Indian Institute of Technology Tirupati
CS3310 Compiler Design Laboratory
Lab #9 Due Date: (October 26, 2019 11:30pm)

Objective: To augment the mini compiler design with the translation to intermediate code and type checking capabilities
In previous lab, you have designed and implemented a mini compiler consisting lexer, parser, and semantic analyzer, to deal with variable declarations in a block structured language. You need to augment the same with translation capabilities. The task is restricted to

- translation of arithmetic expressions declared in a block to the equivalent **three address code**,
- generating error if a variable is not declared, and
- preforming typechecking.

Input : Blocks of C variable declarations and statements containing arithmetic expressions
Output: three address code and errors if any
Execusion: $./minicc prog.c

1. Testcase:

   Input:

```
{
  int x1, x2, y1, y2, dist;
  float m1, m2, m3, total, x, y;
  dist = (x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2);
  total = m1 * m2 * m3;
  x = y + 5;
}
```

   Output:

```
t0=x1-x2
t1=x1-x2
t2=t0*t1
t3=y1-y2
t4=y1-y2
t5=t3*t4
t6=t2+t5
dist=t6
t7=m1*m2
t8=t7*m3
total=t8
t9=y+ (float) 5
x=t9
```

2. Testcase:

   Input:

```
{
int a;
  {
    int b, c;
```

```
    a = b + c;
    x = a + b;
  }
}
```

Output:

```
t0 = b + c
a = t0
error: var 'a' is not declared in the scope
```