How does knowing assembly make you a better programmer?
Confusion when using call by reference
Goal: read the contents of ROM location $4000

```c
// return by reference
// the contents of $4000
void read(char *dpt){
    *dpt = *(char*)0x4000;
}

char *pt;
char data;
void main(void) {
    read(pt);
    read(&data);
    for(;;) {}
}
```

```
org $3800
pt ds.w 1
data ds.b 1
org $4000
read PSHD
LDAB $4000
LDX 0, SP
STAB 0, X
PULD
RTS
main LDD pt
BSR read
LDD #data
BSR read
BRA *
```

Create a new Metrowerks C project
Paste in C code
Make (compile)
Create a new TExaS microcomputer document in Bin folder
Change OpenS19 mode
Add \texttt{pt}, \texttt{data}, D, X, PC, SP to ViewBox \\
Add breakpoint at \texttt{main} \\
Change to Show Instructions \\
Run to breakpoint \\
Single step to see error

10.6. Tables

A \textbf{table} is a collection of identically-sized structures.
\begin{verbatim}
const struct entry{
  unsigned char Name[30];
  unsigned short life[2];
  unsigned short year;
};
typedef const struct entry entryType;
entryType Presidents[3]={
  {"George Washington",{1732,1799},1789},
  {"John Adams",{1735,1826},1797},
  {"Thomas Jefferson",{1743,1826},1801}
};
\end{verbatim}

\textit{Program 10.16. A simple data base with three entries.}

\begin{verbatim}
"George Washington" \\
1732 1799 \\
1789  \\
"John Adams" \\
1735 1826 \\
1797  \\
"Thomas Jefferson" \\
1743 1826 \\
1801
\end{verbatim}

\begin{verbatim}
NAME  equ 0 \\
LIFE  equ NAME+30 \\
YEAR  equ LIFE+4 \\
SIZE  equ YEAR+2 \\
Presidents \\
fcb "George Washington",0 \\
fcb 0,0,0,0,0,0,0,0,0,0,0,0
\end{verbatim}
Program 10.17. The entries of a table written in assembly language.

10.7. Multiple Access Circular Queues

used for data flow problems source to sink
digital filters and digital controllers
fixed length
order preserving
MACQ is always full

source process (producer)
places information into the MACQ
oldest data is discarded when new data is entered

sink process (consumer)
can read any data
MACQ is not changed by the read operation.
Figure 10.7. A multiple access circular queue stores the most recent set of measurements.

**Perform a 60Hz notch filter on a measured signal.**

$v[0]$ $v[1]$ $v[2]$ and $v[3]$ are the most recent data sampled at 360 Hz.

\[
\text{filtered output} = \frac{v[0] + v[3]}{2}
\]

```c
unsigned char v[4];
unsigned char samp(void){
  v[3] = v[2];
  v[2] = v[1];
  v[1] = v[0];
  v[0] = Ad_In(2);
}
```

```assembly
org $3800
v rmb 4
org $4000
samp movb v+2,v+3
movb v+1,v+2
movb v,v+1
ldaa #2
jsr AD_In
staa v
```
| return (v[0]+v[3])/2; | adda v+3 9-bit
|} | rora (v[0]+v[3])/2
| | rts |