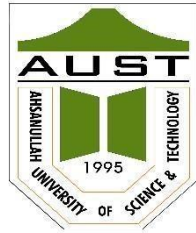


AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY  
DHAKA-1208, BANGLADESH.



Department of Computer Science and Engineering  
Spring 2019

Program: Bachelor of Science in Computer Science and Engineering

Course No: CSE 4130

Course Title: Formal Languages and Compilers Lab

Session: Spring 2019

Assignment No: 02

Date of Submission: 07/08/2019

Submitted to      Md. Aminur Rahman

Lecturer, Department of CSE, AUST.

Shoeb Mohammad Shahriar

Lecturer, Department of CSE, AUST.

Submitted By

Name            : Robiul Hasan Nowshad

Student ID   : 16.01.04.061

Lab Group   : B1

## I. Question

Suppose, we have a C source program scanned and filtered as it was done in Session 1. We now take that modified file as input, and separate the lexemes first. We further recognize and mark the lexemes as different types of tokens like keywords, identifiers, operators, separators, parenthesis, numbers, etc.

## II. Input File

scanned.c

```
char c; int x1, x_2; float y1, y2; x1=5; x_2= 10; y1=2.5+x1*45; y2=100.o5-x_2/3; if (y1<=y2) c='y'; else c='n';
```

## III. Source Code

Assmnt2\_160104061.c

```
1. #include <stdio.h>
2. #include <string.h>
3. #include <ctype.h>
4.
5. int lexemes_Separation();
6. int check_keyword(char lexeme[]);
7. int checkId(char lexeme[]);
8. int check_operator(char lexeme[]);
9. int check_number(char lexeme[20]);
10. int check_bracket(char lexeme[]);
11. int check_separator(char lexeme[]);
12.
13. FILE *f1,*f2,*f3;
```

### Main Function

```
14. int main()
15. {
16.     char c;
17.     f1 = fopen("scanned.txt", "r");
18.     f2 = fopen("lexemes.txt", "w");
19.
20.     lexemes_Separation();
21.
22.     fclose(f1);
23.     fclose(f2);
24.
25.     printf("Input File:\n\n");
26.     f1 = fopen("scanned.txt", "r");
```

```

27. while((c=fgetc(f1))!=EOF)
28. {
29.     printf("%c",c);
30. }
31. printf("\n\n\n");
32. f1 = fopen("lexemes.txt", "r");
33. printf("Lexemes Separated by Space: \n\n");
34. while((c=fgetc(f1))!=EOF)
35. {
36.     printf("%c",c);
37. }
38. printf("\n\n\n");
39. f1 = fopen("lexemes.txt", "r");
40. f2 = fopen("tokens.txt", "w");
41. char str[50];
42. while(fscanf(f1, "%s", &str)!=EOF)
43. {
44.     if(check_keyword(str))
45.         fprintf(f2, "[kw %s] ", str);
46.     else if(check_operator(str))
47.         fprintf(f2, "[op %s] ", str);
48.     else if(check_bracket(str))
49.         fprintf(f2, "[par %s] ", str);
50.     else if(check_separator(str))
51.         fprintf(f2, "[sep %s] ", str);
52.     else if(checkId(str))
53.         fprintf(f2, "[id %s] ", str);
54.     else if(check_number(str))
55.         fprintf(f2, "[num %s]", str);
56.     else
57.     {
58.         fprintf(f2, "[unkn %s]", str);
59.         f3 =fopen("Error.txt","w");
60.         fprintf(f3, "Error : unknown %s", str);
61.         fclose(f3);
62.     }
63. }
64. fclose(f1);
65. fclose(f2);
66. printf("Final Result: \n\n");
67. f1 = fopen("tokens.txt", "r");
68. while((c=fgetc(f1))!=EOF)
69. {
70.     printf("%c",c);
71. }
72. printf("\n\n\n");
73. f3 =fopen("Error.txt","r");
74. while((c=fgetc(f3))!=EOF)
75. {
76.     printf("%c",c);
77. }
78. fclose(f3);
79. printf("\n\n");
80. return 0;

```

```
81. }
```

## Lexemes Separation

```
82. int lexemes_Separation()
83. {
84.     char c;
85.     while((c=fgetc(f1))!=EOF)
86.     {
87.         if(!isalnum(c) && c!=' ' && c!='_' && c!='.')
88.             fputc(' ', f2);
89.         fputc(c, f2);
90.         if(c=='>' || c=='<' || c=='=' || c=='!')
91.         {
92.             char ch;
93.             if((ch=fgetc(f1))=='=')
94.             {
95.                 fputc(ch, f2);
96.                 fputc(' ', f2);
97.             }
98.             else
99.             {
100.                 fputc(' ', f2);
101.                 fputc(ch, f2);
102.                 if(!isalnum(ch) && ch!=' ' && ch!='_' && ch!='.')
103.                     fputc(' ', f2);
104.             }
105.         }
106.         else if(!isalnum(c) && c!=' ' && c!='_' && c!='.')
107.             fputc(' ', f2);
108.     }
109.     return 0;
110. }
111.
112. int check_keyword(char lexeme[])
113. {
114.     int i, s=0;
115.     char keywords[][15] = {"int", "float", "double", "char", "return", "if",
116. "else", "else if", "while", "for", "case", "do while", "goto"};
117.     for(i=0; i<12; i++)
118.         if(!strcmp(lexeme, keywords[i]))
119.         {
120.             s=1;
121.             break;
122.         }
123.     return s;
124. }
```

## Find Identifier

```
124. int checkId(char lexeme[])
125. {
126.     int i, s=0, l;
127.     l=strlen(lexeme);
```

```

128.         if((isalpha(lexeme[0])) || (lexeme[0]=='_'))
129.             s=1;
130.         if(s==1)
131.         {
132.             for(i=1; i<l; i++)
133.             {
134.                 if(!isalnum(lexeme[i]) && lexeme[i]!='_'&& !isalpha(lexeme
[i]))
135.                 {
136.                     s=0;
137.                     break;
138.                 }
139.             }
140.         }
141.         return s;
142.     }

```

### Find Operator

```

143.     int check_operator(char lexeme[])
144.     {
145.         int i, l, s=0;
146.         l=strlen(lexeme);
147.         if(l==1 && (lexeme[0]=='>' || lexeme[0]=='+' || lexeme[0]=='-'
|| lexeme[0]=='*' || lexeme[0]=='/' || lexeme[0]=='<' || lexeme[0]=='='))
148.             return 1;
149.         else if(l==2 && (lexeme[0]=='>' || lexeme[0]=='<' || lexeme[0]=='='
|| lexeme[0]=='!') && lexeme[1]=='=')
150.             s=1;
151.         else s=0;
152.         return s;
153.     }

```

### Find Number

```

154.     int check_number(char lexeme[20])
155.     {
156.         int i, l, s;
157.         i=0;
158.         if(isdigit(lexeme[i]))
159.         {
160.             {
161.                 s=1;
162.                 i++;
163.             }
164.         else if(lexeme[i]=='.')
165.         {
166.             s=2;
167.             i++;
168.         }
169.         else s=0;
170.         l=strlen(lexeme);
171.         if(s==1)

```

```

172.         for(; i<l; i++)
173.         {
174.             if(isdigit(lexeme[i]))
175.                 s=1;
176.             else if(lexeme[i]=='.')
177.             {
178.                 s=2;
179.                 i++;
180.                 break;
181.             }
182.             else
183.             {
184.                 s=0;
185.                 break;
186.             }
187.         }
188.         if(s==2)
189.             if(isdigit(lexeme[i]))
190.             {
191.                 s=3;
192.                 i++;
193.             }
194.             else s=0;
195.         if(s==3)
196.             for(; i<l; i++)
197.             {
198.                 if(isdigit(lexeme[i]))
199.                     s=3;
200.                 else
201.                 {
202.                     s=0;
203.                     break;
204.                 }
205.             }
206.         if(s==3) s=1;
207.         return s;
208.     }

```

## Find Bracket

```

209.     int check_bracket(char lexeme[])
210.     {
211.         int l = strlen(lexeme);
212.         return (l==1 && (lexeme[0]=='(' || lexeme[0]==')' || lexeme[0]=='{'
213.             || lexeme[0]=='}' || lexeme[0]=='[' || lexeme[0]==']'));
214.     }
215.     int check_separator(char lexeme[])
216.     {
217.         int i, l;
218.         l=strlen(lexeme);
219.         if(l==1 && (lexeme[0]==',' || lexeme[0]==39 || lexeme[0]==';' || 1
220.             exeme[0]=='"))
221.             return 1;

```

```

221.         else
222.         return 0;
223.     }

```

## Output:

```

G:\AUST4.1\CompilerLab\lab2\Assignment2_160104061.exe
Input File:
char c; int x1, x_2; float y1, y2; x1=5; x_2= 10; y1=2.5+x1*45; y2=100.o5-x_2/3; if (y1<=y2) c='y'; e
lse c='n';

Lexemes Separated by Space:
char c ; int x1 , x_2 ; float y1 , y2 ; x1 = 5 ; x_2 = 10 ; y1 = 2.5 + x1 * 45 ; y2 = 100.o5
- x_2 / 3 ; if ( y1 <= y2 ) c = ' y ' ; else c = ' n ' ;

Final Result:
[kw char] [id c] [sep ;] [kw int] [id x1] [sep ,] [id x_2] [sep ;] [id float] [id y1] [sep ,] [id y2]
[sep ;] [id x1] [op =] [num 5][sep ;] [id x_2] [op =] [num 10][sep ;] [id y1] [op =] [num 2.5][op +] [
id x1] [op *] [num 45][sep ;] [id y2] [op =] [unkn 100.o5][op -] [id x_2] [op /] [num 3][sep ;] [kw if]
[par (] [id y1] [op <=] [id y2] [par )] [id c] [op =] [sep '] [id y] [sep '] [sep ;] [kw else] [id c]
[op =] [sep '] [id n] [sep '] [sep ;]

Error : unknown 100.o5

Process returned 0 (0x0) execution time : 0.095 s
Press any key to continue.

```