Experiment XV

**Aim**: Implement the back end of the compiler which takes the three address codes and produces the 8086 assembly language instructions that can be assembled and run using an 8086 assembler. The target assembly instructions can be a simple move, add, sub, jump, etc.

Algorithm

1. Start
2. Input the number of alphabets, number of states, number of transitions.
3. Enter the intermediate codes.
4. For each operation, convert it into the following commands
   1. MOV operand1, register
   2. OPERATOR register, operand2
   3. MOV operand1, operand1
5. The target code is obtained as output.
6. Stop

Output

Enter the set of intermediate code(terminated by exit)

a=b+c

b=d/e

exit

Target code generation

Mov b, R 0

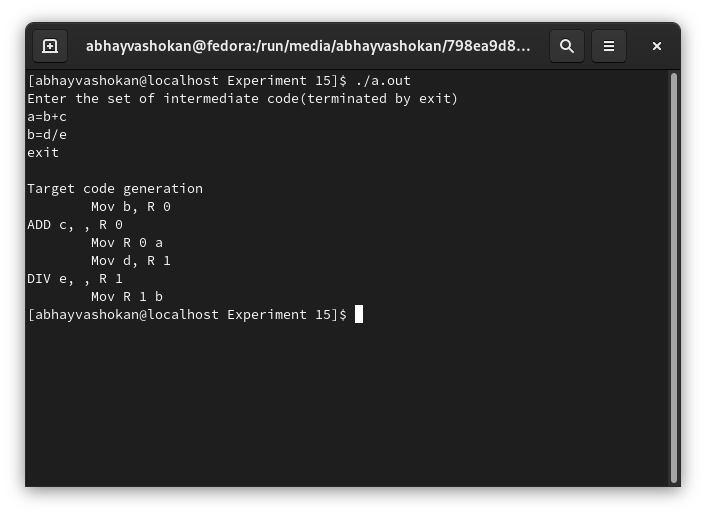
ADD c, , R 0

Mov R 0 a

Mov d, R 1

DIV e, , R 1

Mov R 1 b

Screenshot

Readme

1. Compile and run the C program using the command

**gcc 2Abhay-P415.c && ./a.out**

2. Enter the set of intermediate codes as input

7. The target code is obtained as output.

**Result**: Successfully implemented the back end of the compiler which takes the three address codes and produces the 8086 assembly language instructions that can be assembled and run using an 8086 assembler.