

Ahsanullah University of Science & Technology

Department of Computer Science & Engineering



CSE 4130

Formal Languages & Compilers Lab

Session: 01

Assignment: 01

Submitted By:

Name: Syed Sanzam

ID: 16.01.04.042

Lab Group: A2

Date of Submission: **July 16, 2019**

Question: A C source program with single and multiple line comments is given. As the first step toward compilation you need to remove the comments and white space (extra spaces, tabs and newline characters). Develop a program that takes as input file the given source program and produces a filtered file as stated above. The program must also display both the files.

Input File:

```
1. #include<stdio.h>
2. int main(void)
3. {
4. // Single Line Comment
5. printf ("Hello");
6. /* Multi
7. Line
8.
9. Comment
10.
11. */
12. printf("World");
13. return 0;
14. }
```

Source:

```
1. /**
2.  * Author:    Syed Sanzam
3.  * Subject:   File Trimming
4.  * Created:   7.14.2019
5.  *
6.  */
7.
8. #include <stdio.h>
9. #include <stdlib.h>
10.
11. int main()
12. {
13.     FILE *fpRead;
14.     FILE *fpWrite;
15.
16.     fpRead = fopen("input1_160104042.c","r");
17.     fpWrite = fopen("output.txt","w");
18.
19.     char c;
20.
21.     while( (c = fgetc(fpRead)) != EOF)
22.     {
23.
24.         if(c == '\n')
25.         {
26.             fputc(' ',fpWrite);
27.         }
28.
29.     }
```

```

30.         //Single Line Comment Check
31.     else if(c == '/')
32.     {
33.         char d;
34.         char e;
35.         char f;
36.
37.         d = fgetc(fpRead);
38.         if(d == '/')
39.         {
40.             while( (e = fgetc(fpRead)) != EOF)
41.             {
42.                 if(e == '\n')
43.                     break;
44.             }
45.
46.         }
47.
48.         //Block Comment Check
49.         if(d == '*')
50.         {
51.             while( (e = fgetc(fpRead)) != EOF)
52.             {
53.                 if(e == '/')
54.                 {
55.                     f = fgetc(fpRead);
56.                     break;
57.                 }
58.             }
59.         }
60.     }
61.     //Whitespace Check
62.     else if(c == ' ')
63.     {
64.         //If Character after space is a Parenthesis
65.         char b;
66.
67.         //Spaces found after a 'singular space'
68.         char s;
69.
70.         b = fgetc(fpRead);
71.
72.         if(b != '(' && b!= ' ')
73.         {
74.             fputc(c,fpWrite);
75.         }
76.
77.         fputc(b,fpWrite);
78.         while ((s = fgetc(fpRead)) != EOF)
79.         {
80.             if(s != ' ')
81.                 break;
82.         }
83.         fputc(s,fpWrite);
84.     }
85.
86.     else
87.     {
88.
89.         //Newline & Tab Check
90.         if(c != '\t' && c != '\n')
91.             fputc(c,fpWrite);
92.

```

```

93.     }
94. }
95.
96. fclose(fpRead);
97. fclose(fpWrite);
98.
99.
100.
101.
102.     //Print Both Files on the Console
103.
104.     FILE *fOut1;
105.     FILE *fOut2;
106.
107.     fOut1 = fopen("input1_160104042.c", "r");
108.     fOut2 = fopen("output.txt", "r");
109.
110.     char x;
111.     char y;
112.
113.     printf("\nContents of input.c\n");
114.     while( (x = fgetc(fOut1)) != EOF)
115.     {
116.         printf("%c",x);
117.     }
118.
119.     printf("\n\nContents of output.txt\n");
120.
121.     while( (y = fgetc(fOut2)) != EOF)
122.     {
123.         printf("%c",y);
124.     }
125.
126.     printf("\n\n");
127.
128.     fclose(fOut1);
129.     fclose(fOut2);
130.
131.
132.     return 0;
133.
134.
135. }
```

Output File:

```

1. #include<stdio.h> int main(void) { printf("Hello"); printf("World"); return 0; }
```