**PASS 1- ASSEMBLER**

**CODE:**

#include<stdio.h>

#include<string.h>

int main()

{

char id[10],str[10],

mot[][50]={{"LOAD"},{"STORE"},{"DEC"},{"JNZ"},{"ADD"}},

pot[][50]={{"START"},{"END"},{"DC"}},

st[][50]={{"REPEAT"},{"N1"},{"N2"},{"COUNT"}},

lt[][50]={{"= '0'"},{"= '1'"}},

test[][50]={{""},{""}};

char ch[10]={0};

int x,i,j,k,parsed,motlen,potlen,stlen,ltlen,oc=0;

FILE \*fp1,\*fpmot,\*fppot,\*fpst,\*fplt;

fp1=fopen("pass1File.txt","r");

fpmot=fopen("mot.txt","a");

fppot=fopen("pot.txt","a");

fpst=fopen("symbol.txt","a");

fplt=fopen("literal.txt","a");

motlen=sizeof(mot)/sizeof(mot[0]);

potlen=sizeof(pot)/sizeof(pot[0]);

stlen=sizeof(st)/sizeof(st[0]);

ltlen=sizeof(lt)/sizeof(lt[0]);

//printf("%d %d %d %d\n",motlen,potlen,stlen,ltlen);

for(k=0;fgets(str,50,fp1)!=NULL;k+=2); //to find address of END

k-=2;

fclose(fp1);

fp1=fopen("pass1File.txt","r");

for(x=0;fgets(ch,50,fp1)!=NULL;x+=2)

{

parsed=0;

printf("%d %s",x,ch);

for(i=0;i<motlen&&parsed==0;i++)

{

if(strstr(ch,mot[i]))

{

strcpy(str,strstr(ch,mot[i]));

oc=0;

for(j=0;j<strlen(str);j++)

{

if(str[j]==' ')

oc++; //calc operands

}

//printf("\n1. %d %s",oc,mot[i]);

fprintf(fpmot,"\n%s\t\t2\t\t%d",mot[i],oc);

parsed=1;

break;

}

}

for(i=0;i<potlen&&parsed==0;i++)

{

if(strstr(ch,pot[i]))

{

// printf("\n2. %d %s",x,pot[i]);

fprintf(fppot,"\n%s\t\t%d",pot[i],x);

parsed=1;

break;

}

}

//symbols are followed by pseudo-opcodes

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

{

parsed=0;

for(j=0;j<potlen&&parsed==0;j++)

{

//printf("\n %s",pot[j]);

if((strstr(ch,":")&&strstr(ch,st[i])) || (strstr(ch,pot[j])&&strstr(ch,st[i])))

{

//printf("\n3. %d %s",x,st[i]);

fprintf(fpst,"\n%s\t%d\t2\tR",st[i],x);

parsed=1;

break;

}

}

if(parsed==1)

i++; //to not let N1 twice

}

for(i=0;i<ltlen&&parsed==0;i++)

{

if((strstr(ch,"=")&&strstr(ch,lt[i])))

{

// printf("\n4. %d %s",k,lt[i]);

fprintf(fplt,"\n%s\t%d\t2\tR",lt[i],k);

parsed=1;

break;

}

}

}

fclose(fp1);

fclose(fpmot);

fclose(fppot);

fclose(fpst);

fclose(fplt);

printf("\nPass 1 Complete...Tables updated...");

return 0;

}

**INPUT:**

START

LOAD N1

STORE COUNT

B1 = '0'

REPEAT: ADD N1

DEC COUNT

JNZ REPEAT

N1 DC 07

N2 DC 08

COUNT DC ?

END

**OUTPUT:**

0 START

2 LOAD N1

4 STORE COUNT

6 B1 = '0'

8 REPEAT: ADD N1

10 DEC COUNT

12 JNZ REPEAT

14 N1 DC 07

16 N2 DC 08

18 COUNT DC ?

20 END

Pass 1 Complete...Tables updated...

**MOT:**

MNENONIC LENGTH NO. OF OPERANDS

LOAD 2 1

STORE 2 1

ADD 2 1

DEC 2 1

JNZ 2 1

**POT:**

PSEUDO OPCODE VALUE

START 0

DC 14

DC 16

DC 18

END 20

**SYMBOLS:**

SYMBOL VALUE LENGTH R/A

REPEAT 8 2 R

N1 14 2 R

N2 16 2 R

COUNT 18 2 R

**LITERALS:**

LITERAL VALUE LENGTH R/A

= '0' 20 2 R