**COMPILER DESIGN**

***EX9: INTERMEDIATE CODE GENERATION***

**AIM:** To write a program that produces the intermediate code in

the form of a three-address code and postfix expression for the

given infix expression.

**LANGUAGE USED:** Python

**PROCEDURE:**

* Take the type of expression and the infix notation as inputs.
* Store the input string in a list after removing all white spaces.
* For arithmetic and the RHS of assignment expressions, obtain

their postfix notations.

* Using a stack, obtain the memory registers and their values

used in the three-address code.

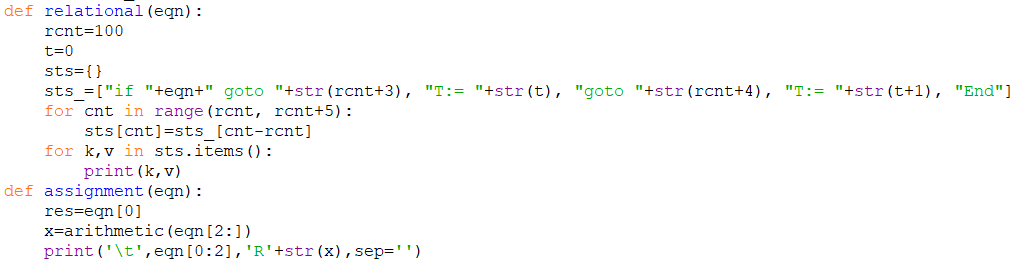
* For assignment expressions, return the value of the last

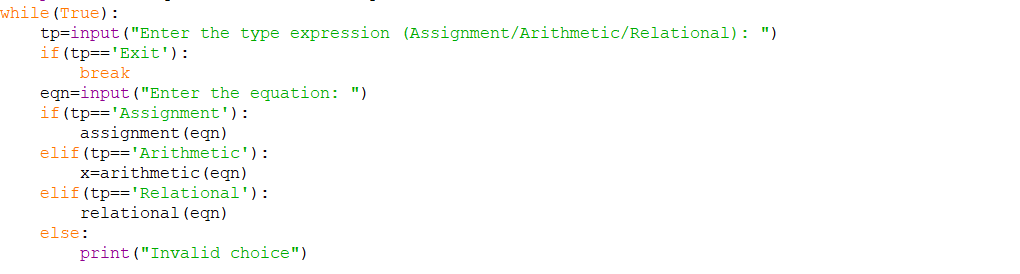
register to the variable on the LHS.

• For relational expressions, write the conditional statements and the Boolean values returned by the execution of the statements.

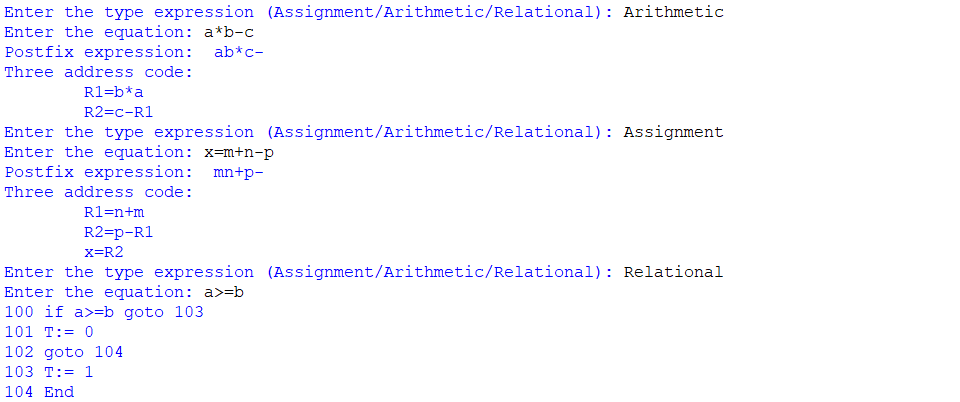
**CODE:**







**OUTPUT:**



**RESULT:** A program that produces the postfix notation and the three-address code for the given expression was written and executed in python successfully.