Exercise 4.1:
Show with the help of the Prolog tree how the cut is used in the following program,

\[
\begin{align*}
\text{r(a).} \\
\text{r(b).} \\
\text{q(a) \leftarrow r(X), !, p(a).} \\
\text{q(f(X)) \leftarrow r(X).} \\
\text{p(X) \leftarrow r(X).} \\
\text{p(f(X)) \leftarrow q(X), !, r(X).} \\
\text{p(g(X)) \leftarrow r(X).}
\end{align*}
\]

and where the query \(?- p(X).\) is taken. What would happen without the cut?

Exercise 4.2:
Define the predicate \texttt{max/3} (using cut \texttt{!}), which returns the maximum of two arguments.
For example \texttt{max(2,3,Max).} returns \texttt{Max=3}.

Exercise 4.3:
The built-in predicate \texttt{fail/0}, fails when Prolog encounters it as a goal. Thus, it can be viewed as an instruction for backtracking. On the other hand, the cut predicate \texttt{!}, blocks backtracking.
Define the predicate \texttt{neg/1} which allows you to express \textit{negation as failure}. 
Exercise 4.4:
Take the following program $P$:

\[
\begin{align*}
p & \leftarrow . \\
p & \leftarrow p. \\
q & \leftarrow r. \\
q & \leftarrow \neg r, p. \\
r & \leftarrow \neg p. \\
t & \leftarrow q. \\
t & \leftarrow r, \neg q. \\
\end{align*}
\]

a) Construct the dependency graph $D_P$ of $P$.

b) Give a stratification of $P$.

c) Using your stratification show how to compute the standard model $M_P$ of $P$. 