Indian Institute of Technology Tirupati
CS3310 Compiler Design Laboratory
Lab #8 Due Date: (October 15, 2019 11:30pm)

Objective: To design and implement a mini compiler consisting lexer, parser, and semantic analyzer, to deal with variable declarations in a block structured language.

As learned in class, semantic analysis/translation can be peformed while parsing itself by implementing the rules/actions with the parser. The task for this lab is the following. Consider type declarations in a block structured language like C, in which a program may be written in terms blocks where each block may have declaration section followed by the statements section. Consider variable declarations consisting of premitive data types **int**, **float**, **char**, composite data type, **array** (including multidimensional arrays), and **pointers**. Assume sizes of char, float, and int data types are 1, 4, and 4 bytes respectively. Also consider the size of a pointer variable as 4 bytes.

Design a grammar to generate the all possible type declarations consisting of data types mentioned above, and implement the necessary functionality

- to add the type information for each identifier defined in the program,
- to find the size requirements for each declaration section,
- to maintain the relative memory address (i.e., offset) for each variable declared. Assume 16-bit memory addresses.

Note that in next lab session, you require use these type declarations for typechecking and intermediate code generation.

Input : Blocks of C variable declarations
Output: Display symbol table entries
Execusion: $./minicc prog.c

1. Testcase:

   Input:

```
{
  int a, b, c;
  char e;
  float pi = 3.24;
}
```

   Output:

```
0x0000 a int
0x0004 b int
0x0008 c int
0x0009 e char
0x000A pi float 3.24
```

2. Testcase:

   Input:

```
{
        int a;
        int x, y;
        float c;
        {
                int a;
                int b;
        }
        {
                 char m;
                 int n[10];
        }
}
```

Output:

```
0x0000 a int
0x0004 x int
0x0008 y int
0x000C c float

0x0000 a int
0x0004 b int

0x0000 m char
0x0001 n intarray 40
```

3. Testcase

Input:

```
{
        int a;
        int b;
        int a;
}
```

Output:

```
error: redeclaration of 'a'
```

4. Testcase

Input:

```
{
        int a;
        char a;
        {
                int c;
                int c;
        }
}
```

Output:

```
error: conflicting types for 'a'
error: redeclaration of 'c'
```