Ishita Hardasmalani
C14-2103058

THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

EXPERIMENT NO. 5

Aim: write a program to implement Three Address code

Theory:

Three address code (TAC) is a low level intermediate representation used in compilers & interpreters for representing statement in a program. It's designed to be simple & easy to manipulate, making it a convenient from various optimization & code generation tasks. In TAC, each instruction typically contains three addresses, hence the name "three address code". These addresses can represent operand, result & sometime control flow targets. The format of a typically TAC instruction is:

result = operand1 operator operand2

Here, result is the variable that will store the result of the operation specified by 'operator' on 'operand1' & 'operand2'.

For example, a = b + c * d

this expression can be represented in TAC as:

t1 = c * d
t2 = b + t1
a = t2

In this TAC representation:

- 't1', 't2' & 'a' are temporary variable
- 'c * d' is represented by 't1'
- 'b + t1' is represented by 't2'

& 't2' is assigned to 'a'.

THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

TAC is also used for representing control flow constructs such as conditionals & loops. For example,

considering the following if-else statement:

```
if X > 0:
        Y = X
    else:
        Y = -X
```

This can be represented in TAC as:

```
if X > 0 goto L1
Y = -X
  goto L2
L1: Y = X
L2:
```

Here, L1 & L2 are labels used for branching

TAC provides a simplified view of the program's computation & control flow, making it easier for subsequent compiler stages to perform optimization or generate machine code.

28/3/24

A+