#include <stdio.h>

#include <malloc.h>

#include <ctype.h>

int compare(char \*a, char \*b)

{

int ct=0;

while(\*(a+ct)!='\0')

{

if(((int)\*(b+ct)-(int)\*(a+ct))!=0) {return 0;}

ct++;

}

return 1;

}

int checkop(char ch1, char ch2)

{

int i=0;

char \*txt = (char\*) malloc(3);

\*txt = ch1; \*(txt+1) = ch2; \*(txt+2) = '\0';

char \*op[] = {"++", "--", "==", ">=", ">>", "<=", "<<", "&&", "||", "!=", "\*\*", "//", "/\*", "\*/", "::"};

while(i<15)

{

if(compare(op[i], txt)) return 1;

i++;

}

return 0;

}

void operate(char \*c, FILE \*wt, int fg)

{

char \*temp = c;

if(compare("#include", temp)) {

while(\*temp!='\0') {

if(\*temp!=' ') fprintf(wt, "%c", \*(temp));

temp++;

}

}

else {

while(\*temp!='\0'){

char ch=\*(temp+1);

if(isalnum(\*temp) || \*temp=='\_' || \*temp=='.' || \*temp=='\"')

{

fprintf(wt, "%c", \*temp);

if(!(isalnum(ch) || ch=='\_' || ch=='.' || ch=='\"')) {

fprintf(wt, "%c", ' ');

}

}

else if(\*temp==',' || \*temp==';' || \*temp==' ') {

fprintf(wt, "%c", ' ');

}

else if(checkop(\*temp,\*(temp+1))) {

fprintf(wt, "%c%c ", \*temp, \*(temp+1));

temp++;

}

else {

fprintf(wt, "%c ", \*temp);

}

temp++;

}

}

}

void tokenize(char \*str, FILE \*ptr)

{

char \*keyset[] = {"auto","break","case","char","const","continue","default","do","double","else","enum","extern","float","for","goto","if","int","long","register","return","printf", "scanf", "short","signed","sizeof","static","struct","switch","typedef","union","unsigned","void","volatile","while"};

char \*opset[] = {"+", "-", "\*", "/", "%", "=", ">", "<", "&", "|", "!", "++", "--", "==", ">=", ">>", "<=", "<<", "&&", "||", "!="};

int i, clss=1;

if(\*str=='#') {

fprintf(ptr, "%s \t --Preprocessor Directive\n", str);

return;

}

if(\*str=='\"') {

fprintf(ptr, "%s \t --Literal\n", str);

return;

}

if(!isalpha(\*(str)) && \*(str)!='\_') {

clss=0;

}

if(clss)

{

for(i=1;\*(str+i)!='\0';i++)

{

if(!isalnum(\*(str+i)) && \*(str+i)!='\_')

{

clss=0;break;

}

}

}

if(clss)

{

for(i=0;i<34;i++)

{

if(compare(keyset[i], str)) {

clss=2; break;

}

}

}

if(!clss)

{

for(i=0;i<21;i++)

{

if(compare(str, opset[i])) {

clss=3; break;

}

}

}

if(!clss)

{

for(i=0;\*(str+i)!='\0';i++) {

if(isdigit(\*(str+i)) || \*(str+i)=='.') {

clss=4;

}

else {

clss=0; break;

}

}

}

if(!clss)

{

if(\*str=='(' || \*str==')' || \*str=='{' || \*str=='}' || \*str=='[' || \*str==']')

{

fprintf(ptr, "%s \t --Separator\n", str);

return;

}

}

switch(clss)

{

case 0: fprintf(ptr, "%s \t --Not Recognized\n", str);

break;

case 1: fprintf(ptr, "%s \t --Identifier\n", str);

break;

case 2: fprintf(ptr, "%s \t --Keyword\n", str);

break;

case 3: fprintf(ptr, "%s \t --Operator\n", str);

break;

case 4: fprintf(ptr, "%s \t --Number\n", str);

}

}

void getwd()

{

FILE \*ptr = fopen("TempStore.txt", "r");

FILE \*wt = fopen("TokenTable.txt", "w");

char \*wd = (char\*) malloc(1);

if(ptr==NULL) {

printf("file not found");

}

while(fscanf(ptr, "%s", wd)!=EOF)

{

tokenize(wd, wt);

}

fclose(ptr);

fclose(wt);

free(wd);

}

int main()

{

FILE \*ptr = fopen("Sample.txt", "r");

FILE \*wt = fopen("TempStore.txt", "w");

char \*wd = (char\*) malloc(1);

int flag = 0;

if(ptr==NULL) {

printf("file not found"); return 0;

}

//while(fscanf(ptr, "%s", wd)!=EOF)

while(fgets(wd, 100, ptr))

{

printf("%s", wd);

operate(wd, wt, flag);

}

fclose(ptr);

fclose(wt);

free(wd);

getwd();

return 0;

}