Gmail

COMPOSE

Inbox

Starred

Important

Sent Mail

Drafts (5)

Circles

Search, chat, or SMS

anand.joy2008@...

anurag kamal

Ayush Dinker

RAHUL RANJAN

shubham verma

abhishek choudhary

Atul Agarwal

Chandra Prakash

NISCHAL KUMAR

Lab: Assignment-1: Part-B: Building a Lexical Analyzer for C

Compiler Design <bit.compiler.2014@gmail.com>
to bcc: me

Write a Lex program, which:
Given an input C-program, outputs a stream of tokens (other than blank, tab, \n) on the screen.

Output should be of the form:
<lineno>: <token_name> <OPTIONAL token_attribute>

As discussed, there are FIVE categories of tokens:
(1) Keywords (e.g. for, while, if, char, int etc.):
- Each keyword should be given a separate token_name.
- No token_attribute
- There are 32 keywords in C-language.

(2) Operators (e.g. +, ++, +=, etc.)
- Each operator should be given a separate token_name.
- No token_attribute

(3) Punctuation marks (e.g. {, }, (,) etc.)
- Each keyword should be given a separate token_name.
- No token_attribute

(4) Identifiers (e.g. name of functions, variables etc.)
- Common token_name = IDENTIFIER
- token_attribute: actual identifier string

(5) Constants
5.1: Whole numbers (positive only, as negative sign should be classified as an operator)
- Common token_name = NUMBER