#include<stdio.h>

#include<string.h>

char ch,x[20];

int i,j,flag;

void print()

{

if((strcmp(x,"include")==0)||(strcmp(x,"void")==0)||(strcmp(x,"char")==0)||(strcmp(x,"float")==0)||(strcmp(x,"double")==0))

{

printf("\n%s-->keyword",x);

}

else

{

if((strcmp(x,"stdio")==0)||(strcmp(x,"conio")==0)||(strcmp(x,"string")==0))

{

printf("\n%s-->Headerfile",x);

}

else

{

if((strcmp(x,"main")==0)||(strcmp(x,"printf")==0)||

(strcmp(x,"scanf")==0))

{

printf("\n%s-->function",x);

}

else

{

if(x[0]!='\0')

printf("\n%s-->identifier",x);

}

}

}

for(i=0;i<20;i++)

x[i]='\0';

i=0;

}

int main()

{

FILE\*ptr;

char infile[12];

printf("\n enter source file name-");

gets(infile);

ptr=fopen(infile,"r");

if(ptr==NULL)

{

printf("\n could not open the file");

}

printf("\n uniform symbol table");

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

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

x[i]='\0';

i=flag=0;

while(!feof(ptr))

{

ch=fgetc(ptr);

switch(ch)

{

case'#':

if(flag==2)

print();

flag=0;

printf("\n%c-->terminals",ch);

break;

case'<':

if(flag==2)

print();

flag=0;

printf("\n%c-->terminals",ch);

break;

case'>':

if(flag==2)

printf("\n%c-->terminals",ch);

break;

case'(':

if(flag==2)

print();

flag=0;

printf("\n%c-->terminals",ch);

break;

case')':

if(flag==2)

print();

flag==0;

printf("\n%c-->terminals",ch);

break;

case'{':

if(flag==2)

print();

flag=0;

printf("\n%c-->terminals",ch);

break;

case'}':

if(flag==2)

print();

flag==0;

printf("\n%c-->terminals",ch);

break;

case',':

if(flag==2)

print();

flag=0;

printf("\n%c-->terminals",ch);

break;

case';':

if(flag==2)

print();

flag=0;

printf("\n%c-->terminals",ch);

break;

case'%':

if(flag==2)

print();

flag=0;

printf("\n%c-->terminals",ch);

break;

case'&':

if(flag==2)

print();

flag=0;

printf("\n%c-->terminals",ch);

break;

case'\*':

if(flag==2)

print();

flag=0;

printf("\n%c-->terminals",ch);

break;

case'-':

if(flag==2)

print();

flag=0;

printf("\n%c-->terminals",ch);

break;

case'+':

if(flag==2)

print();

flag=0;

printf("\n%c-->terminals",ch);

break;

case'=':

if(flag==2)

print();

flag=0;

printf("\n%c-->terminals",ch);

break;

case'"':

if(flag==2)

print();

flag=0;

printf("\n%c-->terminals",ch);

break;

case'0':

flag=1;

printf("\n%c-->Literal",ch);

break;

case'1':

flag=1;

printf("\n%c-->Literal",ch);

break;

case'2':

flag=1;

printf("\n%c-->Literal",ch);

break;

case'3':

flag=1;

printf("\n%c-->Literal",ch);

break;

case'4':

flag=1;

printf("\n%c-->Literal",ch);

break;

case'5':

flag=1;

printf("\n%c-->Literal",ch);

break;

case'6':

flag=1;

printf("\n%c-->Literal",ch);

break;

case'7':

flag=1;

printf("\n%c-->Literal",ch);

break;

case'8':

flag=1;

printf("\n%c-->Literal",ch);

break;

case'9':

flag=1;

printf("\n%c-->Literal",ch);

break;

case'\n':

break;

case'\t':

break;

case' ':

print();

break;

default:

flag=2;

x[i]=ch;

i++;

}

}

fclose(ptr);

return;

}