/\*36 T1

Shikhar Brajesh 2012A7PS114P

Girish Rathi 2012A7PS347P

\*/

Type Checking Rules :

1. The type of an identifier is the type appearing while declaring the variable or the type declared in function defintion.

2. The type of TK\_Numeric has to be int.

3. The type of any symbol inside ' ' is a char.

4. The type of TRUE or FALSE is boolean.

5. The type of all elements of an array is defined in declaration. All elements are of an array are of same type.

6. The return type of function is declared in function definition. The correct type of value has to be returned.

7. The type of a simple expression (E) of the form expression(E1) <operator> Expression(E2)

a. is integer, if both expressions are of type integer and the operator is arithmetic operator.

b. is integer, if both expressions are of type character and the operator is arithmetic operator.

c. is integer, if one of the expressions is of type character and the other arithmetic with the operator being arithmetic.

d. is boolean, if one of the expressions is of type character and the other arithmetic with the operator being relational.

d. is boolean, if both expressions are of type integer and the operator is relational operator.

e. is boolean, if both expressions are of type integer or character and the operator is relational.

f. is boolean, if both expressions are of type boolean and the operator is logical.

8. Unary operations preserve the type value of expression or identifier. Unary operations performed on character/integer type only.

9. The type of the expression is ERROR, if the above rules do not derive the type of E appropriately.

Semantic Rules

1. An identifier cannot be declared multiple times in the same scope.

2. An identifier must be declared before its use.

3. An identifier declared in one scope can be declared again in any other scope.

4. The types of parameter being returned by a function must be the same as that of the parameter in the function.

5. The parameters being returned by a function must be assigned a value. If a parameter does not get a value assigned within the function definition, it reports as an error.

6. The function that does not return any value needs to return NULL.

7. Function input parameters passed while invoking it should be of the same type as those used in the function definition. Number of input parameters must be same as that of those used in the function definition.

8. An if statement must have the expression of boolean type.

9. Function overloading is not allowed.

10. Function name must be a valid identifier and must not have been used before as variable name.

11. The right hand side expression of an assignment statement must be of the same type as that of the left hand side identifier.

12.. Three special words- NULL, TRUE , FALSE