LAB 2

Juhi Mehta
190905412
Roll No: 55
Batch B3

Sample Program 1: Removal of single and multiline comments

Code:

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

int main()
{
	FILE *fa, *fb;
	int ca, cb;
	fa = fopen("exinput.c", "r");
	if (fa == NULL)
	{
		printf("Cannot open file  \n");
		exit(0);
	}
	fb = fopen("exoutput.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:

```
Student@dblab-hp-21:~/Desktop/190905412/Lab_2$ gcc -o ex ex.c
Student@dblab-hp-21:~/Desktop/190905412/Lab_2$ ./ex
Student@dblab-hp-21:~/Desktop/190905412/Lab_2$ cat exoutput.c

#include <stdio.h>
void main()
{
        FILE *fopen(), *fp;
        int c ;
        fp = fopen( “prog.c”, “r” );    c = getc( fp ) ;
        while ( c != EOF )
        {
                putchar( c );
                c = getc ( fp );
        }
        fclose( fp );
Student@dblab-hp-21:~/Desktop/190905412/Lab_2$ cat exinput.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
        c = getc( fp ) ;
        while ( c     !=        EOF )
        {
                putchar( c );
                c = getc ( fp );
        }      /*multiline
        comment */
        fclose(      fp );
Student@dblab-hp-21:~/Desktop/190905412/Lab_2$
```

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

Code:

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

int main()
{
	char c1,c2;
	FILE *f1,*f2;
	f1= fopen("l2q1input.c","r");
	f2= fopen("outputl2q1.c","w");
	if(f1 == NULL || f2 ==NULL)
	{
		printf("Either the input or the output file does not exist \n");
		return 1;
	}
	c1=fgetc(f1);
	while(c1 != EOF)
	{
		if(c1 == '/')
		{
	c2 = getc(f1);
	if(c2 == '/')
	{
	   putc(c1,f2);
	   putc(c2,f2);
	   c1 = getc(f1);
	   while(c1 !='\n')
	   {
	      putc(c1,f2);
	      c1 = getc(f1);
	   }
	}
	else if(c2 == '*')
	{
	   putc(c1,f2);
	   putc(c2,f2);
	   c1 = getc(f1);
	   do
	   {
	      while(c1 != '*')
	      {
	         putc(c1,f2);
	         c1 = getc(f1);
	      }
	      putc(c1,f2);
	      c1 = getc(f1);
	   } while(c1 != '/');
	}
```

```
        }
        if(c1 == "")
        {
           putc(c1,f2);
           c1 = getc(f1);
           while(c1 != "")
           {
              putc(c1,f2);
              c1 = getc(f1);
           }
           putc(c1,f2);
           c1 = getc(f1);
        }
        if(c1 == ' ' || c1 == '\t')
        {
           putc(' ',f2);
           while(c1 == ' ' || c1 == '\t')
           {
              c1 = getc(f1);
           }
        }
        putc(c1,f2);
        c1 = getc(f1);
    }
           fclose(f1);
           fclose(f2);
           return 0;
}
```

Output:

```
Student@dblab-hp-21:~/Desktop/190905412/Lab_2$ gcc -o l2q1 l2q1.c
Student@dblab-hp-21:~/Desktop/190905412/Lab_2$ ./l2q1
Student@dblab-hp-21:~/Desktop/190905412/Lab_2$ cat l2q1input.c
// This is a    single line comment
/* *****This is a
******Multiline          Comment
**** */

#include        <stdio.h>

int main()
{
        printf("Hello   World");
        return   0;
}
Student@dblab-hp-21:~/Desktop/190905412/Lab_2$ cat outputl2q1.c
// This is a    single line comment
/* *****This is a
******Multiline          Comment
**** */

#include <stdio.h>

int main()
{
 printf("Hello  World");
 return 0;
}
Student@dblab-hp-21:~/Desktop/190905412/Lab_2$
```

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

Code:

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

int main()
{
        char c1,c2;
        FILE *f1,*f2;
        f1= fopen("l2q1input.c","r");
        f2= fopen("outputl2q1.c","w");
        if(f1 == NULL || f2 ==NULL)
        {
                printf("Either the input or the output file does not exist \n");
                return 1;
        }
        c1=fgetc(f1);
        while(c1 != EOF)
        {
                if(c1 == '/')
                {
```

```
        c2 = getc(f1);
        if(c2 == '/')
        {
            putc(c1,f2);
            putc(c2,f2);
            c1 = getc(f1);
            while(c1 !='\n')
            {
                putc(c1,f2);
                c1 = getc(f1);
            }
        }
        else if(c2 == '*')
        {
            putc(c1,f2);
            putc(c2,f2);
            c1 = getc(f1);
            do
            {
                while(c1 != '*')
                {
                    putc(c1,f2);
                    c1 = getc(f1);
                }
                putc(c1,f2);
                c1 = getc(f1);
            } while(c1 != '/');
        }
    }
    if(c1 == '"')
    {
        putc(c1,f2);
        c1 = getc(f1);
        while(c1 != '"')
        {
            putc(c1,f2);
            c1 = getc(f1);
        }
        putc(c1,f2);
        c1 = getc(f1);
    }
    if(c1 == '#')
    {
        while(c1 != '\n')
            c1 = getc(f1);
    }
    putc(c1,f2);
    c1 = getc(f1);
}
        fclose(f1);
        fclose(f2);
        return 0;
```

}

Output:

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

Code:

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

int main(){
   char keys[10][20] = {"if","else","while","int","char","float","do","printf","scanf","for"};
   char buffer[100];
   FILE *f1;
   int c1,bufferCounter = 0;
   f1 = fopen("l2q1input.c","r");
   if(f1 == NULL){
```

```
        printf("Cannot open file\n");
        exit(0);
    }
    c1 = getc(f1);
    while(c1 != EOF){
        if(isalpha(c1)){
            buffer[bufferCounter++] = c1;
        }
        else{
            buffer[bufferCounter] = '\0';
            bufferCounter = 0;
            for(int i = 0 ; i < 10 ; i++){
                if(strcmp(keys[i],buffer) == 0){
                    int j = 0;
                    while(buffer[j]){
                        putchar(toupper(buffer[j]));
                        j++;
                    }
                    printf("\n");
                    break;
                }
            }
        }
        c1 = getc(f1);
    }
    fclose(f1);
    return 0;
}
```

Output:

```
Student@dblab-hp-21:~/Desktop/190905412/Lab_2$ gcc -o l2q3 l2q3.c
Student@dblab-hp-21:~/Desktop/190905412/Lab_2$ ./l2q3
INT
CHAR
IF
ELSE
PRINTF
Student@dblab-hp-21:~/Desktop/190905412/Lab_2$ cat l2q1input.c
#include        <stdio.h>

int main()
{
        char c='a';
        if(c=='a')
                c++;
        else
                c='z';
        printf("%c",c);
        return   0;
}
Student@dblab-hp-21:~/Desktop/190905412/Lab_2$
```