#include<stdio.h>

#include<string.h>

#include<stdlib.h>

char\* kw[]={"int","float","double","char","void","if","else","for","while","do","switch","case","continue","break","void"};

char SP[20]={'{','(',')','}',',',';'};

//char\* fc[]={"main","printf","scanf","sizeof"};

int sing\_flag=0,multi\_flag=0,FC\_flag=0,str\_flag=0,id\_flag=0;

int isletter(char str)

{

if(!((str=='\_') || (str>='A' && str <='Z') || (str>='a' && str <='z')))

return 0;

return 1;

}

void substring(char s[], char sub[], int p, int l)

{

int i=0;

while (p < l)

{

sub[i++] = s[p++];

}

sub[p] = '\0';

}

int is\_digit(char num)

{

if(num>='0' && num<='9')

{

return 1;

}

return 0;

}

int isID(char str[])

{

int i=0;

if(!isletter(str[0]))

return 0;

for(i=1;i<strlen(str);i++)

{

if(isletter(str[i])||is\_digit(str[i]))

continue;

else

return 0;

}

return 1;

}

int is\_digits(char num[])

{

int i;

if(strlen(num)==0)

return 0;

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

{

if(is\_digit(num[i]))

{

continue;

}

else

return 0;

}

return 1;

}

int isoptfrac(char num[])

{

int c=0,found=0;

for(c=0;c<strlen(num);c++)

{

if(num[c]=='.')

{

break;

found=1;

}

}

if(!found)

{

return 0;

}

if(c>0 && c<strlen(num))

{

char word[20];

substring(num,word,c+1,strlen(num));

if(is\_digits(word))

{

substring(num,word,0,c);

if(is\_digits(word))

return 1;

}

}

return 0;

}

int isoptexp(char num[])

{

int c=0,found=0;

for(c=0;c<strlen(num);c++)

{

if(num[c]=='E')

{

break;

found=1;

}

}

if(!found)

{

return 0;

}

if(c>0 && c<strlen(num))

{

char word[20];

if(num[c+1]=='+'||num[c+1]=='-')

substring(num,word,c+2,strlen(num));

else if(is\_digit(num[c+1]))

substring(num,word,c+1,strlen(num));

else

return 0;

if(is\_digits(word)||isoptfrac(word))

{

substring(num,word,0,c);

if(is\_digits(word)||isoptfrac(word))

return 1;

}

}

return 0;

}

int isSP(char s)

{

int i=0;

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

{

if(s==SP[i])

return 1;

}

return 0;

}

int iskeyword(char str[])

{

int i;

int n=sizeof(kw)/sizeof(kw[0]);

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

{

if(!strcmp(str,kw[i]))

return 1;

}

return 0;

}

char\* test\_token(char str[])

{

int i=0;

if(str[strlen(str)-1]=='\n')

str[strlen(str)-1]='\0';

if(multi\_flag==1)

{

if(!strcmp(str,"\*/"))

multi\_flag=0;

return "";

}

if(sing\_flag==1)

return "";

if(FC\_flag==1 || FC\_flag==2)

{

if(FC\_flag==1)

{

if(str[strlen(str)-1]==')')

{

FC\_flag=2;

}

}

else

{

if(str[0]=='(')

{

FC\_flag=1;

}

else if(str[0]==';')

{

FC\_flag=0;

}

}

return "";

}

if(str\_flag==1)

{

if(str[0]=='"')

{

str\_flag=0;

}

return "";

}

if(str[0]=='/')

{

if(str[1]=='\*')

{

multi\_flag=1;

return "MULTI";

}

else if(str[1]=='/')

{

sing\_flag=1;

return "SINGLE";

}

else

return "ARITHOP";

}

else if((str[0]=='+' || str[0]=='\*'|| str[0]=='-'||str[0]=='%') && (str[1]=='\0'))

return "ARITHOP";

else if(isSP(str[0]))

return "SP";

else if(str[0]=='<' || str[0]=='>')

return "RELOP";

else if(str[0]=='=')

{

if(str[1]=='=')

return "RELOP";

else if(str[1]=='\0')

return "ASSIGN";

}

else if(str[0]=='!')

{

if(str[1]=='=')

return "RELOP";

else if(str[1]=='\0')

return "LOGICOP";

}

else if((!strcmp(str,"&&"))||(!strcmp(str,"||")))

return "LOGICOP";

else if(iskeyword(str))

return "KW";

else if(str[0]=='\'')

{

if(str[strlen(str)-1]=='\'')

return "CHARCONST";

}

else if(str[0]=='"')

{

str\_flag=1;

if(str[strlen(str)-1]=='"')

return "STRCONST";

}

else if(is\_digits(str)||isoptfrac(str)||isoptexp(str))

{

return "NUMCONST";

}

else if(isID(str))

{

id\_flag=1;

return "ID";

}

return str;

}

int reset\_flag()

{

FC\_flag=0;

str\_flag=0;

sing\_flag=0;

id\_flag=0;

}

int is\_delimiter(char str)

{

char word[20]={'(',',',')',';','"'};

int i;

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

{

if(str==word[i])

return 1;

}

return 0;

}

int is\_opr1(char str)

{

char word[20]={'=','+','<','>','!','&','|','-','\*','/','%'};

int i;

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

{

if(str==word[i])

return 1;

}

return 0;

}

int is\_opr2(char str)

{

char word[20]={'=','&','|','/','\*'};

int i;

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

{

if(str==word[i])

return 1;

}

return 0;

}

void LA(char stmt[200])

{

int i=0,c=0,j,k=0,ld=0,hold=0;

reset\_flag();

char \* token = strtok(stmt, " ");

char tokens[20][20],word[200]="",word1[20]="";

while( token != NULL )

{

strcpy(tokens[c++],token);

token = strtok(NULL, " ");

}

char sub\_tk[10][20];

while(i<c)

{

//test\_token(tokens[i++]2

for(j=0;j<strlen(tokens[i]);j++)

{

if(is\_delimiter(tokens[i][j]))

{

if(ld!=j)

{

substring(tokens[i],sub\_tk[k],ld,j);

sub\_tk[k++][j-ld]='\0';

}

sub\_tk[k][0]=tokens[i][j];

sub\_tk[k++][1]='\0';

ld=j+1;

}

else if(is\_opr1(tokens[i][j]))

{

if(is\_opr2(tokens[i][j+1]))

{

substring(tokens[i],sub\_tk[k],ld,j);

sub\_tk[k++][j-ld]='\0';

substring(tokens[i],sub\_tk[k],j,j+2);

sub\_tk[k++][2]='\0';

j++;

ld=j+1;

}

else

{

substring(tokens[i],sub\_tk[k],ld,j);

sub\_tk[k++][j-ld]='\0';

sub\_tk[k][0]=tokens[i][j];

sub\_tk[k++][1]='\0';

ld=j+1;

}

}

}

substring(tokens[i],sub\_tk[k],ld,strlen(tokens[i]));

sub\_tk[k++][strlen(tokens[i])-ld]='\0';

for(j=0;j<k;j++)

{

strcpy(word1,test\_token(sub\_tk[j]));

if(hold==1)

{

if(!strcmp(sub\_tk[j],"("))

{

strcpy(word1,"FC");

FC\_flag=1;

hold=0;

}

else

{

strcat(word,"ID ");

hold=0;

}

}

else if(!strcmp(word1,"ID"))

{

hold=1;

}

if(!hold)

{

strcat(word,word1);

if(strcmp(word1,""))

strcat(word," ");

}

if(strcmp(word1,"ID"))

{

id\_flag=0;

}

//printf("%s ",sub\_tk[j]);

}

ld=0;

k=0;

i++;

}

//printf("\n");

if(multi\_flag!=1)

printf("%s\n",word);

else

printf("%s",word);

}

void main()

{

char stmt[128];

FILE \*fp = fopen("test\_file.c", "r");

if(fp == NULL)

{

perror("Unable to open file!");

exit(1);

}

while(fgets(stmt, sizeof(stmt), fp) != NULL)

{

LA(stmt);

strcpy(stmt,"\0");

}

fclose(fp);

}

/\*

MULTI

FC

SP

KW ID ASSIGN NUMCONST SP ID ASSIGN NUMCONST SP

SINGLE

KW SP ID RELOP ID SP

FC

KW

FC

SP

FC

SP

SP

\*/