TURN IN THESE SHEETS.

NAME____

PROBLEM 1—NUMERIC CONVERSIONS.

Convert between numeric representations of decimal unsigned, decimal signed, 1-byte hexadecimal, 2-byte hexadecimal, and 8-bit binary as requested.

On each row, fill in any conversion marked with a "?".

Note: When dealing with signed representations, you must know whether you are working with 8-bit or 16-bit numbers. Hence, for the conversions involving negative signed numbers, look at the Hex columns to see whether to work with an 8-bit or 16-bit number (one of them will be dashed out; use the other).

The first few are done for you to give you the idea. **Remember: If a line is dashed out, you don't need to complete it.**

Unsigned Decimal	Signed Decimal	1-Byte Hex	2-Byte Hex	8-Bit Binary
0	0	00	0000	00000000
18	----	12	0012	00010010
255	-1	ff	----	11111111
64	64	?	?	?
30	30	?	?	?
250	?	?	----	?
?	-1	--	ffff	--------
?	?	7f	----	?
?	?	80	----	?
?	?	--	1000	--------
?	?	--	7fff	--------
?	?	--	8000	--------

PROBLEM 2—BRANCHING AND THE CONDITION CODE REGISTER.

For each of the following code excerpts, determine whether or not the branch will be taken:

- Determine the value in accumulator A.
- Fill in the contents of the condition code register. Indicate 1, 0, or – (unknown) for each of the four CCR bits N, Z, V, and C (negative, zero, overflow, and carry)
- Indicate whether the branch will be taken.

Example:

```
ldaa #0
inca
bne target
```

`Accum A=` *0x01* `CCR bits NZVC=` *0000* `Branch Y/N?` *Y*

Problem 2a

```
ldaa #1
deca
beq target
```

`Accum A=` `CCR bits NZVC=` `Branch Y/N?`

Problem 2b

```
clra
deca
beq target
```

`Accum A=` `CCR bits NZVC=` `Branch Y/N?`

Problem 2c

```
ldaa #0xff
inca
beq target
```

`Accum A=` `CCR bits NZVC=` `Branch Y/N?`

Problem 2d

```
    ldaa #10
    suba #10
    bmi target
```

```
Accum  A=                    CCR  bits  NZVC=            Branch  Y/N?
```

Problem 2e

```
    ldaa #0x10
    suba #10
    bhs target
```

```
Accum  A=                    CCR  bits  NZVC=            Branch  Y/N?
```

Problem 2f

```
    ldaa #0x80
    asla
    bcs target
```

```
Accum  A=                    CCR  bits  NZVC=            Branch  Y/N?
```

Problem 2g

```
    ldaa #10
    cmpa #10
    beq target
```

```
Accum  A=                    CCR  bits  NZVC=            Branch  Y/N?
```

Problem 2h

```
    ldaa #0xf0
    anda #0x80
    bmi target
```

```
Accum  A=                    CCR  bits  NZVC=            Branch  Y/N?
```

PROBLEM 3—OPERATION OF THE STACK.

Consider the following HC11 program, which loaded into memory at location 0xb600.

Your task is to specify (1) the contents of the stack and (2) the values in the SP (stack pointer) and X registers at three different points of the program execution.

```
 mem      object              line #       instruction                comment
location  code bytes
  B600 8E 01 FF                  7 start:   lds #0x1ff              ; set stack ptr
                                 8
  B603 CE 10 00                  9          ldx #0x1000             ; initialize X
  B606 3C                       10          pshx                    ; put X on stack
                                11
  B607 BD B6 0D                 12          jsr delay               ; call subr
                                13
  B60A 38                       14          pulx                    ; restore X
                                15
  B60B 20 FE                    16 done:    bra done                ; all done
                                17
  B60D CE 01 F4                 18 delay:   ldx #500
  B610 09                       19 delaylp: dex
  B611 26 FD                    20          bne delaylp
  B613 39                       21          rts
```

Use the diagrams below to fill in your answers. The first one is done for you. Please note:

• Write all values in hex. Do not put the “0x” prefix in the boxes.
• In each stack memory address, write two hex digits (for 1 byte).
• In the SP and X register boxes, write four hex digits (for 2 bytes).
• “After line 9” means: after the instruction at line 5 has executed, but before the very next instruction has been run.
• If a value is unknown or indeterminate, mark it with a dash.
• The first problem is done for you.

after line 9:

STACK

m em addr	contents
0x01FB	
0x01FC	
0x01FD	
0x01FE	
0x01FF	
SP reg	
X reg	

after line 10:

STACK

m em addr	contents
0x01FB	
0x01FC	
0x01FD	
0x01FE	
0x01FF	
SP reg	
X reg	

after line 18:

STACK

m em addr	contents
0x01FB	
0x01FC	
0x01FD	
0x01FE	
0x01FF	
SP reg	
X reg	

after line 14:

STACK

m em addr	contents
0x01FB	
0x01FC	
0x01FD	
0x01FE	
0x01FF	
SP reg	
X reg	