Three Address Code ...

Learn Vid fun !

In three address code, at most three addresses are used to represent any statement.

General form:

The general form of three address code is-

$$a = b \ op \ c$$

where-

- a, b, c are the operands that can be names, constants or compiler generated temporaries
- op represents the operator

Example:

i) $a = b + c$

ii) $c = a * b$

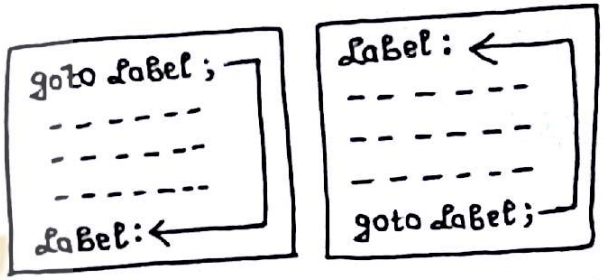
Common Three Address Instruction Forms

(I) **Assignment statements :-**

- $x = y \text{ op } z$
- $x = \text{op } y$

(II) **Copy statement :-** $x = y$

(III) **Conditional jump :-** if x relop y goto L



(IV) **Unconditional jump :-** goto L

(V) **Procedure call :-**

param x call p
return y

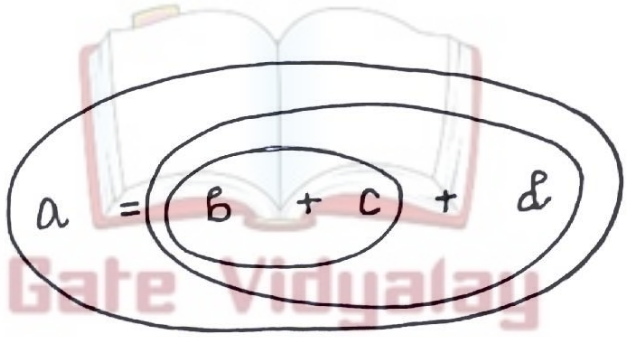
```
p (x)
{
  .....
  return y ;
}
```

Generation of 3 address code

Illustration-1: Generate 3 address code for -

$$a = b + c + d$$

Soln.:



- '+' operator has higher precedence over '=' operator
- '+' operator is left associative.

Three address code will be -

$$T1 = b + c$$

$$T2 = T1 + d$$

$$a = T2$$

Illustration-2: Generate 3 address code for -

$$-(a*b) + (c+d) - (a+b+c+d)$$

→ Unary operators have higher precedence over Binary operators.

Soln: Three address code will be -

$t1 = a * b$

$t2 = \text{uminus } t1$

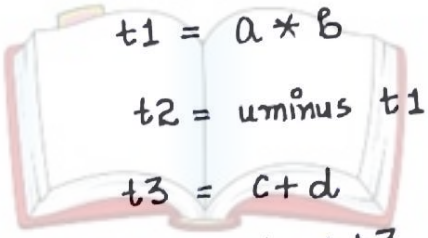
$t3 = c + d$

$t4 = t1 + t3$

$t5 = a + b$

$t6 = t3 + t5$

$t7 = t4 - t6$



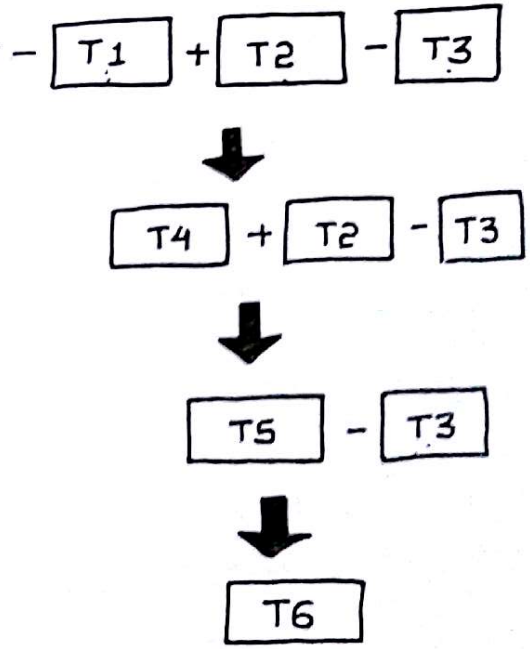


Illustration-3: Generate 3 address code for -

```
if A < B
  then  1
  else  0
```

Soln:

```
(1)  if (A < B)  goto (4)
(2)  T1 = 0
(3)  goto (5)
(4)  T1 = 1
(5)
```

Illustration-4: Generate 3 address code for -

" If a<b and c<d then t=1 else t=0 "

Soln:

(1) If (a<b) goto (3)
(2) goto (4)
(3) If (c<d) goto (6)
(4) t=0
(5) goto (7)
(6) t=1
(7)