Lab 6.4
run solution
allowed to copy/paste existing code
just keep any copyright notices
discuss how to step motor without glitches
start with outputs=$55
how to decide which way to go
let \texttt{goal} be move forward, staying on the track
\textbf{if} off to the \texttt{left} then \textbf{turn right}
\textbf{if} off to the \texttt{right} then \textbf{turn left}

\textbf{Multiply instructions}
\begin{itemize}
\item \texttt{mul} unsigned $A*B$ into $D$
\item \texttt{emul} unsigned $D*Y$ into 32-bit $Y:D$
\item \texttt{emuls} signed $D*RegY$ into $RegY:D$
\end{itemize}

\textbf{Divide instructions}
\begin{itemize}
\item \texttt{idiv} unsigned $D/X$ into $X$, D remainder
\item \texttt{idivs} signed $D/X$ into $X$, D remainder
\item \texttt{fdiv} unsigned $(D:0)/X$ into $X$, D remainder
\item \texttt{ediv} unsigned $(Y:D)/X$ into $Y$, D remainder
\item \texttt{edivs} signed $(Y:D)/X$ into $Y$, D remainder
\end{itemize}

\textbf{Example} Count (0 to 199) = $\frac{5}{9} \times \text{Angle}(0$ to 359)
\begin{verbatim}
* Count  = (5*Angle) / 9
  ldd  Angle
  ldy  #5
  emul
\end{verbatim}

Jonathan W. Valvano
Example Angle (0 to 359) = $\frac{9}{5} \times $Count(0 to 199)

\[
\text{Angle} = \frac{9 \times \text{Count}}{5} \approx \frac{65536 \times \text{Count}}{36409}
\]

Unsigned 8 to 16-bit promotion instructions
; to promote RegB into RegD
  clra
; to promote RegA into RegX
  tfr A,B
  clra
  tfr D,X

Signed 8 to 16-bit promotion instructions
  sex A,D
  sex B,D
  sex A,X
  sex B,X
  sex A,Y
  sex B,Y

16 to 8-bit demotion instructions (signed or unsigned)
  tfr D,A
  tfr D,B
7. Debugging

Chapter 7 objectives are to:

• Describe the basic approach to running programs,
• Discuss the special features of the TExaS debugger,
• Develop methods for observing program execution.

Intrusiveness

degree of perturbation caused by the debugging itself
how much the debugging slows down execution

Nonintrusive

characteristic or quality of a debugger
allows system to operate as if debugger did not exist
e.g., logic analyzer, ICE, BDM

Minimally intrusive

negligible effect on the system being debugged
e.g., dumps (ScanPoint) and monitors

highly intrusive

e.g., print statements, breakpoints and single-stepping