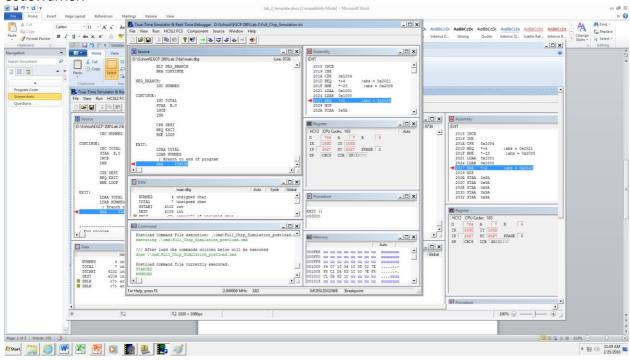
# **Program Code**

```
Provide your program code here for each part of the work task (copy-paste your code)
:-----
; Code Section
; KEEP THIS!!
    ORG PROG
; Insert your code following the label "Entry"
                ; KEEP THIS LABEL!!
Entry:
    ; Enter your code starting here
    LDX #SBLK
    LDY #DBLK
    LDAB #$00
LOOP:
    LDAA X
    CMPA#0
    BLT NEG_BRANCH
    BRA CONTINUE
NEG BRANCH:
    INC NUMNEG
CONTINUE:
    INC TOTAL
    STAA B,Y
    INCB
    INX
    CPX DEST
    BEQ EXIT
    BNE LOOP
EXIT:
    LDAA TOTAL
    LDAB NUMNEG
    ; Branch to end of program
    BRA FINISH
```

## **Screenshots**

Provide your screenshots here for each part of the work task. Include screenshots of the registers and the memory location of the variables for both CodeWarrior and the terminal after execution. Please use a larger image by cropping and resizing the image or use "Alt-PrintScreen" for a Windows computer.

### CodeWarrior:



### Terminal:

```
_ | _ | x |
COM3:9600baud - Tera Term VT
File Edit Setup Control Window Help
                                                                                                            •
D-Bug12 4.0.0b32
Copyright 1996 – 2005 Freescale Semiconductor
For Commands type "Help"
>LOAD
>BR 2027
Breakpoints: 2027
>G 2000
User Bkpt Encountered
            SP
3C00
2000
                    X
100D
                            Y
100D
                                    D = A:B
07:04
$2029
                                                 CCR = SXHI NZUC
1001 0000
xx:2027
                               BRA
>md 100D
1000 04 07 10 06 - 10 0D 00 7E - F8 C1 D4 8D - 1C 00 7E F8
>md 1010
1010 C1 D4 8D 1C - AC 02 6D CF - 85 AC 30 E9 - A5 16 51 C5
                                                                                     ......m...0...Q.
```

# Questions

1. Describe how you copied from the source to the destination block. Also, describe how you knew when you were finished copying. Specifically, I'm looking for addressing modes and conditional branch instructions used and why.

We started by loading the address of sblock to register X, and the address of dblock to register Y. We used a branch not equal loop to keep us inside the loop until the copying was finished. We determined when the copying was done by incrementing register X until it was equal to register Y, which was the start location of the dblock.

2. Describe how you counted the number of negative numbers transferred. Specifically, I'm looking for the conditional branch instructions used and why.

We used a branch less than and compared the value to be copied with 0. If it is less than 0, then the numneg variable would be incremented.