#include<stdio.h>

#include<stdlib.h>

#include<string.h>

//Structure for the symbol Table

struct Sym\_Tab

{

char symbol[10];

int val;

};

//structure of literal table

struct Literal\_Table

{

char literal[10];

int val;

};

void main ()

{

FILE \*f1;

struct Sym\_Tab ST[20];

struct Literal\_Table LT[10];

int i,j;

int LC, ltc = 0, stc = 0;

char label[20], opcode[20], operand[20];

//initialize the literal table

for(i = 0; i<10; i++)

LT[i].val = -1;

f1=fopen("Assembly\_Program3.txt","r"); //Read the file containing assembly language program

fscanf(f1,"%s%s%s",label, opcode, operand);

printf("\n The Source Program is");

printf("\n ------------------------------------------- ");

printf("\n Label\tOpcode\tOperand");

printf("\n -------------------------------------------\n ");

while(strcmp(opcode, "END") != 0)

{

printf("\n %s \t%s \t%s",label, opcode, operand);

//Generation of Symbol Table and Literal Table

i = 0;

while(operand[i] != '\0')

{

if(operand[i] == '=')

{

j = 0;

while(operand[i] != '\0')

LT[ltc].literal[j++] = operand[i++]; //copy the literals

LT[ltc].literal[j] = '\0';

ltc++;

}

else

i++;

}

if(strcmp(opcode, "START") == 0)

{

if(strcmp(operand, "-") != 0)

LC = atoi(operand)-1;

else

LC = -1;

}

if(strcmp(opcode, "EQU") == 0)

{

strcpy(ST[stc].symbol, label);

ST[stc].val = atoi(operand);

stc++;

LC--;

}

else if(strcmp(label, "-") !=0)

{

strcpy(ST[stc].symbol, label);

ST[stc].val = LC;

stc++;

}

if(strcmp(opcode, "ORIGIN") == 0)

{

LC = atoi(operand)-1;

}

if(strcmp(opcode, "LTORG") == 0)

{

for(i = 0; i<ltc; i++)

{

LT[i].val = LC; //copy the literal address

LC++;

}LC--;

}

LC++;

fscanf(f1,"%s%s%s",label, opcode, operand);

}

fclose(f1);

for(i = 0; i<ltc; i++)

{

if(LT[i].val == -1)

{

LT[i].val = LC; //copy the literal address

LC++;

}

}

printf("\n The Symbol Table is ");

printf("\n ------------------------------------------- ");

printf("\nSymbol\tAddress");

printf("\n -------------------------------------------\n");

for(i = 0; i<stc; i++)

printf( "%s\t%d\n",ST[i].symbol,ST[i].val);

printf("\n The Literal Table is ");

printf("\n ------------------------------------------- ");

printf("\nLiteral\tAddress");

printf("\n -------------------------------------------\n");

for(i = 0; i<ltc; i++)

printf("%s \t %d \n",LT[i].literal, LT[i].val);

printf("\n");

}