Policy Reminders

- Include your CS username on your page. You will lose a few points from your score if you do not include it.
- You are allowed to work with other students on this homework, as we will not be grading it for correctness. However, each student must turn in their own copy of the homework.
- Show your work for all problems. While we won’t be grading for correctness, you will not receive full credit unless you show your work.
  After all, showing your work is required on the test - and homeworks are intended to help you practice for the test!

Required Problems:

1(e), 1(f), 1(g), 2(d), 2(e), 3(f)

Problem 1 - Encoding in Binary

Convert the following decimal numbers to binary. You may use any method, but make sure to show your work. Give the answer as in 16-bit 2’s complement form.

- (a) 107
- (b) -3097
- (c) 21720
- (d) -1
- (e) Turn in this one: 68

Solution: 68 = 64 + 4
68_{10} = 0000 0000 0100 0100_2

- (f) Turn in this one: -6143

Solution: Since the value is negative, we’ll convert the positive value first.
6143 = 4096 + 2047
6143 = 4096 + 1024 + 1023
6143 = 4096 + 1024 + 512 + 511
6143 = 4096 + 1024 + 512 + 256 + 255
6143 = 4096 + 1024 + 512 + 256 + 128 + 127
6143 = 4096 + 1024 + 512 + 256 + 128 + 64 + 63
6143 = 4096 + 1024 + 512 + 256 + 128 + 64 + 32 + 31
6143 = 4096 + 1024 + 512 + 256 + 128 + 64 + 32 + 16 + 15
6143 = 4096 + 1024 + 512 + 256 + 128 + 64 + 32 + 16 + 8 + 7
6143 = 4096 + 1024 + 512 + 256 + 128 + 64 + 32 + 16 + 8 + 4 + 3
6143 = 4096 + 1024 + 512 + 256 + 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1

6143_{10} = 0001 0111 1111 1111_{2}

2’s complement:

0001 0111 1111 1111 (positive value)
1110 1000 0000 0000 (negated)
1110 1000 0000 0001 (add one)

ANSWER:
1110 1000 0000 0001

- (g) **Turn in this one:** 937

**Solution:** 937 = 512 + 425
937 = 512 + 256 + 169
937 = 512 + 256 + 128 + 41
937 = 512 + 256 + 128 + 32 + 9
937 = 512 + 256 + 128 + 32 + 8 + 1

937_{10} = 0000 0011 1010 1001_{2}
Problem 2 - Some Binary Arithmetic and Conversions

For each of the pairs of numbers below, compute:

- hexadecimal (base 16) equivalents for both \(a\) and \(b\); assume unsigned numbers.
- octal (base 8) equivalents for both \(a\) and \(b\); assume unsigned numbers.
- decimal (base 10) equivalents for both \(a\) and \(b\); assume signed numbers.
- \(a+b\) - Indicate if overflow and/or carry-out occurs; explain your answer. Assume signed numbers. (Do not convert to another base; do your work, and also give your answer, in binary.)
- \(a-b\) by negating \(b\) and adding. Indicate if overflow and/or carry-out occurs; explain your answer. Assume signed numbers. (Do not convert to another base; do your work, and also give your answer, in binary.)

NOTE: Assume that 16-bit binary numbers are being used in this problem. Signed numbers are always encoded using two’s complement.

(a)
\[
\begin{align*}
  a &= 0100 \ 0111 \ 0101 \ 1000 \\
  b &= 1000 \ 0000 \ 1100 \ 0110
\end{align*}
\]

(b)
\[
\begin{align*}
  a &= 0001 \ 0000 \ 0011 \ 1000 \\
  b &= 0111 \ 0010 \ 0100 \ 1011
\end{align*}
\]

(c)
\[
\begin{align*}
  a &= 0000 \ 0000 \ 0110 \ 1100 \\
  b &= 0000 \ 0001 \ 1010 \ 1001
\end{align*}
\]
Solution:

Hexadecimal

\[ a = 0xdb3f \]
\[ b = 0xacc1 \]

Octal

\[ a = 155477 \text{ (octal)} \]
\[ b = 126301 \text{ (octal)} \]

Decimal

For both numbers, the MSB is 1, so the number is negative. We convert each to positive before conversion:

\[ a = 1101\ 1011\ 0011\ 1111 \]
\[ \sim a = 0010\ 0100\ 1100\ 0000 \]
\[ -a = \sim a + 1 = 0010\ 0100\ 1100\ 0001 \]

\[ -a = 2^{13} + 2^{10} + 2^7 + 2^6 + 2^0 \]
\[ -a = 8192 + 1024 + 128 + 64 + 1 \]
\[ -a = 9409 \]
\[ a = -9409 \]

\[ b = 1010\ 1100\ 1100\ 0001 \]
\[ \sim b = 0101\ 0011\ 0011\ 1110 \]
\[ -b = \sim b + 1 = 0101\ 0011\ 0011\ 1111 \]

\[ -b = 2^{14} + 2^{12} + 2^9 + 2^8 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 \]
\[ -b = 16384 + 4096 + 512 + 256 + 32 + 16 + 8 + 4 + 2 + 1 \]
\[ -b = 21311 \]
\[ b = -21311 \]

\[ a + b \]

\[ 1\ 1111\ 1111\ 1111\ 111 \]
\[ a: \ 1101\ 1011\ 0011\ 1111 \]
\[ b: \ +\ 1010\ 1100\ 1100\ 0001 \]

\[ \text{---------------------} \]
\[ 1000\ 1000\ 0000\ 0000 \]

\textbf{No overflow}. We added a negative to a negative, and got a negative result.
We have already calculated the 2’s complement of b above:

\[-b = 0101 \ 0011 \ 0011 \ 1111\]

\[
\begin{array}{cccccccc}
1 & 1 & 1 & 1 & \text{111} & \text{111} \\
\text{a:} & 1101 \ 1011 \ 0011 \ 1111 \\
+ & 0101 \ 0011 \ 0011 \ 1111 \\
\hline
& 0010 \ 1110 \ 0111 \ 1110
\end{array}
\]

No overflow. We started with a negative value, and subtracted a negative, so overflow was impossible.

(e) - Turn in this one

a = 0101 0111 1001 0000  

b = 1000 1100 0111 0000

Solution:

Hexadecimal

a = 0x5790  
b = 0x8c70

Octal

a = 053620 (octal)  
b = 106160 (octal)

Decimal

\[a = 2^{14} + 2^{12} + 2^{10} + 2^9 + 2^8 + 2^7 + 2^4\]
\[a = 16384 + 4096 + 1024 + 512 + 256 + 128 + 16\]
\[a = 22416\]

The high bit of b is 1, so the number is negative. We invert and add one to get the positive value.

\[b = 1000 \ 1100 \ 0111 \ 0000\]
\[\neg b = \neg b + 1 = 0111 \ 0011 \ 1001 \ 0000\]

\[\neg b = 2^{14} + 2^{13} + 2^{12} + 2^{10} + 2^9 + 2^8 + 2^7 + 2^4\]
\[\neg b = 16384 + 8192 + 4096 + 512 + 256 + 128 + 16\]
\[-b = 29534\]
a+b

\[
\begin{array}{l}
1 & 1111 & 111 \\
a: & 0101 & 0111 & 1001 & 0000 \\
b: & +1000 & 1100 & 0111 & 0000 \\
---------------------------------- \\
1110 & 0100 & 0000 & 0000 \\
\end{array}
\]

No overflow. We added a negative to a positive, and so no overflow was possible.

a−b

We have already calculated the 2’s complement of b above:

\[
\begin{array}{l}
111 & 1111 & 1 \\
a: & 0101 & 0111 & 1001 & 0000 \\
+ & 0111 & 0011 & 1001 & 0000 \\
---------------------------------- \\
1100 & 1011 & 0010 & 0000 \\
\end{array}
\]

Overflow! We started with a positive value, subtracted a negative, and got a negative result.

(f) - Turn in this one

a = 1001 1000 1100 0001 
b = 0100 0111 0111 0110

Solution:

Hexadecimal

a = 0x98c1 
b = 0x4776

Octal

a = 114301$_8$ 
b = 043566$_8$

Decimal

The high bit of a is 1, so the number is negative. We invert and add one to get the positive value.

\[
\begin{array}{l}
a = 1001 & 1000 & 1100 & 0001 \\
\sim a = 0110 & 0111 & 0011 & 1110 \\
\sim a + 1 = 0110 & 0111 & 0011 & 1111 \\
\end{array}
\]
\[ a = 2^{14} + 2^{13} + 2^{10} + 2^9 + 2^8 + 2^7 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 \]
\[ a = 16384 + 8192 + 512 + 256 + 32 + 16 + 8 + 7 + 4 + 2 + 1 \]
\[ a = 26431 \]
\[ a = -26431 \]

\[ b = 2^{14} + 2^{10} + 2^9 + 2^8 + 2^7 + 2^5 + 2^4 + 2^3 + 2^1 \]
\[ b = 16384 + 1024 + 512 + 256 + 64 + 32 + 16 + 4 + 2 \]
\[ b = 18294 \]

**a+b**

\[
\begin{array}{c}
\text{11 1111 1} \\
\text{a: 1001 1000 1100 0001} \\
\text{b: + 0100 0111 0111 0110} \\
\hline
\text{---------------------} \\
\text{1110 0000 0011 0111}
\end{array}
\]

**No overflow.** We added a negative to a positive, and so no overflow was possible.

**a-b**

We calculate the 2’s complement of b:

\[
\begin{align*}
\text{b} &= 0100 0111 0111 0110 \\
\bar{b} &= 1011 1000 1000 1001 \\
-\text{b} &= \bar{b}+1 = 1011 1000 1000 1010
\end{align*}
\]

\[
\begin{array}{c}
\text{111 1} \\
\text{a: 1001 1000 1100 0001} \\
\bar{b}: + 1011 1000 1000 1010 \\
\hline
\text{---------------------} \\
\text{0101 0001 0100 1011}
\end{array}
\]

**Overflow!** We started with a positive value, subtracted a positive, and got a positive result.

**EXAMPLES**

(begin on next page)
Example: Problem 1(a)

\[107 = 64 + 43\]
\[107 = 64 + 32 + 11\]
\[107 = 64 + 32 + 8 + 3\]
\[107 = 64 + 32 + 8 + 2 + 1\]
\[107_{10} = 0000\ 0000\ 0110\ 1011_2\]

Example: Problem 1(b)

Since the number is negative, we convert the positive number to binary, and then do 2’s complement.

\[3097 = 2048 + 1049\]
\[3097 = 2048 + 1024 + 25\]
\[3097 = 2048 + 1024 + 16 + 9\]
\[3097 = 2048 + 1024 + 16 + 8 + 1\]

\[3097_{10} = 0000\ 1100\ 0001\ 1001_2\]

2’s complement:

\[0000\ 1100\ 0001\ 1001\] (positive value)
\[1111\ 0011\ 1110\ 0110\] (negated)
\[1111\ 0011\ 1110\ 0111\] (add one)

ANSWER:

\[1111\ 0011\ 1110\ 0111\]

Example: Problem 1(c)

\[21720 = 16384 + 5336\]
\[21720 = 16384 + 4096 + 1240\]
\[21720 = 16384 + 4096 + 1024 + 216\]
\[21720 = 16384 + 4096 + 1024 + 128 + 88\]
\[21720 = 16384 + 4096 + 1024 + 128 + 64 + 24\]
\[21720 = 16384 + 4096 + 1024 + 128 + 64 + 16 + 8\]

\[21720_{10} = 0101\ 0100\ 1101\ 1000_2\]

Example: Problem 1(d)

-1 is one of the “magic” numbers - we don’t need to do any work to find it:

\[-1_{10} = 1111\ 1111\ 1111\ 1111_2\]
Example: Problem 2(a)

\[ a = 0100 \ 0111 \ 0101 \ 1000 \]
\[ b = 1000 \ 0000 \ 1100 \ 0110 \]

Hexadecimal

\[ a = 0x4758_{\text{hex}} \]
\[ b = 0x80c6_{\text{hex}} \]

Octal

\[ a = 043530_{\text{oct}} \]
\[ b = 100306_{\text{oct}} \]

Decimal

For \( a \), the high bit is 0, so the number is positive.
\[ a = 2^{14} + 2^{10} + 2^9 + 2^8 + 2^6 + 2^4 + 2^3 \]
\[ a = 16384 + 1024 + 512 + 256 + 64 + 16 + 8 \]
\[ a = 18264 \]

For \( b \), the high bit is 1, so the number is negative. We convert it to positive before conversion:

\[ b = 1000 \ 0000 \ 1100 \ 0110 \]
\[ 0111 \ 1111 \ 0011 \ 1001 \ (\text{negated}) \]
\[ 0111 \ 1111 \ 0011 \ 1010 \ (\text{plus one}) \]

\[ -b = 2^{14} + 2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^9 + 2^8 + 2^5 + 2^4 + 2^3 + 2^1 \]
\[ -b = 16384 + 8192 + 4096 + 2048 + 1024 + 512 + 256 + 32 + 16 + 8 + 2 \]
\[ -b = 32570 \]
\[ b = -32570 \]

\( a + b; \) check for overflow

\[ \begin{array}{c}
0100 \ 0111 \ 0101 \ 1000 \\
0111 \ 1111 \ 0011 \ 1001 \ (\text{negated}) \\
0111 \ 1111 \ 0011 \ 1010 \ (\text{plus one}) \\
\end{array} \]

\[ \begin{array}{c}
1100 \ 1000 \ 0001 \ 1110 \\
\end{array} \]

No overflow. Overflow is not possible when we add a positive number to a negative.
a-b; check for overflow

\[
\begin{align*}
    b &= 1000 \ 0000 \ 1100 \ 0110 \\
        &\quad 0111 \ 1111 \ 0011 \ 1001 \quad \text{(negated)} \\
        &\quad 0111 \ 1111 \ 0011 \ 1010 \quad \text{(plus one)} \\

    a: &\quad 0100 \ 0111 \ 0101 \ 1000 \\
        &\quad + 0111 \ 1111 \ 0011 \ 1010 \\

    \hline

        &\quad 1100 \ 0110 \ 1001 \ 0010 \\
\end{align*}
\]

Overflow! We started with a positive value, subtracted a negative (and thus expect positive), but the result was negative.
Example: Problem 2(b)

\[a = 0001\ 0000\ 0011\ 1000\]
\[b = 0111\ 0010\ 0100\ 1011\]

**Hexadecimal**

\[a = 0x1038_{\text{hex}}\]
\[b = 0x724b_{\text{hex}}\]

**Octal**

\[a = 010070_{\text{oct}}\]
\[b = 071113_{\text{oct}}\]

**Decimal**

For \(a\), the high bit is 0, so the number is positive.

\[a = 2^{12} + 2^5 + 2^4 + 2^3\]
\[a = 4096 + 32 + 16 + 8\]
\[a = 4152\]

For \(b\), the high bit is 0, so the number is positive.

\[b = 2^{14} + 2^{13} + 2^{12} + 2^9 + 2^6 + 2^5 + 2^4 + 2^3 + 2^0\]
\[b = 16384 + 8192 + 4096 + 512 + 64 + 8 + 2 + 1\]
\[b = 29259\]

**\(a+b\); check for overflow**

\[
\begin{array}{c}
11 \\
\hline
1111 \\
\hline
a: & 0001\ 0000\ 0011\ 1000 \\
b: & +\ 0111\ 0010\ 0100\ 1011 \\
\hline
1000\ 0010\ 1000\ 0011
\end{array}
\]

**Overflow!** We added a positive to a positive, and got a negative.
a-b; check for overflow

\[
\begin{align*}
\text{b} &= 0111\ 0010\ 0100\ 1011 \\
      &= 1000\ 1101\ 1011\ 0100 \quad \text{(negated)} \\
      &= 1000\ 1101\ 1011\ 0101 \quad \text{(plus one)}
\end{align*}
\]

\[
\begin{align*}
\text{11} \\
\text{a:} &= 0001\ 0000\ 0011\ 1000 \\
      &+ 1000\ 1101\ 1011\ 0101 \\
      \quad \text{***********} \\
      &= 1001\ 1101\ 1110\ 1101
\end{align*}
\]

No overflow. A positive, minus a positive, cannot result in overflow.
Example: Problem 2(c)

\[
a = 0000 \ 0000 \ 0110 \ 1100 \\
b = 0000 \ 0001 \ 1010 \ 1001
\]

**Hexadecimal**

\[
a = 0x006c_{\text{hex}} \\
b = 0x01a9_{\text{hex}}
\]

**Octal**

\[
a = 000154_{\text{oct}} \\
b = 000651_{\text{oct}}
\]

**Decimal**

For a, the high bit is 0, so the number is positive.

\[
a = 2^6 + 2^5 + 2^3 + 2^2 \\
a = 64 + 32 + 8 + 4 \\
a = 108
\]

For b, the high bit is 0, so the number is positive.

\[
b = 2^8 + 2^7 + 2^5 + 2^3 + 2^0 \\
b = 256 + 128 + 32 + 8 + 1 \\
b = 425
\]

\[
a+b; \text{ check for overflow}
\]

\[
\begin{array}{c}
11 \ 11 \ 1 \\
a: \quad 0000 \ 0000 \ 0110 \ 1100 \\
b: \quad + \ 0000 \ 0001 \ 1010 \ 1001 \\
\hline
0000 \ 0010 \ 0001 \ 0101
\end{array}
\]

No overflow. We added a positive to a positive, and the result was positive.
a - b; check for overflow

\[ b = 0000\ 0001\ 1010\ 1001 \]
\[ \quad 1111\ 1110\ 0101\ 0110 \quad \text{(negated)} \]
\[ \quad 1111\ 1110\ 0101\ 0111 \quad \text{(plus one)} \]

\[ \quad 1111\ 1 \]

\[ a: \quad 0000\ 0000\ 0110\ 1100 \]
\[ + \quad 1111\ 1110\ 0101\ 0111 \]
\[ \quad \text{---------------------} \]
\[ \quad 1111\ 1110\ 1100\ 0011 \]

No overflow. A positive, minus a positive, cannot result in overflow.