Lab - Study Questions:

1. The font used for source code is courier new, and is of size 8. It is also a 0 line spacing.
2. Pre-compiler directives are just as their names suggest: instructions that are executed before code is compiled, and it directs the compiler. Also according to the AVR starter guide, "directives are not translated directly into opcodes, instead they are used to adjust the location of the program in memory...". The .DEF defines a symbolic name on a register, and the .EQU sets a symbol equal to an expression. From what I can see in the sample code provided, .DEF simply assigned registers a name that can be used later, and .EQU assigns variables.
3.
    a. 00000001 -> 00001000
    b. 00000010 -> 00001000
    c. 00001000 -> 00000100
    d. 00000001 -> 00000001
    e. 00000011 | 01000000 -> 010000011

Challenge Question:

Modified Code:

```
;*****************************************************************
;* Subroutines and Functions
;*****************************************************************
;-----------------------------------------------------------------
; Sub: HitRight
; Desc: Handles functionality of the TekBot when the right whisker
; is triggered.
;-----------------------------------------------------------------
HitRight:
       push mpr ; Save mpr register
       push waitcnt ; Save wait register
       in mpr, SREG ; Save program state
       push mpr ;

       ; Move Backwards for a second
       ldi mpr, MovBck ; Load Move Backward command
       out PORTB, mpr ; Send command to port
       ldi waitcnt, 200 ; Wait for 1 second **this is what was modified
       rcall Wait ; Call wait function



       ; Turn left for a second
       ldi mpr, TurnL ; Load Turn Left Command
       out PORTB, mpr ; Send command to port
       ldi waitcnt, WTime ; Wait for 1 second
       rcall Wait ; Call wait function



       ; Move Forward again
       ldi mpr, MovFwd ; Load Move Forward command
       out PORTB, mpr ; Send command to port


       pop mpr ; Restore program state
       out SREG, mpr ;
```

```
        pop waitcnt ; Restore wait register
        pop mpr ; Restore mpr
        ret ; Return from subroutine


;-----------------------------------------------------------------
; Sub: HitLeft
; Desc: Handles functionality of the TekBot when the left whisker
; is triggered.
;-----------------------------------------------------------------
HitLeft:
        push mpr ; Save mpr register
        push waitcnt ; Save wait register
        in mpr, SREG ; Save program state
        push mpr ;


        ; Move Backwards for a second
        ldi mpr, MovBck ; Load Move Backward command
        out PORTB, mpr ; Send command to port
        ldi waitcnt, 200 ; Wait for 1 second **modifiedto 200 for twice the movebackwards
        rcall Wait ; Call wait function

        ; Turn right for a second
        ldi mpr, TurnR ; Load Turn Left Command
        out PORTB, mpr ; Send command to port
        ldi waitcnt, WTime ; Wait for 1 second
        rcall Wait ; Call wait function

        ; Move Forward again
        ldi mpr, MovFwd ; Load Move Forward command
        out PORTB, mpr ; Send command to port

        pop mpr ; Restore program state
        out SREG, mpr ;
        pop waitcnt ; Restore wait register
        pop mpr ; Restore mpr
        ret ; Return from subroutine
```