Assignment 2

Problem 1 (1.5 points) (Show all intermediate steps. No intermediate steps will receive 0 point)

Convert integer -3 to 8 digit TC Binary. (0.5 point)

First, convert 3 to binary: $3 = 2^1 + 2^0 =$ `0b 0000 0011`.

Then negate it:

```
  0b 0000 0011    (3 in binary)
  0b 1111 1100    (flip bits)
+ 0b 0000 0001    (add one)
---------------
  0b 1111 1101    (-3)
```

-3 in an 8-bit two's complement binary is `0b 1111 1101`.

Convert decimal -1 to 8 digit TC Hex. (0.5 point)

First, convert 1 to hexadecimal: $1 = 1(16^0) =$ `0x 0000 0001`.

Then, negate it:

```
  0x FFFF FFFF
- 0x 0000 0001    (1 in hex)
---------------
  0x FFFF FFFE    (subtract from 0x FFFF FFFF)
+ 0x 0000 0001    (add one)
---------------
  0x FFFF FFFF    (-1)
```

-1 in an 8-digit two's complement hexadecimal is `0x FFFF FFFF`.

Convert integer -255 to 8 digit Hex (Hint: You can either convert 255 to Hex then negate with TC Hex rule, or you can convert -255 to TC Bin, then convert it to Hex) (0.5 point)

I'm doing the former, so: $255 = 15(16^1) + 15(16^0) =$ `0x 0000 00FF`.

```
  0x FFFF FFFF
- 0x 0000 00FF    (255 in hex)
```

```
---------------
  0x FFFF FF00    (subtract from 0x FFFF FFFF)
+ 0x 0000 0001    (add one)
---------------
  0x FFFF FF01    (-255)
```

-255 in an 8-digit two's complement hexadecimal is `0x FFFF FF01`.

Problem 2 Convert the 2-digit two's complement hexadecimal integer 0x6e to decimal. Show all intermediate steps clearly. (0.5 point)

First, we note that `0x6e` is a positive integer because the leading digit here is less than eight. So, we can just convert it to decimal where `E` correspond to 14.

As such, `0x6e` $= 6(16^1) + 14(16^0) = 110$.

Problem 3 Convert the decimal integer -61 to an 8-bit two's complement binary integer. Show all intermediate steps clearly. (0.5 point)

First, convert 61 into hexadecimal: $61 = 3(16^1) + 13(16^0) =$ `0x 0000 003D`.

Using a reference table, we convert `0x 0000 003D` to binary: `0b 0011 1101`.

Dec	Bin	Hex
3	`0011`	`3`
13	`1101`	`D`

Then, negate it.

```
  0b 0011 1101    (61 in binary)
  0b 1100 0010    (flip bits)
+ 0b 0000 0001    (add one)
---------------
  0b 1100 0011    (-61)
```

-61 in an 8-bit two's complement binary is `0b 1100 0011`.

Problem 4 You're given two 4-digit, 2's complement hexadecimal numbers X = `0xa731` and Y = `0xe6a2`. Compute X-Y. Remember to indicate overflow if it

occurs. Show all intermediate steps clearly. (1 point)

First, negate Y to find $-Y$.

```
  0x FFFF
- 0x E6A2     (Y)
----------
  0x 195D     (subtract from 0x FFFF)
+ 0x 0001     (add one)
----------
  0x 195E     (-Y)
```

Then, $X - Y = X + (-Y)$. So:

```
     1
  0x A731     (X)
+ 0x 195E     (-Y)
----------
  0x C08F
```

In a 4-digit two's complement hexadecimal system, $X - Y =$ `0x C08F`.

Sanity check

	X	Y	$-Y$	$X - Y$
Hex	`A731`	`E6A2`	`195E`	`C08F`
Dec	-22735	-6494	6494	-16241

$$\begin{aligned} X - Y &= -22735 - (-6494) \\ &= -22735 + 6494 \\ &= X + (-Y) \\ &= -22735 + 6494 \\ &= -16241 \end{aligned}$$

Problem 5 (1.5 point)

(1) Please insert the screenshot of your MARS register components. (0.25 point)

Registers | Coproc 1 | Coproc 0

Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x0000000a
$v1	3	0x00000000
$a0	4	0x10010000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400018
hi		0x00000000
lo		0x00000000

(2) What is the data in Register $v0? Write it down in hexadecimal. (In the example above, the data is 0xa). Your answer should start with "0x" which indicates it is a hexadecimal number. (0.25 point)

Data in Register `$v0` is `0x0000000a`.

(3) Can you convert the data in Register $v0 to decimal? Can you convert it to 32-bit binary? (Your answer should start with "0b" which indicates it is a binary number) (0.25 point)

`0x0000000a` in decimal is 10.

Since $10 = 1(2^3) + 0(2^2) + 1(2^1) + 0(2^0)$. In 32-bit binary, 10 is `0b 0000 0000 0000 0000 0000 0000 0000 1010`.

(4) Can you convert the data in Register $gp to 32-bit binary? (0.25 point)

Data in Register `$gp` is `0x10008000`.

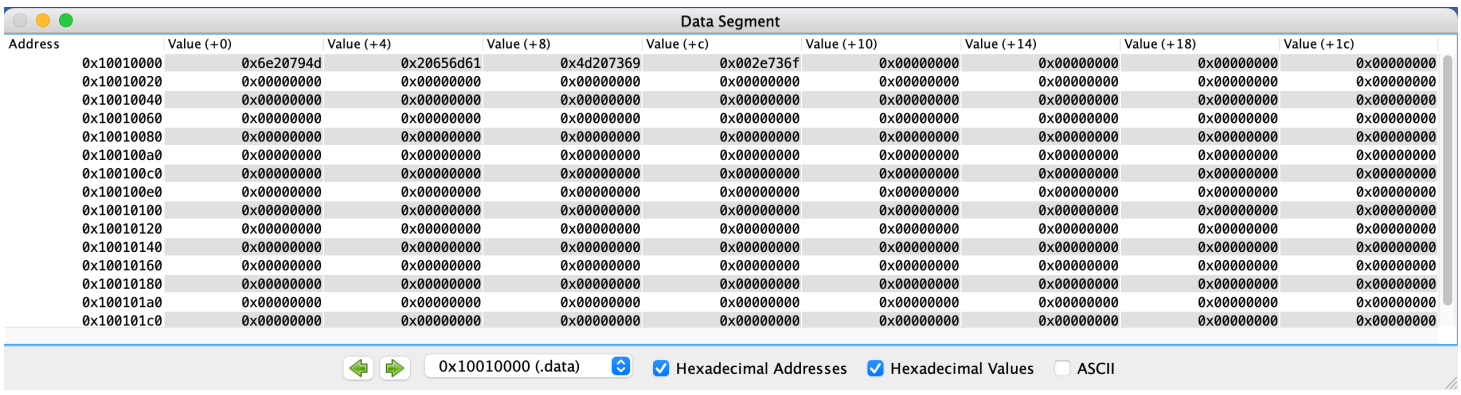
For each hexadecimal digit, there are eight bits. There are only two non-zero hexadecimal digits, namely `1` and `8`.

Using a reference table, we convert `0x1` and `0x8` to binary as `0b0001` and `0b1000`, respectively.

Dec	Bin	Hex
1	`0001`	`1`
8	`1000`	`8`

The rest are zeroes, so we just replace `1` and `8`. So, `0b 1000 8000` is `0b 0001 0000 0000 0000 1000 0000 0000 0000`.

(5) Please insert your MARS data segment screenshot here (0.1 point)



(6) Please write down the 8 digit hexadecimal data saved in the memory address 0x10010018 according to your screenshot (0.2 point)

The data stored at `0x10010018` is `0x00000000`.

(7) Please write down the 8 digit hexadecimal data saved in the memory address 0x10010024 according to your screenshot. (0.2 point)

The data stored at `0x10010024` is `0x00000000`.