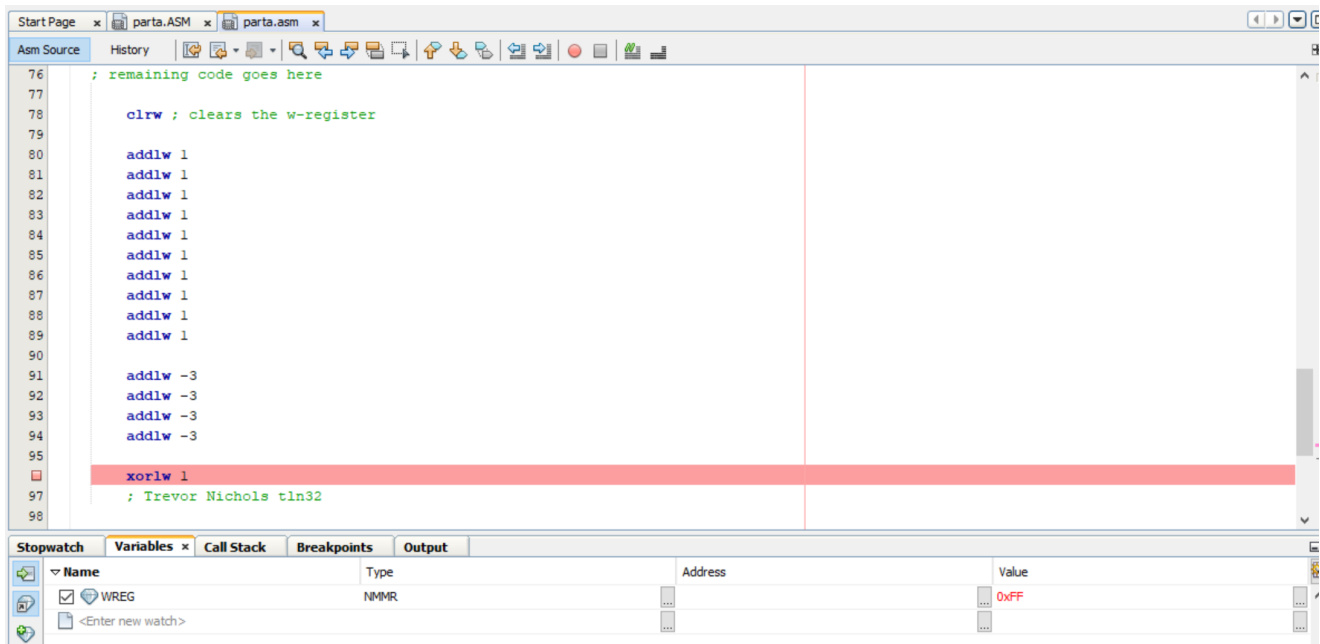


Deliverables

Part a



start:

; remaining code goes here

clr ; clears the w-register

addl 1

addl 1

addl 1

addl 1

addl 1

addl 1

addl 1

addl 1

addl 1

addl 1

addl -3

addl -3

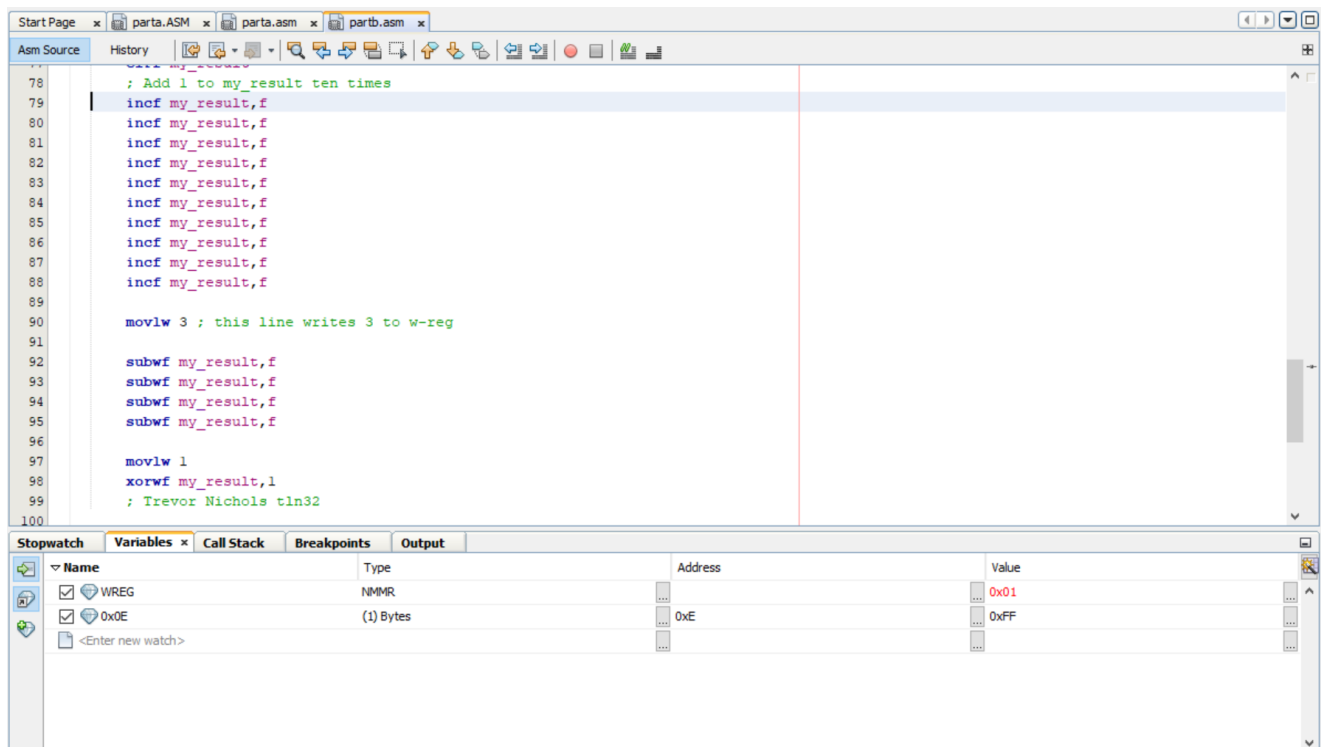
addl -3

addl -3

xorl 1

; Trevor Nichols tln32

Part b



start:

; remaining code goes here

clrf my_result

; Add 1 to my_result ten times

incf my_result,f

incf my_result,f

incf my_result,f

incf my_result,f

incf my_result,f

incf my_result,f

incf my_result,f

incf my_result,f

incf my_result,f

incf my_result,f

movlw 3 ; this line writes 3 to w-reg

subwf my_result,f

subwf my_result,f

subwf my_result,f

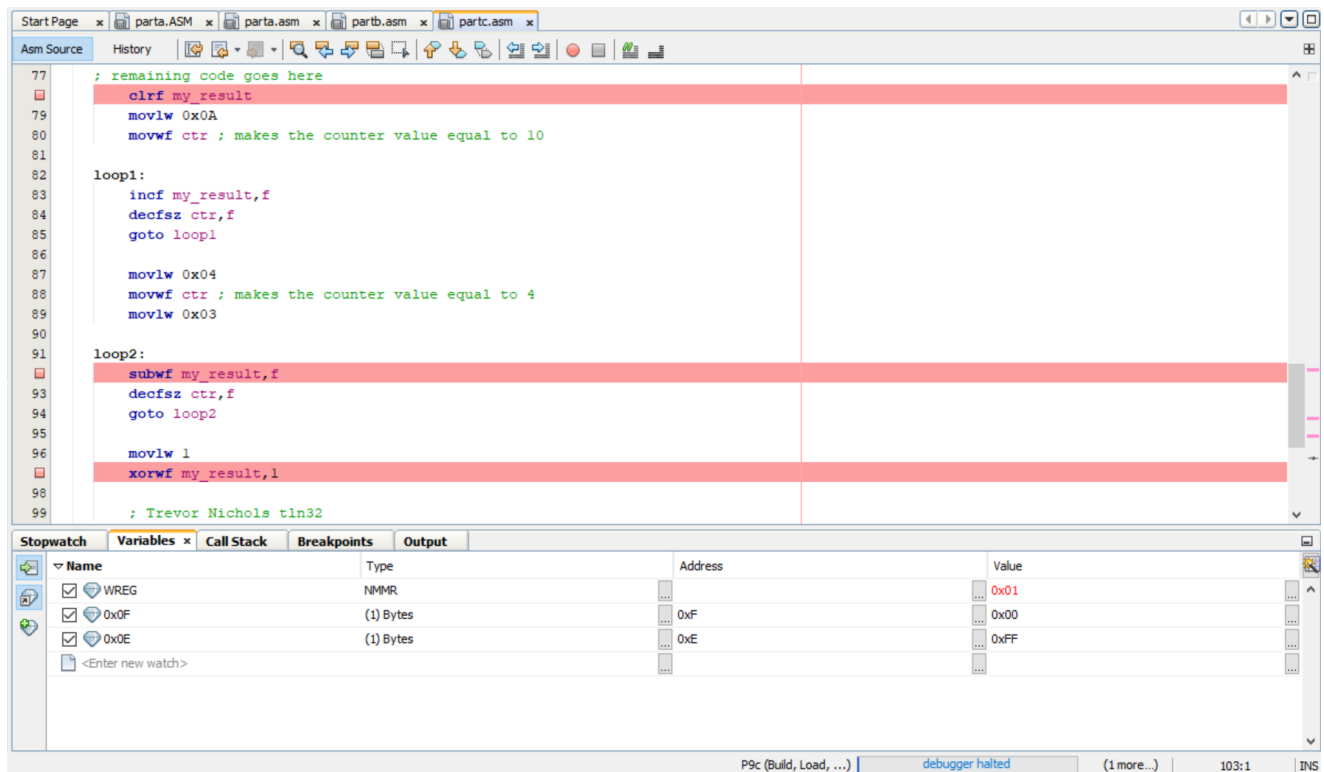
subwf my_result,f

movlw 1

xorwf my_result,1

; Trevor Nichols tln32

Part c



start:

```
; remaining code goes here
  clr f my_result
  movlw 0x0A
  movwf ctr ; makes the counter value equal to 10
```

```
loop1:
  incf my_result,f
  decfsz ctr,f
  goto loop1
```

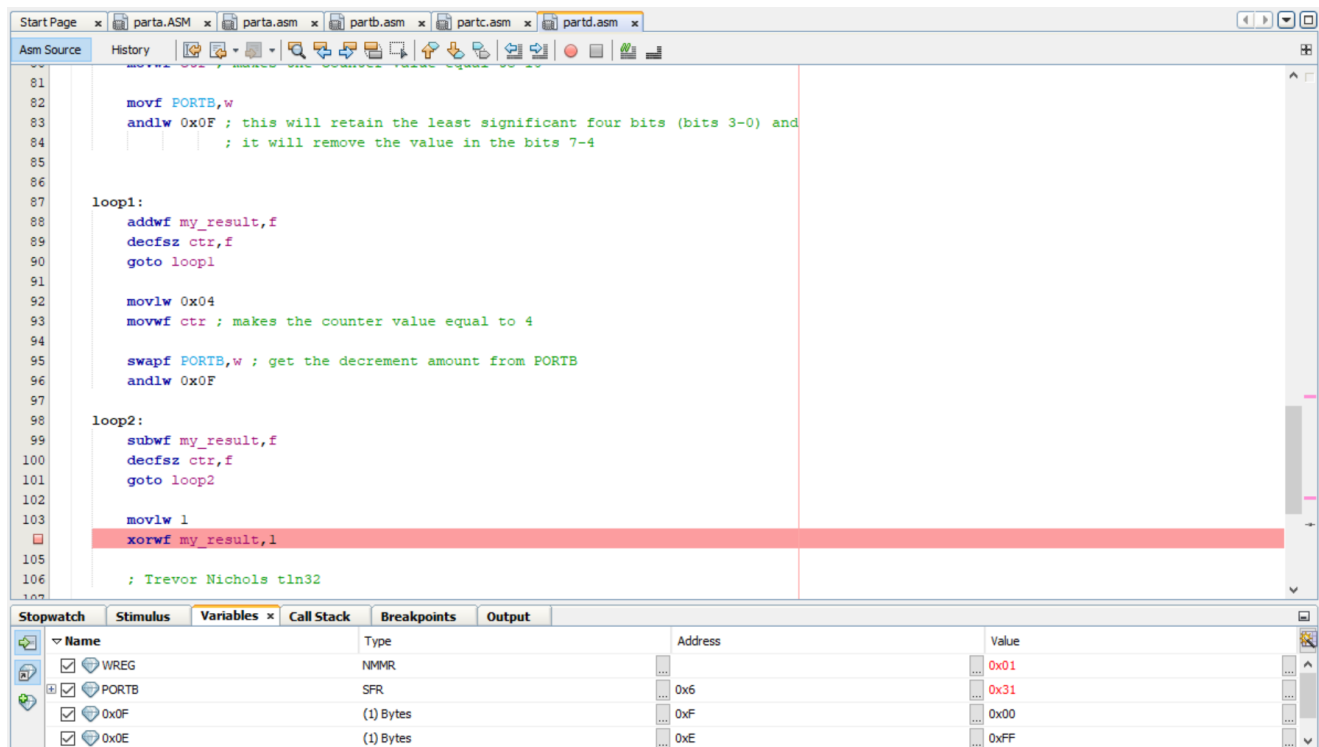
```
  movlw 0x04
  movwf ctr ; makes the counter value equal to 4
  movlw 0x03
```

```
loop2:
  subwf my_result,f
  decfsz ctr,f
  goto loop2
```

```
  movlw 1
  xorwf my_result,1
```

```
; Trevor Nichols tln32
```

Part d



start:

; remaining code goes here

clrf my_result

movlw 0x0A

movwf ctr ; makes the counter value equal to 10

movf PORTB,w

andlw 0x0F ; this will retain the least significant four bits (bits 3-0) and

; it will remove the value in the bits 7-4

loop1:

addwf my_result,f

decfsz ctr,f

goto loop1

movlw 0x04

movwf ctr ; makes the counter value equal to 4

swapf PORTB,w ; get the decrement amount from PORTB

andlw 0x0F

loop2:

subwf my_result,f

decfsz ctr,f

goto loop2

```
movlw 1  
xorwf my_result,1  
  
; Trevor Nichols tln32
```