Consider the following recursive program to compute powers of 2.

\% \textit{pow}(N,P) holds when } P = 2^N.
\texttt{pow}(0,1).
\texttt{pow}(N,P) :- N>0, N1 is N-1, \texttt{pow}(N1,P1), P is 2*P1.

Briefly explain how this program can be made more efficient without using divide-and-conquer.

Write a program to implement your solution. The first two clauses are given for you.

\texttt{powA}(N,P) :- \texttt{powA}(N,1,P).

\textbf{Answer} Use and accumulator and make the program tail-recursive.

\texttt{powA}(0,A,A).
\texttt{powA}(N,A,P) :- N > 0, N1 is N-1, A1 is 2*A, \texttt{powA}(N1,A1,P).