

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  FILE *fp , *fp2;
5
6  void check_comment(char a)
7  {
8      char x;
9
10     if( a == '/' )    //checking if the character starts with '/', it will be a comment
11     {
12         if((x=fgetc(fp))== '*')
13             check_block_comment();
14
15         else if( x == '/' )    // else if the next character '/', it is the beginning of single line comment
16         {
17             check_single_comment();
18
19         }
20         else
21         {
22             // when both the cases fail then it is not a comment
23             fputc(a,fp2);
24             fputc(x,fp2);
25         }
26     }
27
28     // when all the conditions are false, add the character as it is in the new file.
29     else
30         fputc(a,fp2);
31 }
32
33 // function for block comments
34 void check_block_comment()
35 {
36
37     char x,y;
38
39     while((x=fgetc(fp))!=EOF)    // the block comment has started
40     {
41
42         if(x=='*')
43         {
44             y=fgetc(fp);    // check if it ends
45
46             if(y=='/')
47                 return;
48         }
49     }
50 }
51
52
53
54 // function for single line comments
55 void check_single_comment()
56 {
57     char x,y;
58
59     while((x=fgetc(fp))!=EOF)
60     {
61
62         if(x=='\n')
63             return;    // if the comment ends return from the function
64
65     }
66 }

```

```
67
68 }
69
70
71 int main(void)
72 {
73     char c;
74
75     fp = fopen ("testfile.txt","r") ;    // first file in read mode
76     fp2 = fopen ("solved.txt","w") ;    // second file in write mode
77
78     while(
79         (c=fgetc(fp))!=EOF)
80         check_comment(c);    // checking for the beginning of a comment
81
82     // closing both files
83     fclose(fp);
84     fclose(fp2);
85
86     return 0;
87 }
88
```