```
;
; FinalProject.asm
;
; Created: 4/12/2022 12:14:47 PM
; Author : Tyler Crowley
;

;Setup included in every program
.include "m2560def.inc"

.cseg
.org 0x0000
ldi R16, HIGH(RAMEND)
out SPH, R16
ldi R16, LOW(RAMEND)
out SPL, R16


;Sets PortC on the Arduino to be entirely output
ldi R16, 0b11111111
out DDRC, R16

;Calculates the duty scale, which regulates the speed of the fan.
;Duty Cylce is given in base 10, Duty Scale converts that to a binary number for
  the arduino to use
.equ initDutyCycle = 50
.equ initDutyScale = (256 * initDutyCycle / 100)-1

.equ maxDutyCycle = 100
.equ maxDutyScale = (256 * maxDutyCycle / 100)-1
.equ minDutyCycle = 25
.equ minDutyScale = (256 * minDutyCycle / 100)-1

;Setup PMW & ADC
call initializePMW
call initializeADC

;Tells the ADC to read off and send back a new voltage reading.
;If the reading is at the desired voltage it will loop back, if it is too high it
  will call the increment subroutine,
;and if it is too low it will call the decrement subroutine
main:
rcall startConversion
rcall readADC
sub R25, R26
breq main
brlo reduce
call incDuty
jmp skip
```

```
Reduce: call decDuty
skip: out OCR0B, R23
jmp main

;Checks that the duty scale is not already at its max, then increses it by 1 bit if
   it is not at the maximum
incDuty:
cpi R23, maxDutyScale
brsh maxed
inc R23
jmp skips
maxed:ldi R23, maxDutyScale
skips: ret

;Checks that the duty scale is not already at the minimum, then decreases it by 1
  bit if it not at the minimum
decDuty:
dec R23
cpi R23, minDutyScale
brsh noChange
ldi R23, minDutyScale
noChange: out OCR0B, R23
ret

;Enables digital pin 4 as our output to the fan control, set the PWM to fast and
  non-invert mode, and then sets a prescaler of 1024
initializePMW:
sbi DDRG, 5
ldi R16, 0b00100011
out TCCR0A, R16
ldi R16, 0b00000101
out TCCR0B, R16
ldi R23, initDutyScale
out OCR0B, R23
ret

;Initializes ADC0 as our input, sets a left adjust to ignore the first 2 bits, and
  stores a reference voltage value around 30C to compare off of
initializeADC:
ldi R16, 0b00100000
sts ADMUX, R16
clr R16
ldi R16, (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0)
sts ADCSRA, R16
ldi R26, 0b10001101
ret

;Sets the bit to tell the ADC to read off a new voltage
startConversion:
```

```
lds R16, ADCSRA
ori R16, 0b01000000
sts ADCSRA, R16
ret

;Checks the 4th bit from the left (ADIF) to check that the read is done and data is 
   ready to be downloaded,
;if it is it sends the data to R25 and sends it to the display
readADC:
lds R16, ADCSRA
sbrs R16, ADIF
jmp readADC
clr R24
clr R25
lds R24, ADCL
lds R25, ADCH
out PortC, R25
rcall delay
ret

;A generic delay subroutine, to make the voltage output at a readable speed
delay:
ldi r18, 82
ldi r19, 43
ldi r20, 0

L1:
dec r20
brne L1
dec r19
brne L1
dec r18
brne L1
ret
```