Q1. Semantic analysis is major part of compiler construction. SDT is used to define grammar for semantic analysis.
Consider the following SDT.

E-> E+T	{E.val = E.val + T.val}
E->T	{E.val = T.val}
T->T*F	{T.val = T.val * F.val}
T->F	{T.val = F.val}
T→ T^G	{ T.val= T.val ^ G.val}
T→G	{T.val = G.val}
G->num	{G.val = num}

Here ^ is meant power.

You are required to generate tokens from input, parse it (using any parser) and perform semantic analysis.

e.g input is 3^2*2

it will generate num ^ num * num tokens

Will parse it and ultimate result will be 18.