—-Final Project—- Program report

CSE-0302 Summer 2021

Arafat mia
Department of Computer Science and Engineering
State University of Bangladesh (SUB)
Dhaka, Bangladesh
ID :- UG02-50-19-011

***Abstract*—**

I. Assignment no :- 4

make a text file and and present line number in c program
,

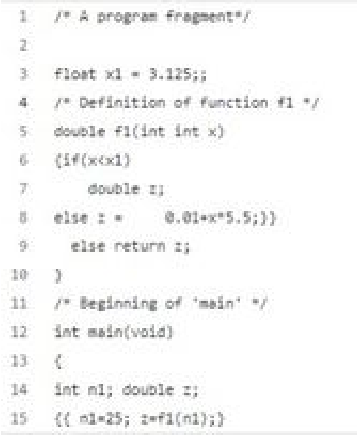
II. Program Algorithm :-

I have written a C++ program that opens a text file and compares the given string with the string present in the file. I'm trying to print the line number in which the same string occurs, but I am unable to get the proper output: output does print the correct line number.

III. Conclusion

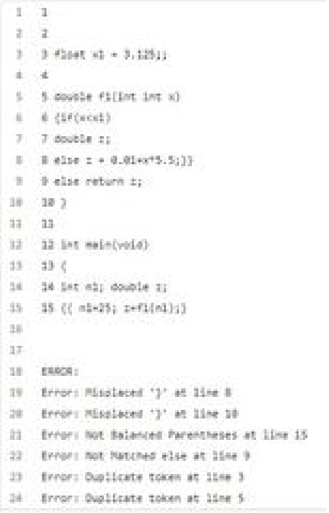
c++ the code has been success run

IV. Picture



```
1   /* A program fragment*/
2
3   float x1 = 3.125;;
4   /* Definition of function f1 */
5   double f1(int int x)
6   {if(x<x1)
7       double z;
8   else z =     0.01+x*5.5;}}
9     else return z;
10  }
11  /* Beginning of 'main' */
12  int main(void)
13  {
14  int n1; double z;
15  {{ n1=25; z=f1(n1);}
```

Fig. 1. 4 no code input



```
1
2
3 float x1 = 3.125;;
4
5 double f1(int int x)
6 {if(x<x1)
7 double z;
8 else z = 0.01+x*5.5;}}
9 else return z;
10 }
11
12 int main(void)
13 {
14 int n1; double z;
15 {{ n1=25; z=f1(n1);}


ERROR:
Error: Misplaced '}' at line 8
Error: Misplaced '}' at line 10
Error: Not Balanced Parentheses at line 15
Error: Not Matched else at line 9
Error: Duplicate token at line 3
Error: Duplicate token at line 5
```

Fig. 2. 4 no code output

V. Assignment no :- 5

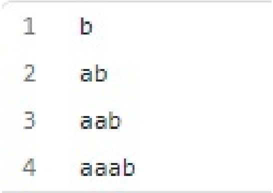
A parser usually checks all data provided to ensure it is sufficient to build a data structure in the form of a parse tree or an abstract syntax tree.

VI. Program Algorithm :-

A parser is a compiler or interpreter component that breaks data into smaller elements for easy translation into another language. A parser takes input in the form of a sequence of tokens, interactive commands, or program instructions and breaks them up into parts that can be used by other components in programming

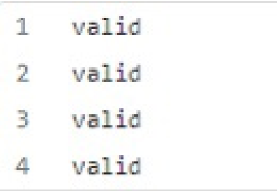
VII. Conclusion

c++ the code has been success run



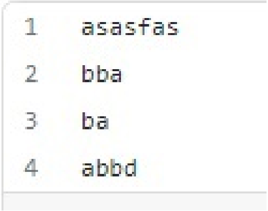
```
b
ab
aab
aaab
```

Fig. 3. 5.1 no code input



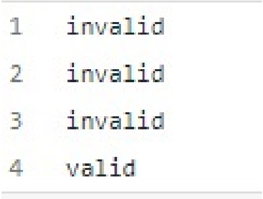
```
valid
valid
valid
valid
```

Fig. 4. 5.1 no code output



```
asasfas
bba
ba
abbd
```

Fig. 5. 5.2 no code input



```
invalid
invalid
invalid
valid
```

Fig. 6. 5.2 no code output

VIII. PICTURE

IX. ASSIGNMENT NO :- 6

Removing ambiguity from the program lines .

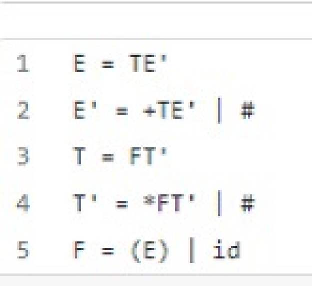
X. PROGRAM ALGORITHM :-

Predictive parsing is a special form of recursive descent parsing, where no backtracking is required, so this can predict which products to use to replace the input string. ... The main problem during predictive parsing is that of determining the production to be applied for a non-terminal

XI. CONCLUSION

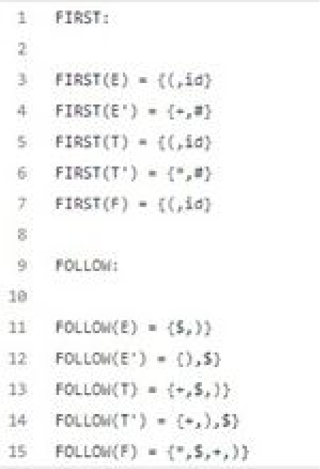
c++ the code has been success run

XII. PICTURE



```
E = TE'
E' = +TE' | #
T = FT'
T' = *FT' | #
F = (E) | id
```

Fig. 7. 6 no code input



```
FIRST:

FIRST(E) = {(,id}
FIRST(E') = {+,#}
FIRST(T) = {(,id}
FIRST(T') = {*,#}
FIRST(F) = {(,id}

FOLLOW:

FOLLOW(E) = {$,)}
FOLLOW(E') = {),$}
FOLLOW(T) = {+,$,)}
FOLLOW(T') = {+,),$}
FOLLOW(F) = {*,$,+,)}
```

Fig. 8. 6 no code output