, y)))$	Conditional Identity
$\forall x \exists y\ \neg((\neg P(x, y) \lor P(y, x)) \land (P(y, x) \rightarrow P(x, y)))$	Conditional Identity
$\forall x \exists y\ \neg((\neg P(x, y) \lor P(y, x)) \land (\neg P(y, x) \lor P(x, y)))$	Conditional Identity
$\forall x \exists y (\neg(\neg P(x, y) \lor P(y, x)) \lor \neg(\neg P(y, x) \lor P(x, y)))$	De Morgan's Law
$\forall x \exists y ((\neg\neg P(x, y) \land \neg P(y, x)) \lor \neg(\neg P(y, x) \lor P(x, y)))$	De Morgan's Law
$\forall x \exists y ((P(x, y) \land \neg P(y, x)) \lor \neg(\neg P(y, x) \lor P(x, y)))$	Double Negation
$\forall x \exists y ((P(x, y) \land \neg P(y, x)) \lor (\neg\neg P(y, x) \land \neg P(x, y)))$	De Morgan's Law
$\forall x \exists y ((P(x, y) \land \neg P(y, x)) \lor (P(y, x) \land \neg P(x, y)))$	Double Negation

Therefore, the statement is $\forall x \exists y ((P(x, y) \land \neg P(y, x)) \lor (P(y, x) \land \neg P(x, y)))$.

e) $\exists x\ \exists y\ P(x, y) \wedge \forall x\ \forall y\ Q(x, y)$

Statement	Law
$\neg(\exists x\ \exists y\ P(x, y) \wedge \forall x\ \forall y\ Q(x, y))$	N/A
$\neg\exists x\ \exists y\ P(x, y) \vee \neg\forall x\ \forall y\ Q(x, y)$	De Morgan's Law
$\forall x\ \neg\exists y\ P(x, y) \vee \neg\forall x\ \forall y\ Q(x, y)$	De Morgan's Law
$\forall x\ \forall y\ \neg P(x, y) \vee \neg\forall x\ \forall y\ Q(x, y)$	De Morgan's Law
$\forall x\ \forall y\ \neg P(x, y) \vee \exists x\ \neg\forall y\ Q(x, y)$	De Morgan's Law
$\forall x\ \forall y\ \neg P(x, y) \vee \exists x\ \exists y\ \neg Q(x, y)$	De Morgan's Law

Therefore, the negated statement can be expressed as $\forall x\ \forall y\ \neg P(x, y) \vee \exists x\ \exists y\ \neg Q(x, y)$.