Cycle 4

Malavika R Vikraman

CS7B – Roll No :31

Question 2

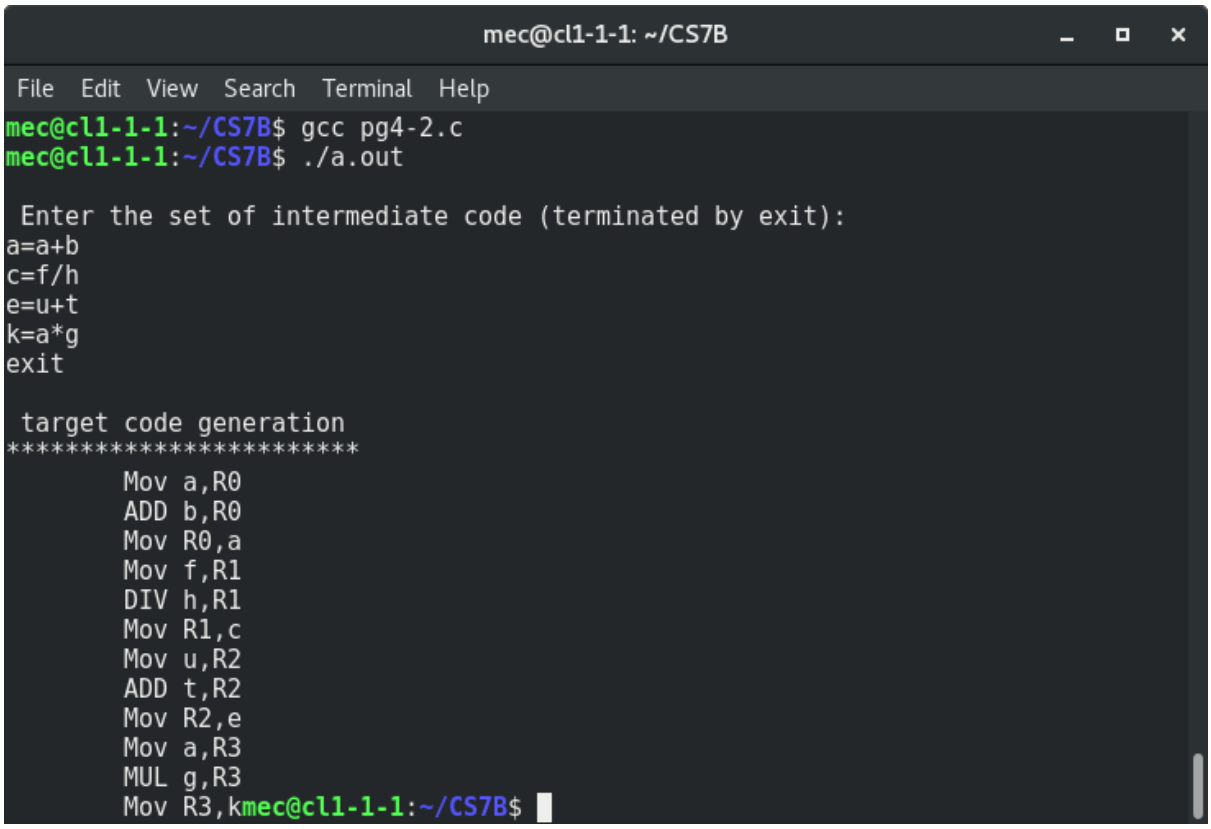
Implement the back end of the compiler which takes the three address code and produces the 8086 assembly language instructions that can be assembled and run using an 8086 assembler. The target assembly instructions can be simple move, add, sub, jump etc

Program :

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

void main()
{
    char icode[10][30], str[20], opr[10];
    int i = 0;
    //clrscr();
    printf("\n Enter 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;
      }
      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);
    //getch();
  }
```

Output:


mec@cl1-1-1: ~/CS7B
File Edit View Search Terminal Help
mec@cl1-1-1:~/CS7B$ gcc pg4-2.c
mec@cl1-1-1:~/CS7B$ ./a.out
Enter the set of intermediate code (terminated by exit):
a=a+b
c=f/h
e=u+t
k=a*g
exit
target code generation
****
Mov a,R0
ADD b,R0
Mov R0,a
Mov f,R1
DIV h,R1
Mov R1,c
Mov u,R2
ADD t,R2
Mov R2,e
Mov a,R3
MUL g,R3
Mov R3,kmec@cl1-1-1:~/CS7B$