Program Code

Provide your program code here for each part of the work task (copy-paste your code). You only need to provide the code with the default post fix equation.

Step 1: ----------

```
        ORG     DATA

ARRAY FCB $88,$77,$66,$55,$44,$33,$22,$11

;---------------------------------------------------
; Code Section
; KEEP THIS!!
;---------------------------------------------------
        ORG     PROG

; Insert your code following the label "Entry"
Entry:                  ; KEEP THIS LABEL!!

        ; Enter your code starting here
       LDS #PROG
       LDAA #00
       LDX #0000
LOOP:
       CPX #08
       BEQ EXIT
       LDAA ARRAY,X
       PSHA
       INX
       BRA LOOP
EXIT:
       TSX
       LDAA 3,X
       LDAB 6,X

        ; Branch to end of program
        BRA     FINISH


;---------------------------------------------------
; End program
; KEEP THIS!!
;---------------------------------------------------
FINISH:
        NOP
                END
```

Step 2: ----------

```
;---------------------------------------------------
; Code Section
; KEEP THIS!!
;---------------------------------------------------
        ORG     PROG

; Insert your code following the label "Entry"
Entry:                          ; KEEP THIS LABEL!!

       ; Enter your code starting here
       LDS #PROG
       LDAA #1
       LDAB #5
       PSHD
       LDAA #2
       PSHA
       PULA
       PULB
       ABA
       PSHA
       LDAA #4
       PSHA
       PULA
       PULB
       MUL
       PSHB
       PULA
       PULB
       ABA
       PSHA
       LDAA #3
       PSHA
       PULB
       PULA
       SBA
       PSHA
       ; Branch to end of program
       BRA     FINISH


;---------------------------------------------------
; End program
; KEEP THIS!!
;---------------------------------------------------
FINISH:
        NOP
                        END
```

Step 3: ----------

```
;---------------------------------------------------
; Variable/Data Section
; KEEP THIS!!
;---------------------------------------------------
       ORG     DATA
RPN_IN    FCB     $06,$03,$2F,$04,$2A,$02,$2B
RPN_OUT   RMB     1

RPN_START FDB     RPN_IN
RPN_END   FDB     RPN_OUT-1

OPER      FCB     $2A,$2B,$2D,$2F
LOC1      RMB     1
;---------------------------------------------------
; Code Section
; KEEP THIS!!
;---------------------------------------------------
       ORG     PROG

; Insert your code following the label "Entry"
Entry:                      ; KEEP THIS LABEL!!
      ; Enter your code starting here
       LDX   #0006
       LDS   #PROG
LOAD_STACK:
       CPX   #00
       BLT   LOAD_END
       LDAA  RPN_IN,X
       PSHA
       DEX
       BRA   LOAD_STACK
LOAD_END:
       LDY   #00
POSTFIX_LOOP:
       LDAA  RPN_IN, Y
       CMPA  #$2A
       BEQ   MULTIPLY
       BRA   SKIP1
MULTIPLY:
       PULA
       PULB
       MUL
       PULA
       PSHB
SKIP1:
       CMPA  #$2B
       BEQ   ADDITION
```

```
        BRA   SKIP2
ADDITION:
        PULA
        PULB
        ABA
        PULB
        PSHA
SKIP2:
        CMPA  #$2D
        BEQ   SUBTRACT
        BRA   SKIP3
SUBTRACT:
        PULA
        PULB
        SBA
        PULB
        PSHA
SKIP3:
        CMPA  #$2F
        BEQ   DIVIDE
        BRA   SKIP4
DIVIDE:
        LDAA  #00
        PULB
        STD   LOC1
        LDAA  #00
        PULB
        XGDX
        LDD   LOC1
        IDIV
        PULB
        PSHX
        PULA     ;to remove extra 00
SKIP4:
        INY
        CPY   #07
        BEQ   POSTFIX_END
        BRA   POSTFIX_LOOP
POSTFIX_END:
        ; Branch to end of program
        BRA     FINISH


;---------------------------------------------------
; End program
; KEEP THIS!!
;---------------------------------------------------
FINISH:
```

NOP

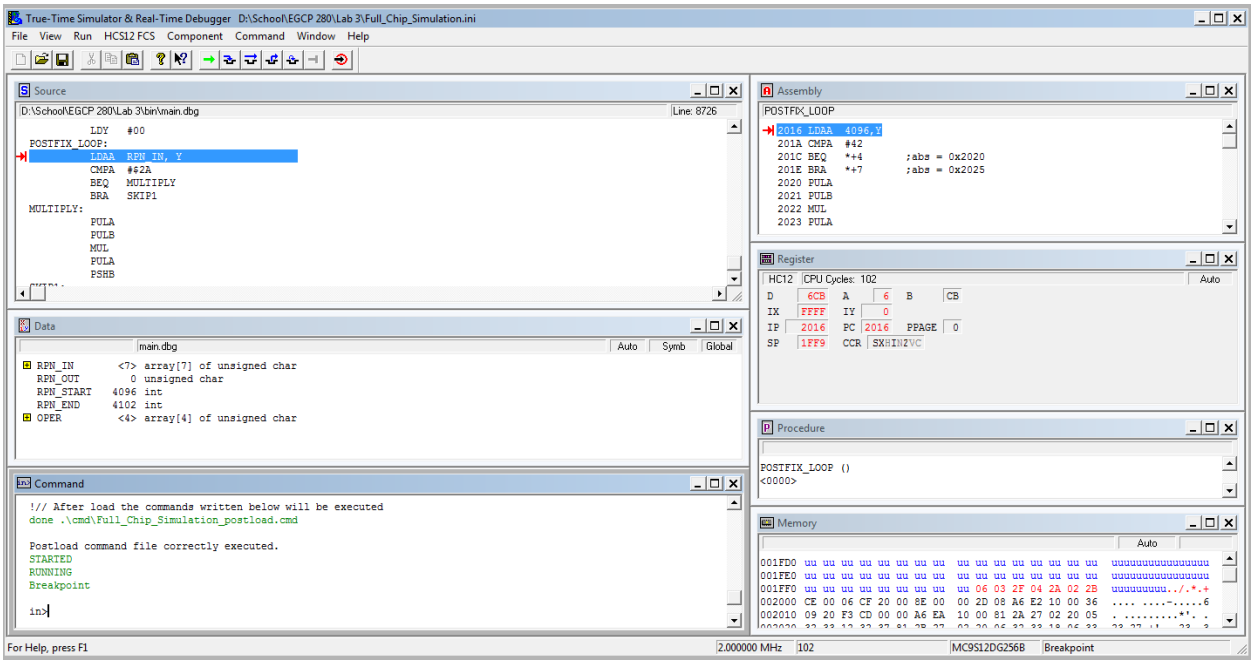
END

Screenshots

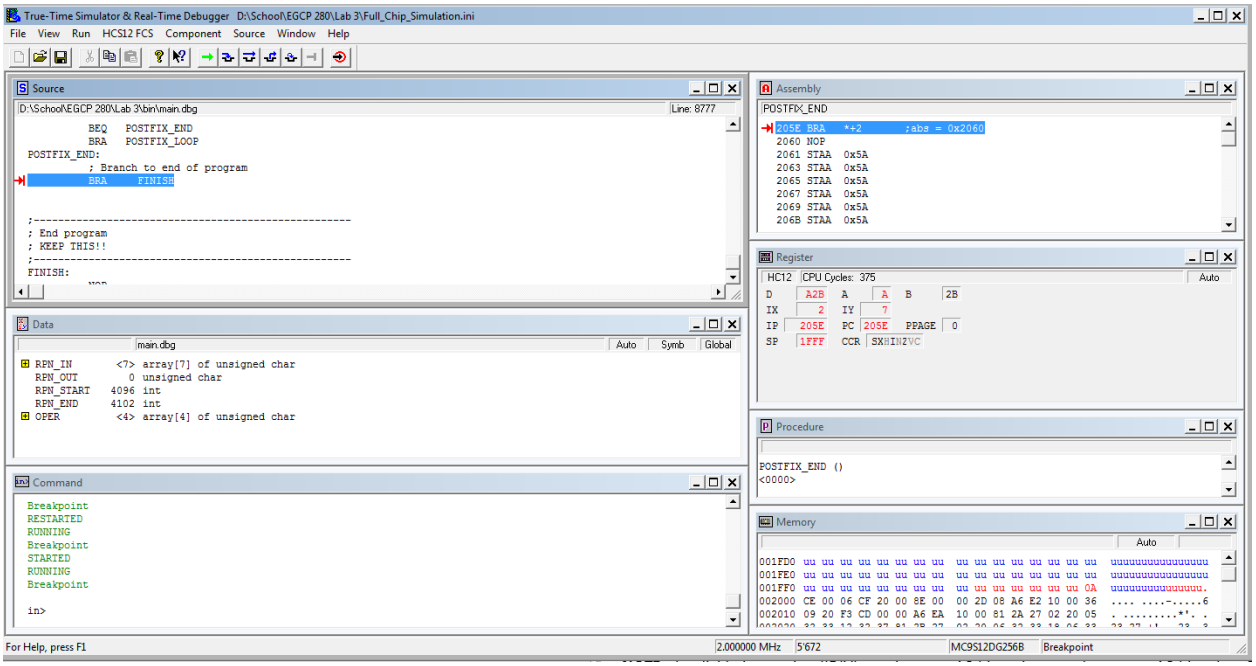
Code the infix notation equations below as postfix notation in your program. Provide your screenshots that show the answers for these equations. I want screenshots of the registers and the memory location of the variables for both CodeWarrior and terminal after execution. Please use a larger image by cropping and resizing the image or use “Alt-PrintScreen” for a Windows computer.

1. ((6 / 3) * 4) + 2

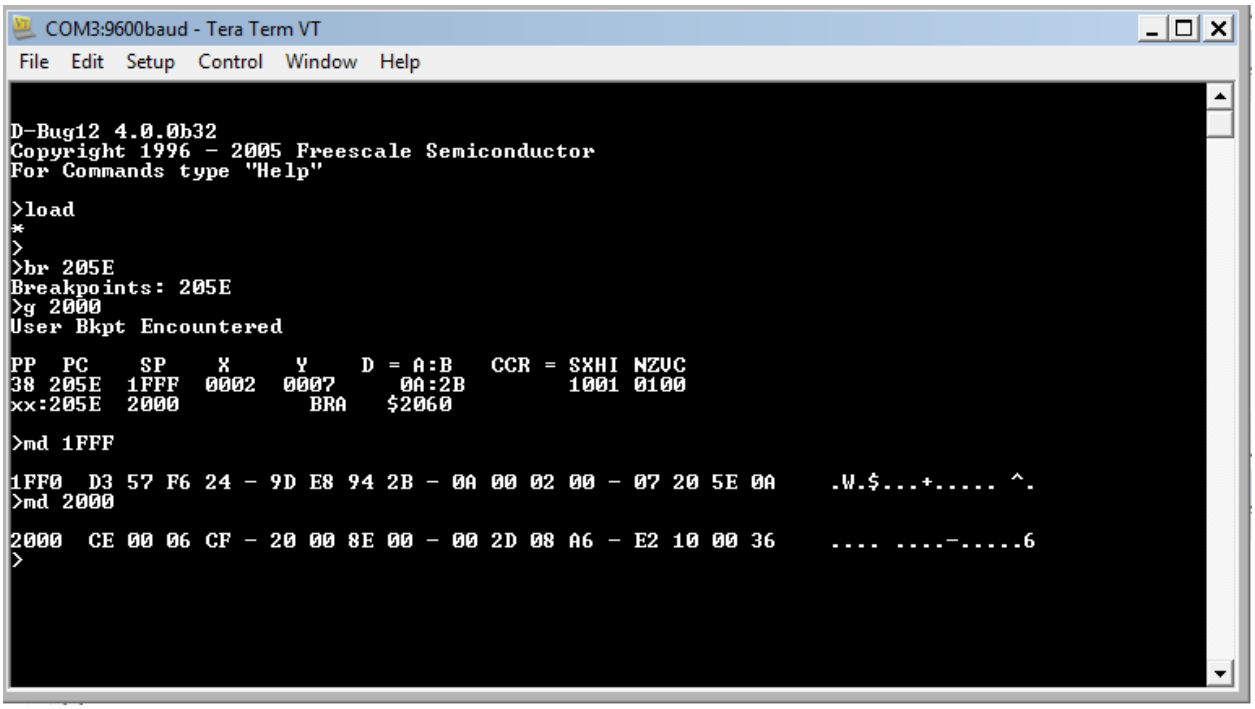
Before:



After:

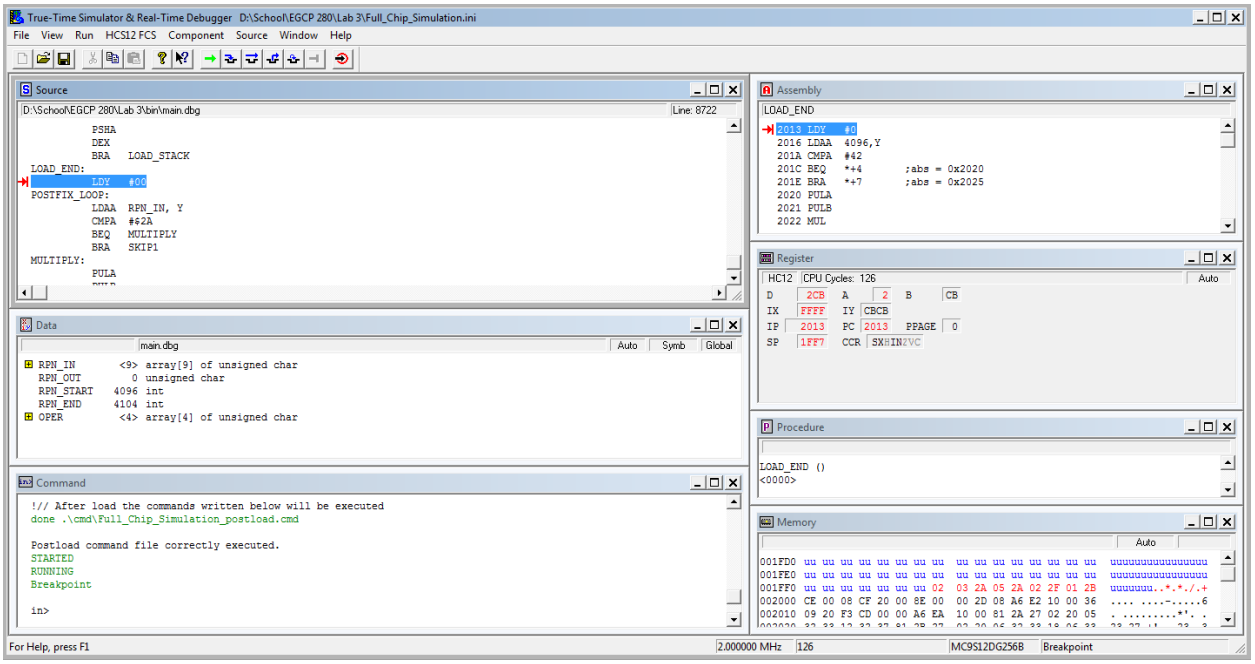


Terminal:

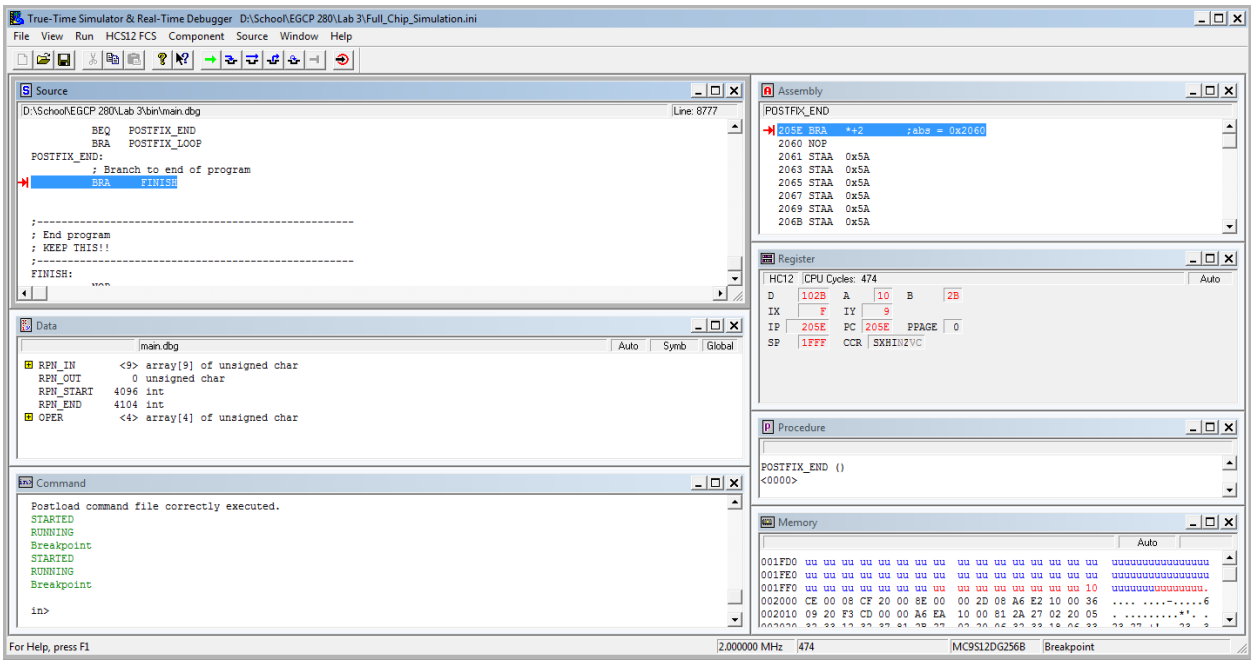


2. (((2 * 3) * 5) / 2) + 1 Postfix = 2 3*5*2/1+

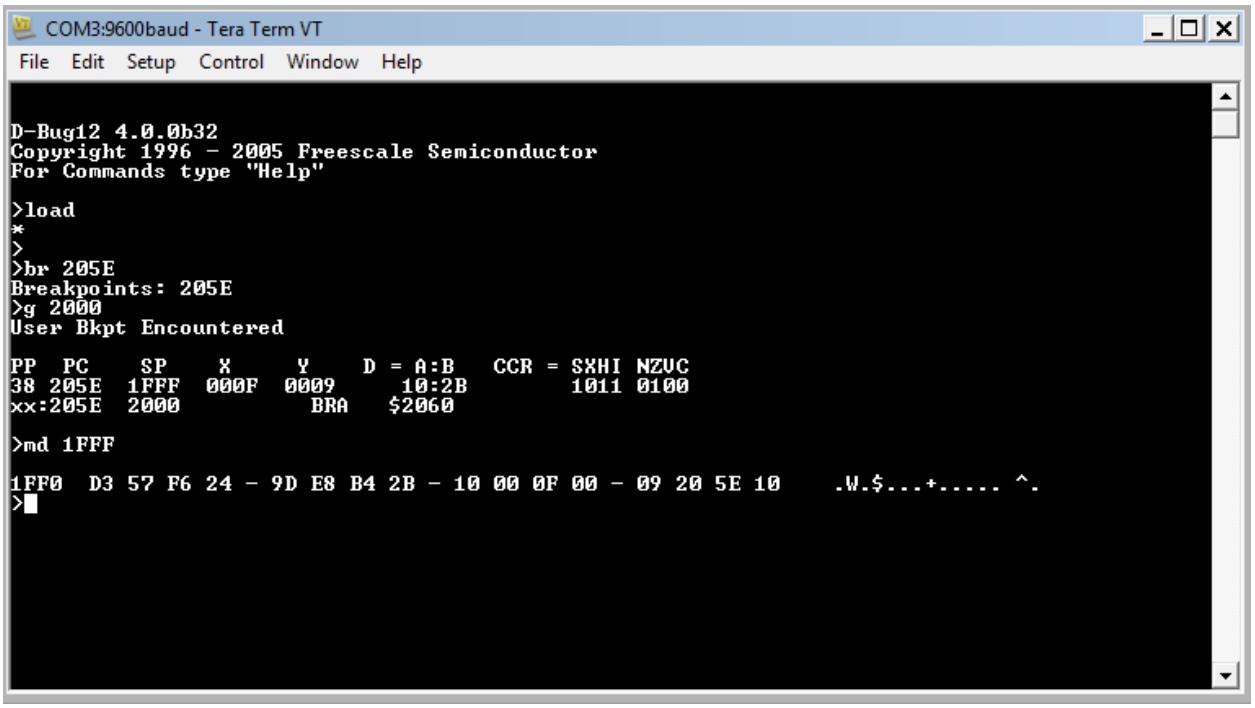
Before:



After:

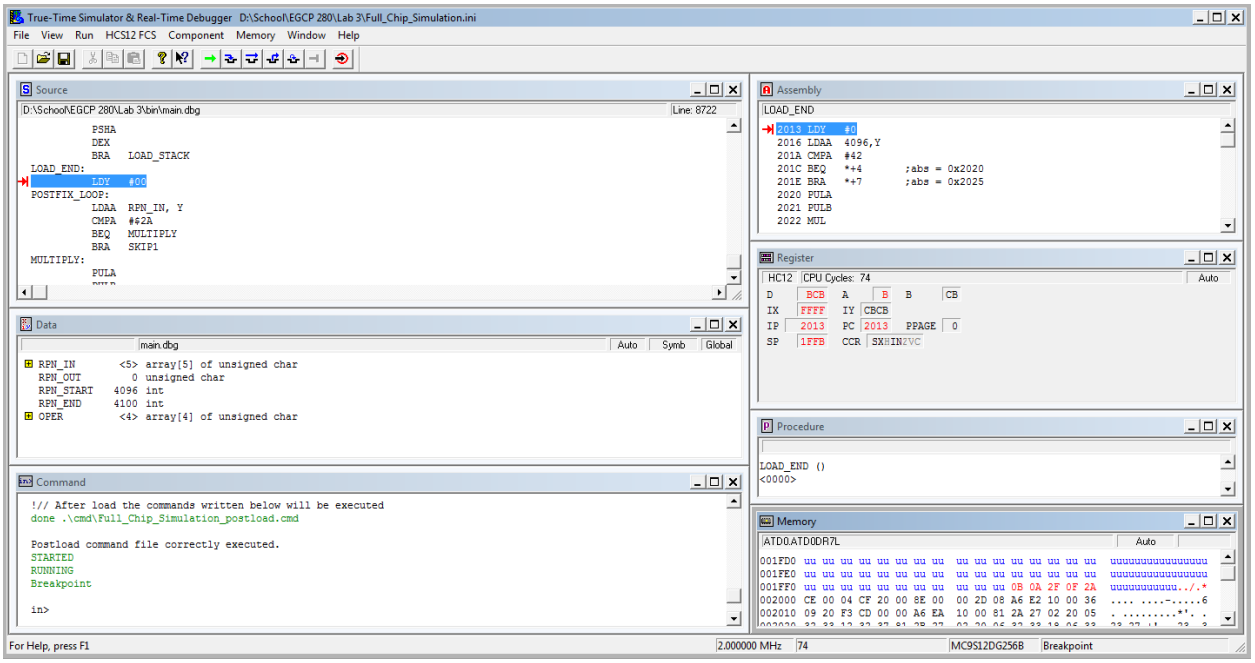


Terminal:

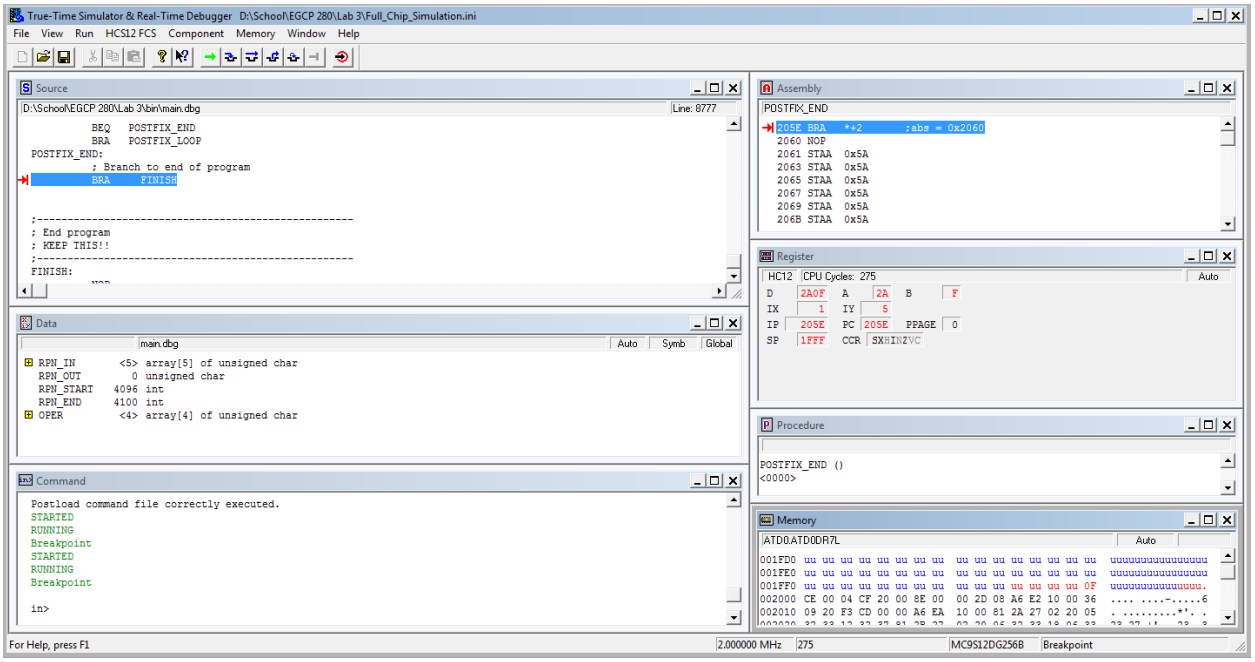


3. ((11 / 10)) * 15 Postfix = 11 10/15*

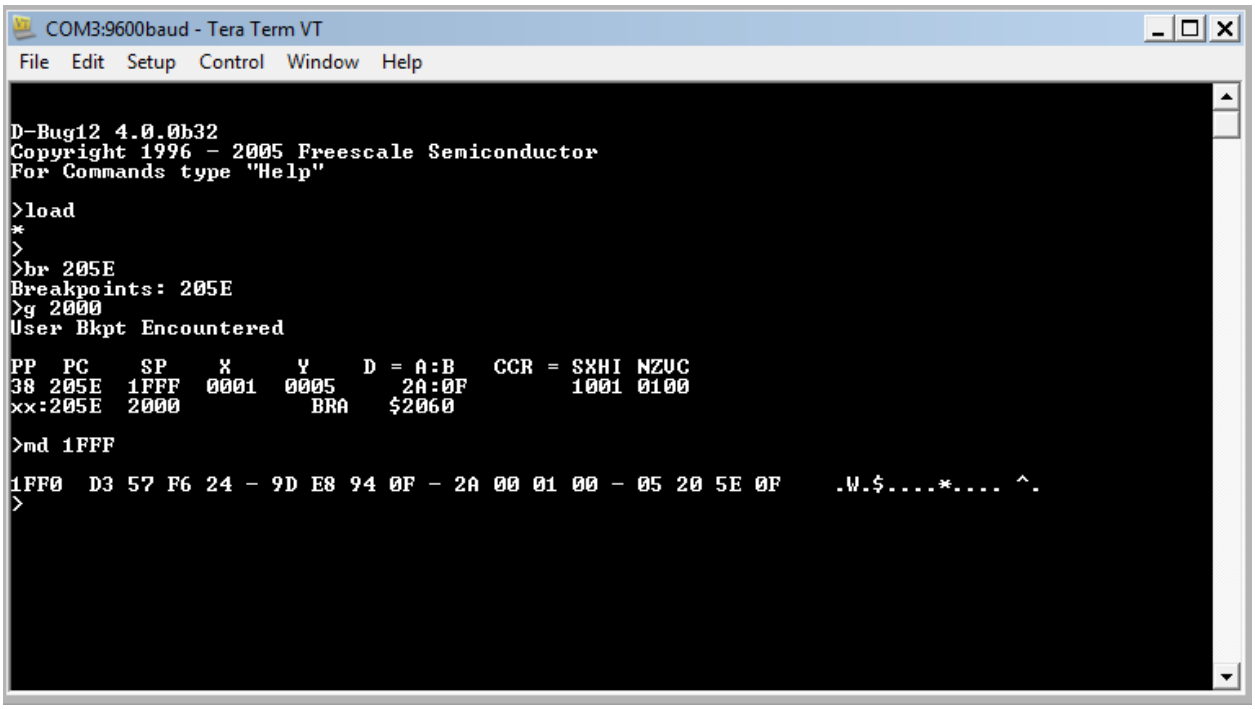
Before:



After:



Terminal:



Questions

1. What are the advantages/disadvantages for allocating the stack starting at PROG?

   Depending on the amount of memory DATA uses, you can possibly overwrite some of your DATA if your stack gets large enough. As the stack size increases, the stack pointer can reach memory locations holding critical information stored by DATA. Overwriting this can cause problems with the running of the program.

The advantages can be that starting the stack at PROG insures that you don't overwrite the code written that the board is supposed to execute. The only condition this would fail is if you pull to many values after the stack is empty.

2. For step 1, does accessing the stack using index mode change the SP? What are some advantages/disadvantages for accessing the stack data this way?

   No, using the index mode to access the stack doesn't change the SP. Some advantages are that you can access any values in the stack without having to remove all the values on top of that stack location. You can also still push or pull values since the SP will always be at the top of the stock. But this brings up the disadvantages, that if you need to modify a value then you must pull all values above that stack location to access that value.