PROGRAM

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<ctype.h>

#include<conio.h>

int main()

{

FILE \*fint,\*ftab,\*flen,\*fsym,\*fout;

int op1[10],txtlen,txtlen1,i,j=0,len,count=0,c=0;

char add[5],symadd[5],op[5],start[10],temp[30],line[20],label[20],mne[10],operand[10],symtab[10],opmne[10];

fint=fopen("intermediate1.txt","r");

flen=fopen("length.txt","r");

ftab=fopen("optab1.txt","r");

fsym=fopen("symbol.txt","r");

fout=fopen("Output.txt","w");

fscanf(fint,"%s\t%s\t%s\t%s",add,label,mne,operand);

if(strcmp(mne,"START")==0) //mneumonic code is start

{

strcpy(start,operand);

fscanf(flen,"%d",&len);

}

printf("H^%s^00%s^%d\nT^00%s^09^",label,start,len,start);

fprintf(fout,"H^%s^00%s^%d\nT^00%s^09^",label,start,len,start);

fscanf(fint,"%s\t%s\t%s\t%s",add,label,mne,operand);

while((c = fgetc(fint)) != EOF) //repeat till END mneumonic code

{

fscanf(ftab,"%s\t%s",opmne,op);

while((c = fgetc(ftab)) != EOF)

{

if(strcmp(mne,opmne)==0) //if mneumonic code found in optab

{

//fclose(ftab);

fscanf(fsym,"%s\t%s",symadd,symtab);

while((c = fgetc(fsym)) != EOF)

{

if(strcmp(operand,symtab)==0) //if symbolic operand found in symtab

{

printf("%s%s^",op,symadd);

fprintf(fout,"%s%s^",op,symadd);

count++;

break;

}

else

{

fscanf(fsym,"%s\t%s",symadd,symtab);

}

}

break;

}

else

{

fscanf(ftab,"%s\t%s",opmne,op);

}

}

if((strcmp(mne,"BYTE")==0)||(strcmp(mne,"WORD")==0)) //mneumonic code is a byte or word

{

if(strcmp(mne,"WORD")==0)

{

printf("00000%s^",operand);

count++;

fprintf(fout,"00000%s^",operand);

}

else

{

len=strlen(operand);

for(i=2;i<(len-1);i++)

{

printf("%X",operand[i]);

fprintf(fout,"%X",operand[i]);

}

printf("^");

fprintf(fout,"^");

count++;

}

}

fscanf(fint,"%s\t%s\t%s\t%s",add,label,mne,operand);

if(count==3)

{

count=0;

printf("\nT^00%s^09^",add);

fprintf(fout,"\nT^00%s^09^",add);

}

//ftab=fopen("optab.txt","r");

fseek(ftab,SEEK\_SET,0);

fseek(fsym,SEEK\_SET,0);

}

printf("\nE^00%s",start);

fprintf(fout,"\nE^00%s",start);

fclose(fint);

fclose(ftab);

fclose(fsym);

fclose(flen);

fclose(fout);

getch();

return 0;

}