(5) Question 1. The format is 8-bit signed. What is the hexadecimal representation of the value -50? First way using basis. -128 needed -50+128 = 78, 64 needed 78-64 = 14, 8 needed 14-8 = 6, 4 needed 6-4=2, 2 needed. Binary is 11001110, which is $CE. Second way, first calculate +50 = 32+16+2, which in binary is 00110010. Next complement 11001101, then add one 11001110 = $CE.

(5) Question 2. Which of the following techniques can be used to handle the problem of overflow?

D) Implement ceiling and floor.

F) Use promotion.

(5) Question 3. Consider the following two instructions

```
ldab #-2
subb #250
```

To determine the overflow (V) bit, first convert both to signed -128 to +127

```
ldab #-2
subb #-6
```

-2 – (-6) is **+4,** so **V=0**

To determine the carry (C) bit, first convert both to unsigned 0 to +255

```
ldab #254
subb #250
```

254-250 is **4,** so **C=0**

(10) Question 4. For the circuit, see Figure 2.17 (b). The desired operating point is 2.5V at 20 mA.

$$R = \frac{5 - V_d - V_{OL}}{I_d} = \frac{5 - 2.5 - 0.5}{0.02} = 100\Omega$$

(10) Question 5. $000A is pushed first, $500B is the return address. Both numbers are big endian

$3FFC =	**$50**	**<= SP**
$3FFD =	**$0B**	
$3FFE =	**$00**	
$3FFF =	**$0A**	

Part b) The subroutine will be executed **10 times** because X is pushed and pulled (eliminating the action caused by **inx**.)

(5) Question 6. Fetch all machine bytes, then store D to memory. The effective address of **2,x** is X+2.

R/W	Addr	Data	Changes to D,X,Y,S,PC,IR,EAR
R	$5000	$6C	IR = $6C, PC = $5001
R	$5001	$02	EAR = $3002, PC = $5002
W	$3002	$22	(RegD and RegX are not changed)
W	$3003	$33	

(20) Question 7. Write assembly code that waits until the switch at PT6 is pressed.

```
Wait ldaa PTT
     anda #$40
     beq  Wait
```

```
Wait brclr PTT,#$40,Wait
```

(20) Question 8. Write assembly code that initializes all numbers to its index value. Implement

```
      ldx  #Buffer  ;pointer
      ldy  #0       ;index
loop  sty  0,x
      inx
      inx
      iny
      cpy  #100
      blo  loop
```

```
      ldx  #Buffer  ;pointer
      ldy  #0       ;index
loop  sty  2,x+
      iny
      cpy  #100
      blo  loop
```

(20) Question 9. If Reg B is greater than 100, turn on the LED at PT5

```
LEDout  cmpb #100
        ble  done
        bset PTT,#$20
done    rts
```