Name: Gokulnath M Prabhu
Class: CS7B
Roll No: 21

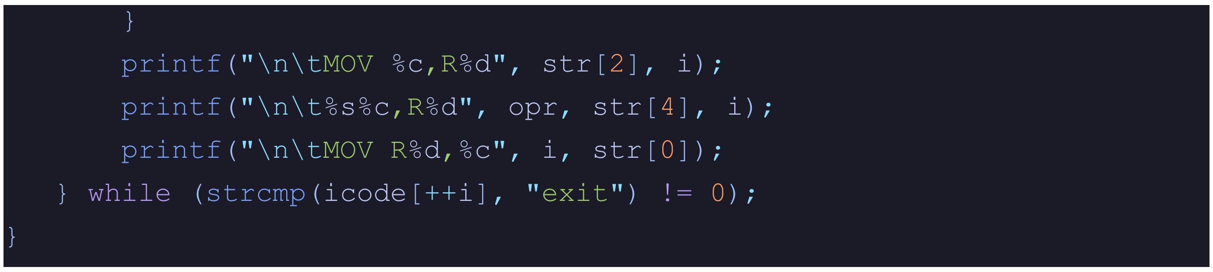
Lab Cycle 4 - Experiment 19

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

Code:

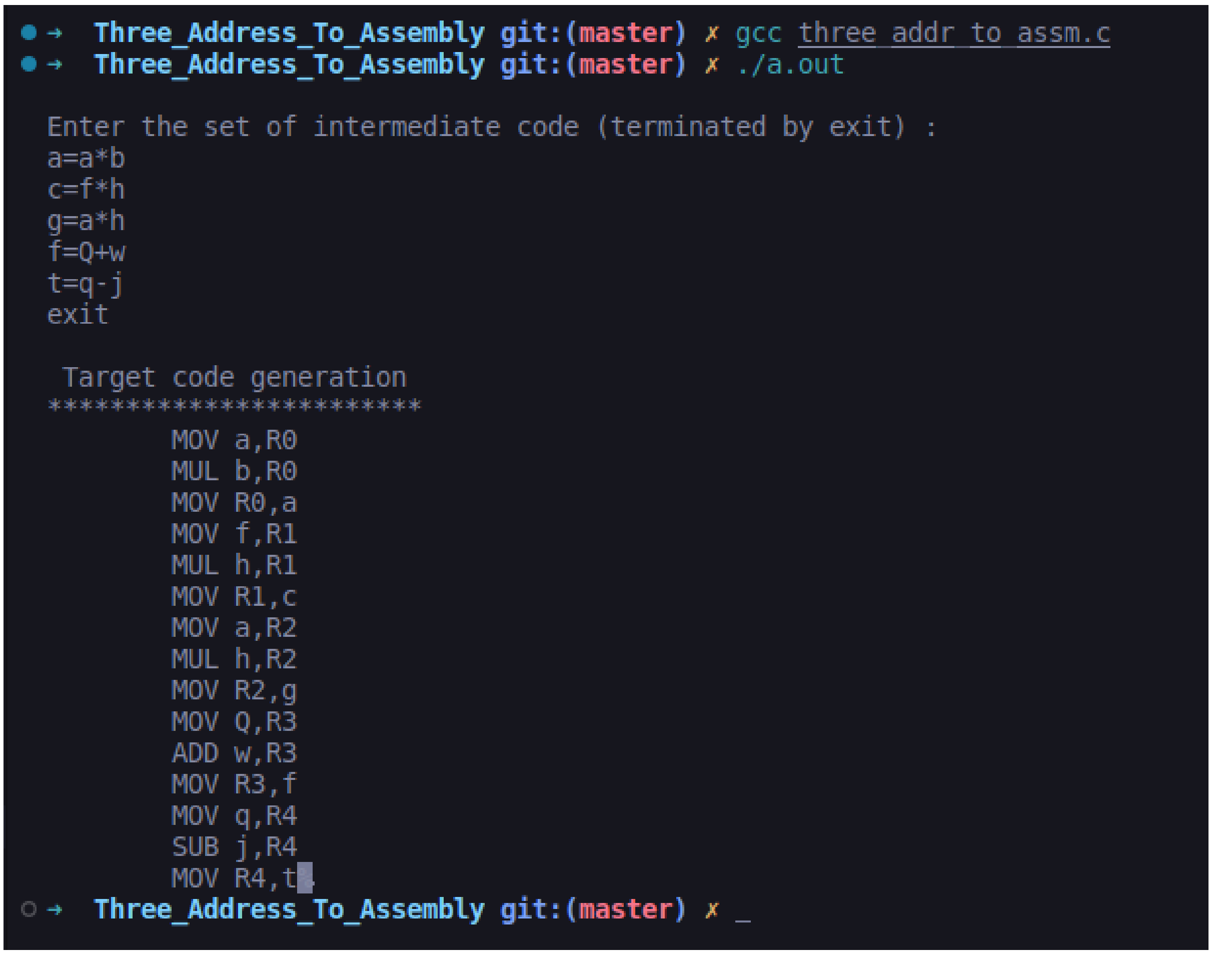
```
#include <stdio.h>
#include <stdio.h>
#include <string.h>

void main()
{
   char icode[10][30], str[20], opr[10];
   int i = 0;
   printf("\nEnter the set of intermediate code (terminated by exit)
:\n");
   do
   {
       scanf("%s", icode[i]);
   } while (strcmp(icode[i++], "exit") != 0);
   printf("\n Target code generation");
   printf("\n************************");
   i = 0;
   do
   {
       strcpy(str, icode[i]);
       switch (str[3])
       {
       case '+':
           strcpy(opr, "ADD ");
           break;
       case '-':
           strcpy(opr, "SUB ");
           break;
       case '*':
           strcpy(opr, "MUL ");
           break;
       case '/':
           strcpy(opr, "DIV ");
           break;
```



```c
        }
        printf("\n\tMOV %c,R%d", str[2], i);
        printf("\n\t%s%c,R%d", opr, str[4], i);
        printf("\n\tMOV R%d,%c", i, str[0]);
    } while (strcmp(icode[++i], "exit") != 0);
}
```

Output:



```
● ➜  Three_Address_To_Assembly git:(master) ✗ gcc three_addr_to_assm.c
● ➜  Three_Address_To_Assembly git:(master) ✗ ./a.out

 Enter the set of intermediate code (terminated by exit) :
 a=a*b
 c=f*h
 g=a*h
 f=Q+w
 t=q-j
 exit

  Target code generation
 ************************
        MOV a,R0
        MUL b,R0
        MOV R0,a
        MOV f,R1
        MUL h,R1
        MOV R1,c
        MOV a,R2
        MUL h,R2
        MOV R2,g
        MOV Q,R3
        ADD w,R3
        MOV R3,f
        MOV q,R4
        SUB j,R4
        MOV R4,t%
○ ➜  Three_Address_To_Assembly git:(master) ✗ _
```