compiler design

Gmail

Move to Inbox

More

COMPOSE

Inbox

Starred

Important

Sent Mail

Drafts (6)

Circles

Search, chat, or SMS

kunal acharya

nimesh ghelani

RAHUL RANJAN

Ankit Agarwal

abhishek choudhary

Ashhar Jawaid

Atul Agarwal

Ayush Dinker

Chandra Prakash

Problem: Construct a Symbol Table for all the identifiers appearing in the input C-program.
Input: A C-program as earlier (argv[1])
Output:

- print error message and exit on duplicate declarations
- print error message and exit on usage without declarations
- if no error, print the symbol table after the parse is complete.

The symbol table should contain these FIVE information for each identifier: name, type, line number of declaration, size (1 for non-arrays), initial value if any.

Explanation:
An identifier is used in 2 ways in a C-program:
- Declaration/definition:
    - int a;
    - char x, y, z;
    - float b = 10;
    - double w[10];
    - int c[ ] = {1, 2, 3};

    There are more ways, but we will handle only these.
- Usage in expressions

You need to *IDENTIFY* the grammar productions for both of these, and write appropriate bison actions (in addition to the printfs you already have in Assignment-2, Part-A). For instance,
(1) when you see an identifier as part of rule 23-a (or possibly one of its parents), you need to add that to symbol table (after possibly checking that it doesn't exist).
(2) when you see a type specifier (or, initializer, etc.) you need to update the entry in the symbol table of the corresponding identifier(s).
(3) instead of (1), (2) you can remember the information and add to symbol table when the parsing reduces rule-4.
(4) when you encounter an identifier as part of an expression, e.g. in rule 63-a. you need to check if it is defined or not in the symbol