//Ex -8 Token Seperation using Lexical Analysis

//Aumrudh Lal Kumar TJ

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<ctype.h>

int isspecial(char c){

char spc[10] = {'}','{',',',';','[',']','.','(',')','"'};

int i;

for(i=0;i<10;i++)

{

if(spc[i] == c)

return 1;

}

return 0;

}

int isvar(char str[]){

if(isdigit(str[0]) && str[0] !='\n' && str[0]!=' ')

return 0;

else

return 1;

}

int iskeyword(char str[]){

char kw[10][10] = {"printf","scanf","int","void","main","char","float","include","stdio","double"};

int i;

for(i=0;i<10;i++)

{

if(strcmp(kw[i],str)==0)

return 1;

}

return 0;

}

int istype(char str[]){

char kw[4][10] = {"int","char","float","double"};

int i;

for(i=0;i<4;i++)

{

if(strcmp(kw[i],str)==0)

return 1;

}

return 0;

}

int isoper(char c){

char oper[8]= {'+','-','\*','/','=','>','<','%'};

int i;

for(i=0;i<8;i++)

{

if(oper[i] == c)

return 1;

}

return 0;

}

char \*substr(char str[],int l, int r){

char \*s = (char\*)malloc((r-l+2)\*sizeof(char));

int i,j=0;

for(i=l;i<=r;i++,j++)

s[j]=str[i];

return s;

}

int main(){

FILE \*f = fopen("prog.txt","r");

FILE \*opf,\*kf,\*sf,\*vf,\*fpf;

int rt,lt;

opf = fopen("operators.txt","w");

kf = fopen("keywords.txt","w");

sf = fopen("specialcharacters.txt","w");

vf = fopen("variable.txt","w");

fpf = fopen("formatspecifier.txt","w");

char \*p,\*str,\*fstr;

int i,j,len,f1,f2,vvar=0,dq;

char line[50];

while(1) {

fscanf(f,"%s\n",line);

len = strlen(line);

if(iskeyword(line)==1){

fprintf(kf,"%s\n",line);

//printf("\nKW:%s",line);

continue;

}

lt=0;rt=0;f1=0;f2=0;dq=0;

vvar=1; // to reject var in printf, scanf

if(line[0] == '#')

vvar=0;

while(lt<=rt && rt<=len){

f1=0;f2=0;

if(isspecial(line[rt])==1){

fprintf(sf,"%c\n",line[rt]);

//printf("\nSP:%c",line[rt]);

}

else

f1=1;

if(isoper(line[rt])==1){

fprintf(opf,"%c\n",line[rt]);

//printf("\nOP:%c",line[rt]);

vvar =0;

}

else

f2=1;

if((int)line[rt]==34)

dq = (dq+1)%2;

if(dq==1 && line[rt]=='%'){

fstr = substr(line,rt,rt+1);

fprintf(fpf,"%s\n",fstr);

}

if(strlen(line)==1)

break;

if(isoper(line[rt])==1 || rt ==len || isspecial(line[rt])==1){

if(isalpha(line[lt]))

str = substr(line,lt,rt-1);

else

str = substr(line,lt+1,rt-1);

if(iskeyword(str)==1){

fprintf(kf,"%s\n",str);

//printf("\nKW:%s",str);

if(strcmp(str,"printf")==0 || strcmp(str,"scanf")==0){

//printf("\nNot Var set\n");

vvar=0;

}

if(istype(str)==1)

vvar =1;

}

else if(isvar(str)==1 && vvar==1){

fprintf(vf,"%s\n",str);

//printf("\nVAR:%s",str);

}

lt = rt;

}

rt++;

}

if(feof(f)>0)

break;

}

fclose(f);

fclose(opf);

fclose(kf);

fclose(sf);

fclose(vf);

fclose(fpf);

printf("Seperated All Successfully\n");

return 0;

}

root@AumrudhLalKumarTJ:~/lab/systemsoftware/ex8/final\_version# ./a.out

Seperated All Successfully

root@AumrudhLalKumarTJ:~/lab/systemsoftware/ex8/final\_version# cat formatspecifier.txt

%d

root@AumrudhLalKumarTJ:~/lab/systemsoftware/ex8/final\_version# cat keywords.txt

include

stdio

void

main

int

printf

root@AumrudhLalKumarTJ:~/lab/systemsoftware/ex8/final\_version# cat operators.txt

<

>

=

+

%

root@AumrudhLalKumarTJ:~/lab/systemsoftware/ex8/final\_version# cat specialcharacters.txt

.

(

)

{

,

,

;

;

(

"

"

,

)

;

}

root@AumrudhLalKumarTJ:~/lab/systemsoftware/ex8/final\_version# cat variable.txt

a

b

c

root@AumrudhLalKumarTJ:~/lab/systemsoftware/ex8/final\_version# cat prog.txt

#include<stdio.h>

void main()

{

int a,b,c;

c=a+b;

printf("%d",c);

}.