```
/*
Name:Shubham Satish Tembhurkar
PRN:1641060
Class:L.Y.B.TECH.
Batch:B1
Subject:CCL
Aim:Write a C program to generate machine code from abstract
syntax tree generated by the parser.
*/

#include<stdio.h>
//#include<conio.h>
#include<string.h>
#include<stdlib.h>
struct quadraple
{
      int pos;
      char op;
      char arg1[5];
      char arg2[5];
      char result[5];
}quad[15];
int n=0;
void assignment(int);
void uminus(int );
void explore();
void codegen(char op[5],int);
char tuple[15][15];
int main(void)
{
      FILE *src;
      int nRetInd,i;
      char str[15];
//    clrscr();
      src=fopen("code.txt","r");
      fscanf(src,"%s",str);
      while(!feof(src))
      {
            strcpy(tuple[n++],str);
            fscanf(src,"%s",str);
      }
      printf("INPUT:\nIntermiate codes:\n");
      for(i=0;i<n;i++)
            printf("%s\n",tuple[i]);
      explore();
//    getch();
//    clrscr();
      printf("OUTPUT:\n");
      printf("Quadruple: \n");
      printf("pos\topr\targ1\targ2\tresult\n");
      for(i=0;i<n;i++)

      printf("\n%d\t%c\t%s\t%s\t%s",quad[i].pos,quad[i].op,quad[i]
.arg1,quad[i].arg2,quad[i].result);
```

```
      i=0;
      printf("\n\ncode generated :\n");
      while(i<n)
      {
            if(quad[i].op=='+')
                  codegen("ADD",i);
            if(quad[i].op=='=')
                  assignment(i);
            if(quad[i].op=='-')
                        if(!strcmp(quad[i].arg2,"\0"))
                              uminus(i);
                        else
                              codegen("SUB",i);
            if(quad[i].op=='*')
                  codegen("MUL",i);
            if(quad[i].op=='/')
                  codegen("DIV",i);
      i++;
      }
//    getch();
      _fcloseall();
      return 0;
}
void codegen(char op[5],int t)
{
      char str[25];
      printf("MOV %s,R0\n",quad[t].arg1);
      printf("%s %s,R0\n",op,quad[t].arg2);
      printf("MOV R0,%s\n",quad[t].result);
}
void assignment(int t)
{
      char str[25];
      printf("MOV %s,%s\n",quad[t].arg1,quad[t].result);
}
void uminus(int t)
{
      char str[25];
      printf("MOV R0,0\n");
      printf("SUB %s,R0\n",quad[t].arg1);
      printf("MOV R0,%s\n",quad[t].result);
}

void explore()
{
      int i,j,t,t1,t2;
      for(i=0;i<n;i++)
      {
            quad[i].pos=i;
            for(j=0,t=0;j<strlen(tuple[i])&&tuple[i][j]!='=';j++)
            {
                  quad[i].result[t++]=tuple[i][j];
            }
            t1=j;
            quad[i].result[t]='\0';
            if(tuple[i][j]=='=')
            {
                  quad[i].op='=';
            }
            if(tuple[i][j+1]=='+'||tuple[i][j+1]=='-
'||tuple[i][j+1]=='*'||tuple[i][j+1]=='/')
            {
                  quad[i].op=tuple[i][j+1];
                  t1=j+1;
            }

      for(j=t1+1,t=0;j<strlen(tuple[i])&&tuple[i][j]!='+'&&tuple[i
][j]!='-'&&tuple[i][j]!='*'&&tuple[i][j]!='/';j++)
            {
                  quad[i].arg1[t++]=tuple[i][j];
            }
            t2=j;
            quad[i].arg1[t]='\0';
            if(tuple[i][j]=='+'||tuple[i][j]=='-
'||tuple[i][j]=='*'||tuple[i][j]=='/')
                  {
                        quad[i].op=tuple[i][j];
                  }
            for(j=t2+1,t=0;j<strlen(tuple[i]);j++)
            {
                  quad[i].arg2[t++]=tuple[i][j];
            }
            quad[i].arg2[t]='\0';
      }
}
/**************OUTPUT*************
gcoej@gcoej-ThinkCentre-M70z:~$ gcc codegen.c
gcoej@gcoej-ThinkCentre-M70z:~$ ./a.out
INPUT:
Intermiate codes:
t1=b*c
t2=d*f
t3=t1+a
t4=t3+t2
t5=t4+g
a=t5
OUTPUT:
Quadruple:
pos   opr   arg1  arg2  result

0     *     b     c     t1
1     *     d     f     t2
2     +     t1    a     t3
3     +     t3    t2    t4
4     +     t4    g     t5
5     =     t5          a

code generated :
```

```
MOV b,R0
MUL c,R0
MOV R0,t1
MOV d,R0
MUL f,R0
MOV R0,t2
MOV t1,R0
ADD a,R0
MOV R0,t3
MOV t3,R0
ADD t2,R0
MOV R0,t4
MOV t4,R0
ADD g,R0
MOV R0,t5
MOV t5,a
gcoej@gcoej-ThinkCentre-M70z:~$
*/
```