|  |  |
| --- | --- |
| Muhammad Hamza | SP17-BCS-114 |

Assignment 2 - Analyzer

Compiler : gcc

Environment: linux

Cpp version : 11

|  |
| --- |
| **output** |
|  |
| **Input file (testing.txt)** |
| @a = 123433^112  @a = 123433.112  @aa = 123433  IF (@m || @y)  {  @m  @m= 1  123433.  123433^  CONST  IF  @a ++  @a += 5  //asasda  {}  []  <>  <<<  >>>  <  /\*asdasd\*asdasd\*/  > := -- -= (&& ||)  }  ELSE {  @a += 1  }  SWITCH ( @a ) {  asdasdasd  ascklascnkl  While  } |
| **Cpp code** |
| **#include<stdio.h>**  **#include<stdlib.h>**  **#include <fstream>**  **#include <iostream>**  **#include <string.h>**  **using namespace std;**  **enum COL\_NAME**  **{**  **A=128,R,T**  **};**  **enum TOKENS**  **{**  **IDENTIFIER,RESERVE,PUNCTUATION,INPUT,OUTPUT,OPERATOR,NUMERICAL,COMMENT**  **};**  **int\*\* getTT(){**  **ifstream obj;**  **obj.open("/home/mehroz/repo/cLang/compiler\_analyser/TT3.csv");**  **string line;**  **int\*\* arr =new int\*[70];**  **int i=0,j=0;**  **//skiping header**  **getline(obj, line);**  **//after header**  **while(i<70){**  **arr[i] = new int[131];**  **getline(obj, line);**  **char \*dup = strdup(line.c\_str());**  **char \*p = strtok(dup, ",");**  **//skip first col**  **p = strtok(NULL, ",");**  **//start reading**  **while (p) {**  **if(strcmp(p,"IDENTIFIER")==0){**  **arr[i][j++]=IDENTIFIER;**  **}else if(strcmp(p,"RESERVE")==0){**  **arr[i][j++]=RESERVE;**  **}else if(strcmp(p,"PUNCTUATION")==0){**  **arr[i][j++]=PUNCTUATION;**  **}else if(strcmp(p,"INPUT")==0){**  **arr[i][j++]=INPUT;**  **}else if(strcmp(p,"OUTPUT")==0){**  **arr[i][j++]=OUTPUT;**  **}else if(strcmp(p,"OPERATOR")==0){**  **arr[i][j++]=OPERATOR;**  **}else if(strcmp(p,"NUMERICAL")==0){**  **arr[i][j++]=NUMERICAL;**  **}else if(strcmp(p,"COMMENT")==0){**  **arr[i][j++]=COMMENT;**  **}else{**  **arr[i][j++]=atoi(p);**  **}**  **p = strtok(NULL, ",");**  **}**  **j=0;i++;**  **arr[i] = new int[128];**  **}**  **return arr;**  **}**  **int main(){**  **string t[] = {"IDENTIFIER","RESERVE","PUNCTUATION","INPUT","OUTPUT","OPERATOR","NUMERICAL","COMMENT"},error,tokens;**  **char b1[4096], b2[4096],lex[100];**  **int \*\* TT = getTT();**  **ifstream obj;**  **obj.open("/home/mehroz/repo/cLang/compiler\_analyser/testing.txt");**  **char \* curr\_buff = b1;**  **obj.get(curr\_buff,4096, EOF);**  **int count=0, c=0,state=0, i=0, j=0;**  **while(curr\_buff[count]!='\0'){**  **if(!TT[state][A]){**  **state=TT[state][(int)curr\_buff[count]];**  **if(state==-1){**  **error+= curr\_buff[count];**  **j=0;state=0;count++;**  **}else{**  **lex[j++]=curr\_buff[count++];**  **}**  **}**  **if(TT[state][A]){**  **lex[TT[state][R]==1?j-1:j]='\0';**  **// cout<<lex<<" " << t[TT[state][T]]<<"\n";**  **tokens.append(lex);**  **tokens.append(" ");**  **tokens.append( t[TT[state][T]]);**  **tokens.append( "\n");**  **j=0;state=0;**  **if(TT[state][R])**  **count--;**  **}**  **if(count>4095){**  **if(curr\_buff==b1){**  **curr\_buff = b2;**  **}else**  **curr\_buff = b1;**  **obj.get(curr\_buff,4096, EOF);**  **count=0;**  **}**  **}**  **cout<<"\n\n-----------tokens----------\n"<<tokens;**  **ofstream token("/home/mehroz/repo/cLang/compiler\_analyser/tokens.txt");**  **token << tokens;**  **token.close();**  **cout<<"\n\n-----------ERRORS----------\n"<<error;**  **ofstream errors("/home/mehroz/repo/cLang/compiler\_analyser/errors.txt");**  **errors << error;**  **errors.close();**  **return 0;**  **}** |
| **Errors.txt** |
| asdasdasd  ascklascnkl  While |
| **tokens.txt** |
| @a IDENTIFIER  123433^112 NUMERICAL  @a IDENTIFIER  123433.112 NUMERICAL  @aa IDENTIFIER  123433 NUMERICAL  IF RESERVE  ( PUNCTUATION  @m IDENTIFIER  || OPERATOR  @y IDENTIFIER  { PUNCTUATION  @m IDENTIFIER  @m IDENTIFIER  1 NUMERICAL  123433**.** NUMERICAL  123433**^** NUMERICAL  CONSTRESERVE  IF RESERVE  @a IDENTIFIER  ++ OPERATOR  @a IDENTIFIER  += OPERATOR  5 NUMERICAL  //asasda  COMMENT  { PUNCTUATION  } PUNCTUATION  [ PUNCTUATION  ] PUNCTUATION  <> OPERATOR  <<< OUTPUT  >>> INPUT  < OPERATOR  /\*asdasd\*asdasd\*/ COMMENT  > OPERATOR  := OPERATOR  -- OPERATOR  -= OPERATOR  ( PUNCTUATION  && OPERATOR  || OPERATOR  ) PUNCTUATION  } PUNCTUATION  ELSE RESERVE  { PUNCTUATION  @a IDENTIFIER  += OPERATOR  1 NUMERICAL  } PUNCTUATION  SWITCH RESERVE  ( PUNCTUATION  @a IDENTIFIER  ) PUNCTUATION  { PUNCTUATION  } PUNCTUATION |