**EX.NO**:**8**   
**DATE:**

**INTERMEDIATE CODE GENERATION**

**AIM:**

To implement intermediate code generation using C. **ALGORITHM:**

**Step 1**: Start.

**Step 2**: Get address code sequence.

**Step 3**: Determine current location of 3 using address (for 1st operand).

**Step 4**: If current location not already exist generate move (B,O).

**Step 5**: Update address of A(for 2nd operand). Step 6: If current value of B and () is null,exist. Step 7: If they generate operator () A,3 ADPR. Step 8: Store the move instruction in memory. Step 9: Stop.

**PROGRAM:**

#include<stdio.h>

int main()

{

char input[50],operators[5]="/\*+-"; printf("%s","Enter the arithmetic Expression : "); scanf("%s",input);

int k=1;

for(int i=0;i<4;i++)

{

for(int j=2;input[j]!='\0';j++)

{

if(operators[i]==input[j])

{

printf("t%d := ",k);

if(input[j-1]>='0' && input[j-1]<='9')

printf("t%c",input[j-1]); else

printf("%c",input[j-1]);

printf("%c",input[j]); if(input[j+1]>='0' && input[j+1]<='9') printf("t%c",input[j+1]); else

printf("%c",input[j+1]);

input[j-1]=k+48; int g=j;

for(;input[g+2]!='\0';g++)

{

input[g]=input[g+2];

}

input[g]='\0'; k++;

printf("\n");

}

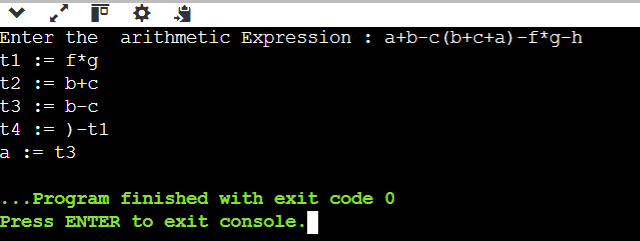
}

}

printf("%c := t%c",input[0],input[2]);

}

**OUTPUT:**



**RESULT:**

Thus, the C program to implement Intermediate Code Generator has been executed and verified successfully.