#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

FILE \*input, \*output;

void enterInputFILE(void)

{

int i;

char inputDir[256];

do{

printf("Vuvedete input faila (C programa koqto SE KOMPILIRA USPE6NO!): ");

scanf("%s", &inputDir);

for(i = 0; inputDir[i] != '\0'; i++);

if(inputDir[i-1] != 'c' || inputDir[i-2] != '.')

{

printf("Faila Trqbva da e s raz6irenie .c \n");

continue;

}

input = fopen(inputDir, "r");

if(input==NULL)

{

printf("Gre6ka pri otvarqne na faila!\n");

}

}while(input == NULL);

}

void enterOutput(void)

{

char outputDir[222];

printf("\nEnter output filelocation: ");

scanf("%s",&outputDir);

output=fopen(outputDir,"a+");

}

void calculateIFELSE(void)

{

int ifCounter = 0, elseCounter = 0;

int i, currnetRowLength, rowsCounter = 0, row;

char currnetRow[1000];

int currentChar, isDoFound = 0;

int check1 = 0, check2 = 0, check3 = 0;

//////////////////////////////

rewind(input);

do{

currentChar=fgetc(input);

if(currentChar == '\n')

{

rowsCounter++;

}

}while(currentChar != EOF);

rowsCounter++;

rewind(input);

for(row = 0; row < rowsCounter; row++)

{

for(i=0;;i++)

{

currentChar=fgetc(input);

if(currentChar == EOF || currentChar == '\n')

{

currnetRow[i] = '\0';

currnetRowLength = i;

break;

}

else

{

currnetRow[i] = currentChar;

}

}

//////////////////////////

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

{

if(currnetRow[i]=='\"')

{

for(;;)

{

i++;

if(currnetRow[i]=='\\')

{

i++;

if(currnetRow[i]=='\"')

{

continue;

}

}

if(currnetRow[i]=='\"') break;

if(currnetRow[i]=='\0') break;

}

}

if( (check1 == 1) && (currnetRow[i] == '\*'))

{

for(;;)

{

i++;

if(currnetRow[i]=='\*')

{

check3=1;

continue;

}

if((check3==1) && (currnetRow[i]=='/'))

{

check1 = 0;

check2 = 0;

check3 = 0;

break;

}

else

{

check3=0;

}

if(currnetRow[i]=='\0')

{

//read new line and continue

for(i=0;;i++)

{

currentChar=fgetc(input);

if(currentChar == EOF || currentChar == '\n')

{

currnetRow[i] = '\0';

currnetRowLength = i;

break;

}

else

{

currnetRow[i] = currentChar;

}

}

i=0;

continue;

}

}

continue;

}

else

{

check1 = 0;

}

if( (check2 == 1) && (currnetRow[i] == '/'))

{

check1 = 0;

check2 = 0;

do{

i++;

}while(currnetRow[i]!='\0');

if(currnetRow[i]=='\0')

{

break;

}

else

{

continue;

}

}

else

{

check2 = 0;

}

if(currnetRow[i] == '/')

{

if(i < currnetRowLength - 1)

{

if(currnetRow[i + 1] == '/' || currnetRow[i + 1] == '\*')

{

check1 = 1;

check2 = 1;

continue;

}

}

}

if(currnetRow[i] == 'i')

{

if( i > 0)

{

if(!(currnetRow[i-1] == ' ' || currnetRow[i-1] == '\t' || currnetRow[i-1] == '\v' || currnetRow[i-1] == '\n'))

{

continue;

}

}

if(i < currnetRowLength - 1)

{

i++;

if(currnetRow[i] == 'f')

{

if(i < currnetRowLength - 1)

{

i++;

if(currnetRow[i] == '(' || currnetRow[i] == ' ' || currnetRow[i] == '\t' || currnetRow[i] == '\v' || currnetRow[i] == '\n')

{

ifCounter++;

}

}

else

{

ifCounter++;

}

}

}

}

///////

if(currnetRow[i] == 'e')

{

if( i > 0)

{

if(!(currnetRow[i-1] == ' ' || currnetRow[i-1] == '\t' || currnetRow[i-1] == '\v' || currnetRow[i-1] == '\n'))

{

continue;

}

}

if(i < currnetRowLength - 1)

{

i++;

if(currnetRow[i] == 'l')

{

if(i < currnetRowLength - 1)

{

i++;

if(currnetRow[i] == 's')

{

if(i < currnetRowLength - 1)

{

i++;

if(currnetRow[i] == 'e')

{

if(i < currnetRowLength - 1)

{

i++;

if(currnetRow[i] == '(' || currnetRow[i] == ' ' || currnetRow[i] == '\t' || currnetRow[i] == '\v' || currnetRow[i] == '\n')

{

elseCounter++;

}

}

else

{

elseCounter++;

}

}

}

}

}

}

}

}

}

}

printf("if - %d\nelse - %d \n",ifCounter, elseCounter);

fprintf(output, "if - %d\nelse - %d \n",ifCounter, elseCounter);

}

void func3(void)

{

int currentChar, prevChar = EOF;

int pogolqmo = 0, pomalko = 0, ravno = 0;

int check1 = 0, check2 = 0, check3 = 0;

rewind(input);

while((currentChar=fgetc(input))!=EOF)

{

if(currentChar=='\"')

{

for(;;)

{

currentChar=fgetc(input);

if(currentChar=='\\')

{

currentChar=fgetc(input);

if(currentChar=='\"')

{

continue;

}

}

else if(currentChar=='\"')

{

break;

}

}

prevChar = currentChar;

continue;

}

if( (check1 == 1) && (currentChar == '\*'))

{

for(;;)

{

currentChar=fgetc(input);

if(currentChar=='\*')

{

check3=1;

continue;

}

if((check3==1) && (currentChar=='/'))

{

check1 = 0;

check2 = 0;

check3 = 0;

break;

}

else

{

check3=0;

}

if(currentChar==EOF)

{

break;

}

}

}

else

{

check1 = 0;

}

if( (check2 == 1) && (currentChar == '/'))

{

check1 = 0;

check2 = 0;

do{

currentChar=fgetc(input);

if(currentChar==EOF)

{

break;

}

}while(currentChar!='\n');

if(currentChar==EOF)

{

break;

}

}

else

{

check2 = 0;

}

if(currentChar == '/')

{

check1 = 1;

check2 = 1;

}

if(currentChar == '>')

{

currentChar = fgetc(input);

if(currentChar != '=')

{

pomalko++;

}

}

else if(currentChar == '<')

{

currentChar = fgetc(input);

if(currentChar != '=')

{

pomalko++;

}

}

else if(currentChar == '=')

{

currentChar = fgetc(input);

if(currentChar == '=')

{

if(prevChar != EOF)

{

if(prevChar != '!' && prevChar != '>' && prevChar != '<')

{

ravno++;

}

}

}

}

prevChar = currentChar;

}

printf("Broq na \n> %d \n< %d \n== %d\n", pogolqmo, pomalko, ravno);

fprintf(output, "Broq na: \n> %d \n< %d \n== %d\n", pogolqmo, pomalko, ravno);

}

void function4(void)

{

int currentChar, comentCounter = 0;

int check1 = 0, check2 = 0, check3 = 0;

rewind(input);

while((currentChar=fgetc(input))!=EOF)

{

if(currentChar=='\"')

{

for(;;)

{

currentChar=fgetc(input);

if(currentChar=='\\')

{

currentChar=fgetc(input);

if(currentChar=='\"')

{

continue;

}

}

else if(currentChar=='\"')

{

break;

}

}

continue;

}

if( (check1 == 1) && (currentChar == '\*'))

{

for(;;)

{

currentChar=fgetc(input);

if(currentChar=='\*')

{

check3=1;

continue;

}

if((check3==1) && (currentChar=='/'))

{

check1 = 0;

check2 = 0;

check3 = 0;

break;

}

else

{

check3=0;

}

if(currentChar==EOF)

{

break;

}

}

}

else

{

check1 = 0;

}

if( (check2 == 1) && (currentChar == '/'))

{

comentCounter++;

check1 = 0;

check2 = 0;

do{

currentChar=fgetc(input);

if(currentChar==EOF)

{

break;

}

}while(currentChar!='\n');

if(currentChar==EOF)

{

break;

}

}

else

{

check2 = 0;

}

if(currentChar == '/')

{

check1 = 1;

check2 = 1;

}

}

printf("Broq na redovete sudurja6ti komentar pt tipa // sa %d.\n", comentCounter);

fprintf(output, "Broq na redovete sudurja6ti komentar pt tipa // sa %d.\n", comentCounter);

}

int main(void)

{

int choice, firstCheck = 0;

do{

system("cls");

printf("1. Izbor na fail za obrabotka.\n");

printf("2. Broq na if-ovete i broq na else-ovete.\n");

printf("3. Broi <, >, ==.\n");

printf("4. Broi na komentarite ot tipa //.\n");

printf("0. Izxod.\n");

printf("Izberete opciq ot slednoto menu: ");

fflush(stdin);

scanf("%d", &choice);

switch(choice)

{

case 1:

enterInputFILE();

enterOutput();

firstCheck = 1;

break;

case 2:

if(firstCheck == 1)

{

calculateIFELSE();

}

else

{

printf("Purvo prez opciq 1!!\n");

}

system("Pause");

break;

case 3:

if(firstCheck == 1)

{

func3();

}

else

{

printf("Purvo prez opciq 1!!\n");

}

system("Pause");

break;

case 4:

if(firstCheck == 1)

{

function4();

}

else

{

printf("Purvo prez opciq 1!!\n");

}

system("Pause");

break;

case 0:

break;

default:

printf("Nevalidna opciq\n");

system("Pause");

break;

}

}while(choice != 0);

system("Pause");

}