A compiler for a tiny programming language (c-like language), which consists mainly of two phases :

**(1) Scanner phase :** Also known as Lexical analysis phase, which recognizes the basic language units (called tokens) where the exact

characters in a token is called its lexeme. Tokens are classified by token types, e.g. identifiers, constant literals, strings, operators,

punctuation marks, and key words. Different types of tokens may have their own semantic attributes (or values) which must be extracted and stored in the symbol table.

**(2) Parser phase :** Also known as Synatx analysis phase, which takes the token produced by lexical analysis as input and generates a

parse tree (or syntax tree). In this phase, token arrangements are checked against the source code grammar, i.e. the parser checks if

the expression made by the tokens is syntactically correct.