Practice Problems for Topic 3

CIS*2030: Structure and Application of Microcomputers

The practice problems below are important, but will *not* be marked. Their purpose is to ensure that you understand the major concepts covered in Topic 3. Doing these problems by yourself is imperative, as a portion of the marks on the midterm and final exam will be based on questions related to Topic 3. Solutions will be covered during the following (Monday) lecture.

1. What is the difference between the *program counter* and the *location counter*?

2. Write an assembler directive to set the location counter to the hexadecimal address $400.

3. Convert the following C declarations into assembly language. Assume that **ints** are 4 bytes, **short ints** are 2 bytes and **chars** are 1 byte.

```
int time;         /* declare integer “time”           */
short int x,y;    /* declare 2 short integers x and y  */
int z=11;         /* declare integer z with value 11   */
char term='@';    /* declare character with value '@'  */
char buf[7];      /* declare character array of size 7 */
```

4. Draw a memory map to illustrate the following assembly-language code:

```
  ORG $1000
  DS.B 4
A DC.L 12
B DC.B 1,2,3
C DC.W 5
D DC.B 'A'
```

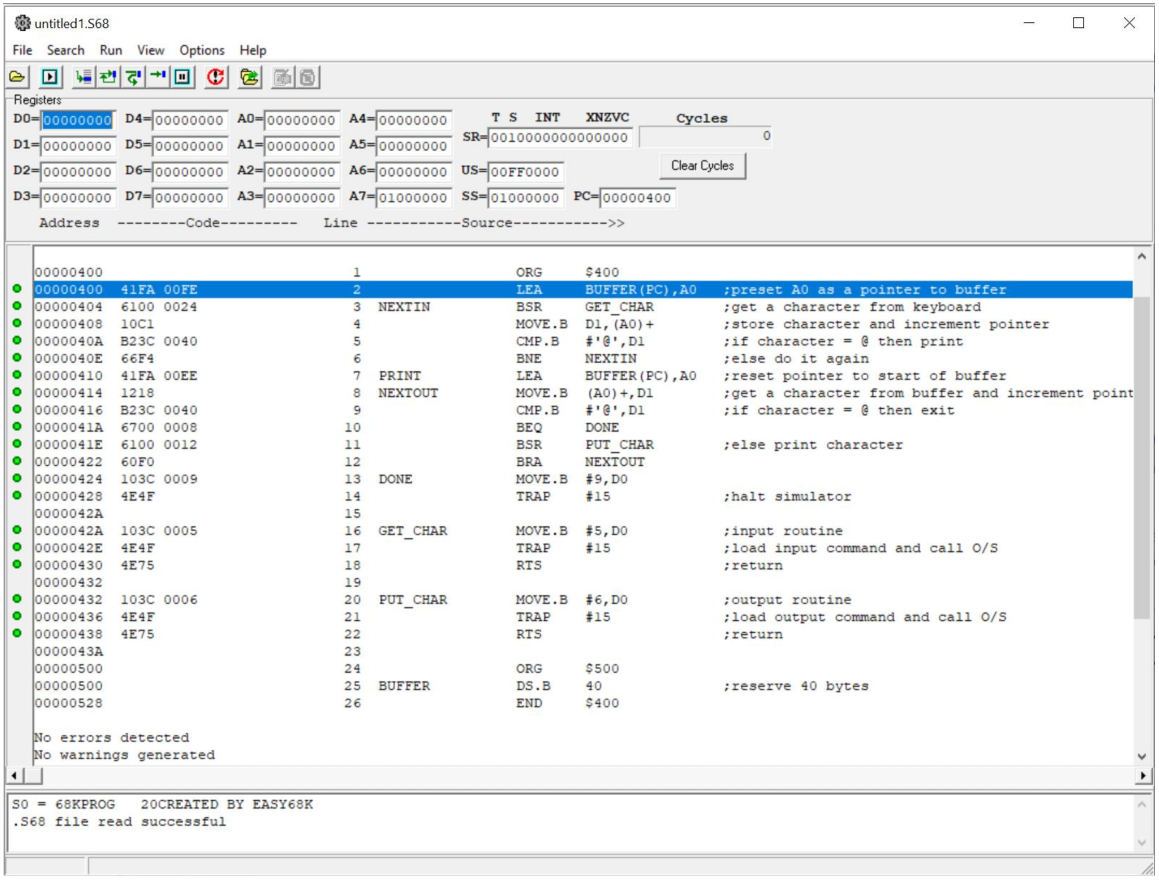
5. The last line of code below seeks to compute the number of words in a buffer. The expression is wrong. Show the correct assemble-time expression in the space below.

```
BUFFER DS.L 30
LENGTH DC.B (*-LENGTH)
```

6. Write a single 68000 instruction to load the number of long words between the location given by label **start** (first location) and the label **last** (first location past the data) into the least-significant word in data register D0. Use an assemble-time expression for the source operand.

7. Consider the listing file below produced by Easy68K.

    a) What is the first and last (32-bit hexadecimal) address of the buffer used in the program?
    b) What is the (32-bit hexadecimal) starting address of the program?
    c) What is the operation word of the TRAP 15 instruction?
    d) What is the (hexadecimal) address of the extension word for the instruction on line 20?



```
D0=00000000  D4=00000000  A0=00000000  A4=00000000      T S  INT  XNZVC     Cycles
D1=00000000  D5=00000000  A1=00000000  A5=00000000  SR=0010000000000000          0
D2=00000000  D6=00000000  A2=00000000  A6=00000000  US=00FF0000
D3=00000000  D7=00000000  A3=00000000  A7=01000000  SS=01000000  PC=00000400
   Address  --------Code---------   Line -----------Source----------->>

00000400                      1            ORG      $400
00000400  41FA 00FE           2            LEA      BUFFER(PC),A0   ;preset A0 as a pointer to buffer
00000404  6100 0024           3  NEXTIN    BSR      GET_CHAR        ;get a character from keyboard
00000408  10C1                4            MOVE.B   D1,(A0)+        ;store character and increment pointer
0000040A  B23C 0040           5            CMP.B    #'@',D1         ;if character = @ then print
0000040E  66F4                6            BNE      NEXTIN          ;else do it again
00000410  41FA 00EE           7  PRINT     LEA      BUFFER(PC),A0   ;reset pointer to start of buffer
00000414  1218                8  NEXTOUT   MOVE.B   (A0)+,D1        ;get a character from buffer and increment point
00000416  B23C 0040           9            CMP.B    #'@',D1         ;if character = @ then exit
0000041A  6700 0008          10            BEQ      DONE
0000041E  6100 0012          11            BSR      PUT_CHAR        ;else print character
00000422  60F0               12            BRA      NEXTOUT
00000424  103C 0009          13  DONE      MOVE.B   #9,D0
00000428  4E4F               14            TRAP     #15             ;halt simulator
0000042A                     15
0000042A  103C 0005          16  GET_CHAR  MOVE.B   #5,D0           ;input routine
0000042E  4E4F               17            TRAP     #15             ;load input command and call O/S
00000430  4E75               18            RTS                      ;return
00000432                     19
00000432  103C 0006          20  PUT_CHAR  MOVE.B   #6,D0           ;output routine
00000436  4E4F               21            TRAP     #15             ;load output command and call O/S
00000438  4E75               22            RTS                      ;return
0000043A                     23
00000500                     24            ORG      $500
00000500                     25  BUFFER    DS.B     40              ;reserve 40 bytes
00000528                     26            END      $400

No errors detected
No warnings generated
```

```
S0 = 68KPROG   20CREATED BY EASY68K
.S68 file read successful
```