Prolog for Artificial Intelligence

Consider the following problem to be solved using a Prolog program:

Given a closed planar polygon chain represented as a list of \( n \) vertices

\[
[v(x_1,y_1), v(x_2,y_2), \ldots, v(x_n,y_n)]
\]

compute the area of the enclosed polygon, and the orientation of the chain. The area is computed by the line integral \( \frac{1}{2} \int x \, dy - y \, dx \) where the integral is over the polygon chain. A naïve solution is given by the following program, which defines the predicate \textit{area}. The goal \textit{area(Chain,Area)} succeeds when \textit{Chain} is the list of vertices, and the magnitude of \textit{Area} is the area of the polygon bounded by the chain. The sign of \textit{Area} is positive if the orientation of the polygon is anticlockwise and negative if it is clockwise:

\[
\begin{align*}
\text{area}([X],0).
\text{area}([v(X_1,Y_1),v(X_2,Y_2)|VS],Area):-
\quad & \text{area}([v(X_2,Y_2)|VS],Temp), \\
\quad & \text{Area is Temp + (X_1 \times Y_2 - Y_1 \times X_2) / 2.}
\end{align*}
\]

Explain how vertices are processed by this procedure. [4 marks]

Why does this program execute inefficiently? [3 marks]

Write an alternative definition that is tail-recursive and makes use of accumulator variables. [10 marks]

Explain why your alternative definition executes more efficiently. [3 marks]