Compiler Design Lab – 2

Dipesh Singh – 190905520

Solved Exercise : Removal of single and multiline comments

Source Code :

```
#include <stdio.h>

int main(){
     FILE *fa, *fb;
     int ca, cb;
     fa = fopen("q4in.c", "r");
     if (fa == NULL){
          printf("Cannot open file \n");
          exit(0);
     }
     fb = fopen("q4out.c", "w");
     ca = getc(fa);
     while (ca != EOF){
          if(ca==' '){
               putc(ca,fb);
               while(ca==' ')
                    ca = getc(fa);
          }
          if (ca=='/'){
               cb = getc(fa);
               if (cb == '/'){
                    while(ca != '\n')
                    ca = getc(fa);
               }
               else if (cb == '*'){
                    do{
                         while(ca != '*')
                              ca = getc(fa);
                         ca = getc(fa);
                    } while (ca != '/');
               }
               else{
                    putc(ca,fb);
                    putc(cb,fb);
               }
          }
          else putc(ca,fb);
          ca = getc(fa);
     }
     fclose(fa);
     fclose(fb);
     return 0;
}
```

Output :

```
ugcse@prg28:~/Desktop/190905520/lab2$ cat q4in.c
// This is a single line comment
/* *****This is a
******Multiline Comment
**** */
#include <stdio.h>
void main()
{
FILE *fopen(),       *fp;
int c ;
fp = fopen( “prog.c”, “r” ); //Comment
12c = getc( fp ) ;
while ( c     !=     EOF )
{
putchar( c );
c = getc ( fp       );
}
/*multiline
comment */
fclose(
fp );
}
ugcse@prg28:~/Desktop/190905520/lab2$ ./solved
ugcse@prg28:~/Desktop/190905520/lab2$ cat q4out.c

#include <stdio.h>
void main()
{
FILE *fopen(), *fp;
int c ;
fp = fopen( “prog.c”, “r” ); 12c = getc( fp ) ;
while ( c != EOF )
{
putchar( c );
c = getc ( fp );
}

fclose(
fp );
}
ugcse@prg28:~/Desktop/190905520/lab2$
```

Question 1 : That takes a file as input and replaces blank spaces and tabs by single space and writes the output to a file.

Source Code :

```
#include <stdio.h>

int main(){
     FILE *fa, *fb;
     int ca, cb;
     fa = fopen("q4in.c", "r");
     if (fa == NULL){
          printf("Cannot open file \n");
          exit(0);
     }
     fb = fopen("q4out.c", "w");
     ca = getc(fa);
     while (ca != EOF){
          if(ca==' ' || ca == '\t'){
               putc(' ',fb);
               while(ca==' ' || ca == '\t')
                    ca = getc(fa);
          }
          if (ca=='/'){
               cb = getc(fa);
               if (cb == '/'){
                    while(ca != '\n')
                    ca = getc(fa);
               }
               else if (cb == '*'){
                    do{
                         while(ca != '*')
                              ca = getc(fa);
                         ca = getc(fa);
                    } while (ca != '/');
               }
               else{
                    putc(ca,fb);
                    putc(cb,fb);
               }
          }
          else putc(ca,fb);
          ca = getc(fa);
     }
     fclose(fa);
     fclose(fb);
     return 0;
}
```

Output :

```
ugcse@prg28:~/Desktop/190905520/lab2$ cat q4in.c
// This is a single line comment
/* *****This is a
******Multiline Comment
**** */
#include <stdio.h>
void main()
{
FILE             *fopen(),       *fp;
int c           ;
fp =    fopen( “prog.c”, “r” ); //Comment
12c             = getc( fp ) ;
while ( c     !=     EOF )
{
putchar( c );
c = getc ( fp       );
}
/*multiline
comment */
fclose(
fp );
}
ugcse@prg28:~/Desktop/190905520/lab2$ ./q1
ugcse@prg28:~/Desktop/190905520/lab2$ cat q4out.c

#include <stdio.h>
void main()
{
FILE *fopen(), *fp;
int c ;
fp = fopen( “prog.c”, “r” ); 12c = getc( fp ) ;
while ( c != EOF )
{
putchar( c );
c = getc ( fp );
}

fclose(
fp );
}
ugcse@prg28:~/Desktop/190905520/lab2$
```

Question 2 : To discard preprocessor directives from the given input ‘C’ file.

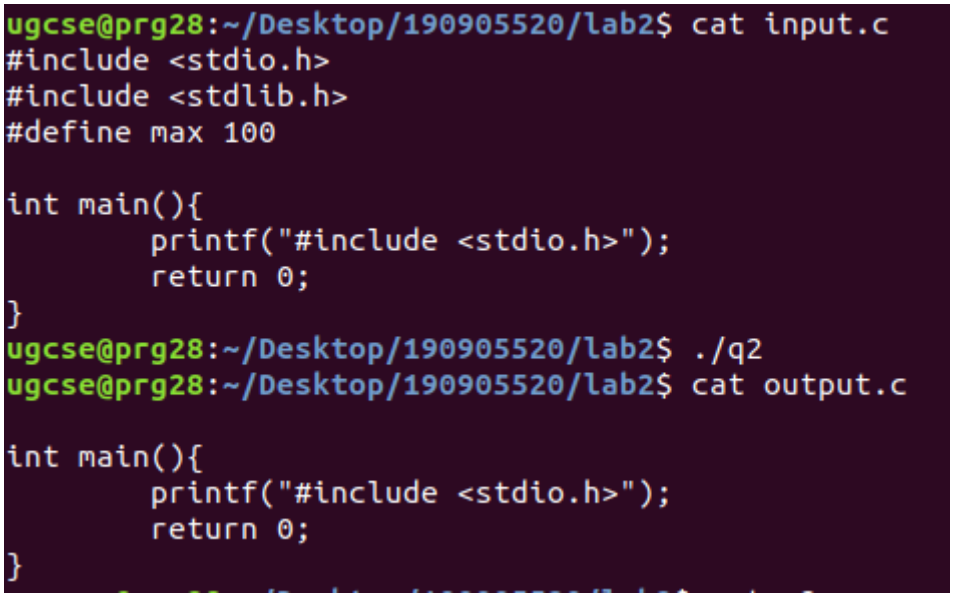
Source Code :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
     FILE *finp = fopen("input.c", "r");
     FILE *fout = fopen("output.c", "w+");
     char c = 0;
     char buffer[100];
     buffer[0]= '\0';
     int i = 0;
     char *includeStr = "include", *defineStr = "define",
*mainStr="main";
     int mainFlag = 0;
     while( c != EOF){
          c = fgetc(finp);
          if(c == '#' && mainFlag == 0){
               while(c!=' '){
                    c = fgetc(finp);
                    buffer[i++] = c;
               }
               buffer[i] = '\0';
               if( strstr(buffer, includeStr)!=NULL ||
strstr(buffer, defineStr)!=NULL){
                    while(c!='\n'){
                         c = fgetc(finp);
                    }
               }
               else{
                    fputc('#', fout);
                    for(int j = 0; j<i; j++){
                         fputc(buffer[j], fout);
                    }
                    while(c!='\n'){
                         c = fgetc(finp);
                         fputc(c, fout);
                    }
               }
               i = 0;
               buffer[0]= '\0';
          }
          else{
               if(mainFlag == 0){

                    buffer[i++] = c;
                    buffer[i] = '\0';
                    if(strstr(buffer, mainStr)!=NULL){
                         mainFlag = 1;
                    }
```

```
                }
                if(c == ' ' || c == '\n'){
                        buffer[0] = '\0';
                        i = 0;
                }
                fputc(c, fout);
        }
    }
    fclose(finp);
    fclose(fout);
    return 0;
}
```

Output :



Question 3 : That takes C program as input, recognizes all the keywords and prints them in upper case.

Source Code :

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

char keywords[32][10] = {"auto", "double", "int", "struct",
"break", "else", "long","switch", "case", "enum", "register",
"typedef", "char","extern", "return", "union", "const", "float",
"short", "unsigned", "continue", "for", "signed", "void",
"default", "goto", "sizeof", "voltile", "do", "if", "static",
"while"};

int compare(char buffer[]){
     for(int i = 0; i<32; i++){
          if(strcmp(buffer, keywords[i])==0){
               return 1;
          }
     }
     return 0;
}

int main(){
     FILE* finp = fopen("q3inp.c", "r");
     if(finp == NULL){
          printf("Cannot Find file, exiting ... ");
          return 0;
     }
     char buffer[100], c = 0;
     int i = 0;
     while(c!=EOF){
          if(isalpha(c) != 0){
          while(isalpha(c)!=0){
               c = fgetc(finp);
               if(isalpha(c)!=0)
                    buffer[i++] = c;
          }
          buffer[i] = '\0';
          if(compare(buffer) == 1){
               for(int j = 0; j<i ; j++){
                    printf("%c", toupper(buffer[j]));
               }
               printf("\n");
          }
          i = 0;
          buffer[0] = '\0';
          }
          else{
               c = fgetc(finp);
```

```
                if(isalpha(c)!=0){
                     buffer[i++] = c;
                }
           }
      }
      return 0;

}
```

Output :

```
ugcse@prg28:~/Desktop/190905520/lab2$ cat q3inp.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(){
        int a = 0;
        double b = 0.0;
        switch(a){
                case 0 :
                        break;
                default :
                        printf("hello world");
        }
        while(1){
                printf("hello world");
                continue;
        }
        char ctypee[100];
        if(a == 1){
                return 0;
        }
        else{
                return 1;
        }
        return 0;
}
ugcse@prg28:~/Desktop/190905520/lab2$ ./q3
INT
INT
DOUBLE
SWITCH
CASE
BREAK
DEFAULT
WHILE
CONTINUE
CHAR
IF
RETURN
ELSE
RETURN
RETURN
ugcse@prg28:~/Desktop/190905520/lab2$
```