Assignment No:1

Course Name: Formal Languages and Compilers Lab

ID: 170104082

Group: B2

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

Ans:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>

int provideOneSpace =0;

void writeOutputFile (char c , FILE *P2)
{
   fputc(c, P2);
}
```

```
int checkSlash(FILE *P1)
{
  char check = fgetc(P1);
  if (check == '/')
  {
    return 1;
  }
  else if (check == '*')
  {
    return 2;
  }
  else
  {
    return 3;
  }
}
void removeMultilineComment(FILE *P1)
{
  Char c;
  while((c=fgetc(P1)) != EOF)
  {
    if (c=='*')
    {
      if ((c=fgetc(P1))=='/') return;
    }
  }
}
```

```
void removeSingleLineComment(FILE *p1)
{
    char c;
    while ((c = fgetc(p1)) != '\n')
    {
            continue;
    }
}

void removeExtraSpacesNewlines(FILE *p2)
{
    if (provideOneSpace) == 0)
    {
      writeOutputFile(' ', p2);
     provideOneSpace = 1;
    }
}

void readInputFile()
{
    FILE *p1 = fopen("input.txt", "r");
    char c;
    if (! p1)
    {
      printf(" File can't be opened");
    }
    else
    {
      while ((c = fgetc(p1)) != EOF)
      { printf("%c", c);
      }
    }
    fclose(P1);
}

void readOutputFile ()
{
    FILE * P1 = fopen("output.txt", "r");
    char c;
    if (! P1)
    {
        printf("File can't be opened");
    }
    else
    {
        while ((c=fgetc(P1)) != EOF)
        {
            printf("%c",c);
        }
    }
    fclose(P1);
}
```

```
int main(void)
{
 FILE *P1 , *P2;
 Char c;
 P1 = fopen("input.txt", "r");
 P2 = fopen("output.txt", "w");
 if(!P1 || ! P2)
 {
    printf("File can't be opened");

 }
 else
 {
    while((c=fgetc(P1)) != EOF)
    {
       if(c!=' ' && c!='\n')
       {
          if(c=='/')
          {
           int check = checkSlash(P1);
           if(check==1)
           {
              removeSingleLineComment(P1);
           }
           else if(check==2)
           {
             removeMultiLineComment(P1);
           }
```

```
            else
            {
              provideOnespace=0;
              writeOutputfile(c, p2);
            }
        }
        else
        {
            provideOneSpace = 0
            writeOutputfile(c, p2);
        }
        else if (c==' ' || c=='\n')
        {
            removeExtraSpacesNewlines(p2);
        }
      }
   }
   fclose(p1)
   fclose(p2);
   readInputFile();
   readOutputFile();
}
```