Ayush Goyal
190905522 CSE D Roll 62

Compiler Design Lab 2: Preliminary Scanning Applications

Solved Excercise : Removal of single and multiline comments

Code:

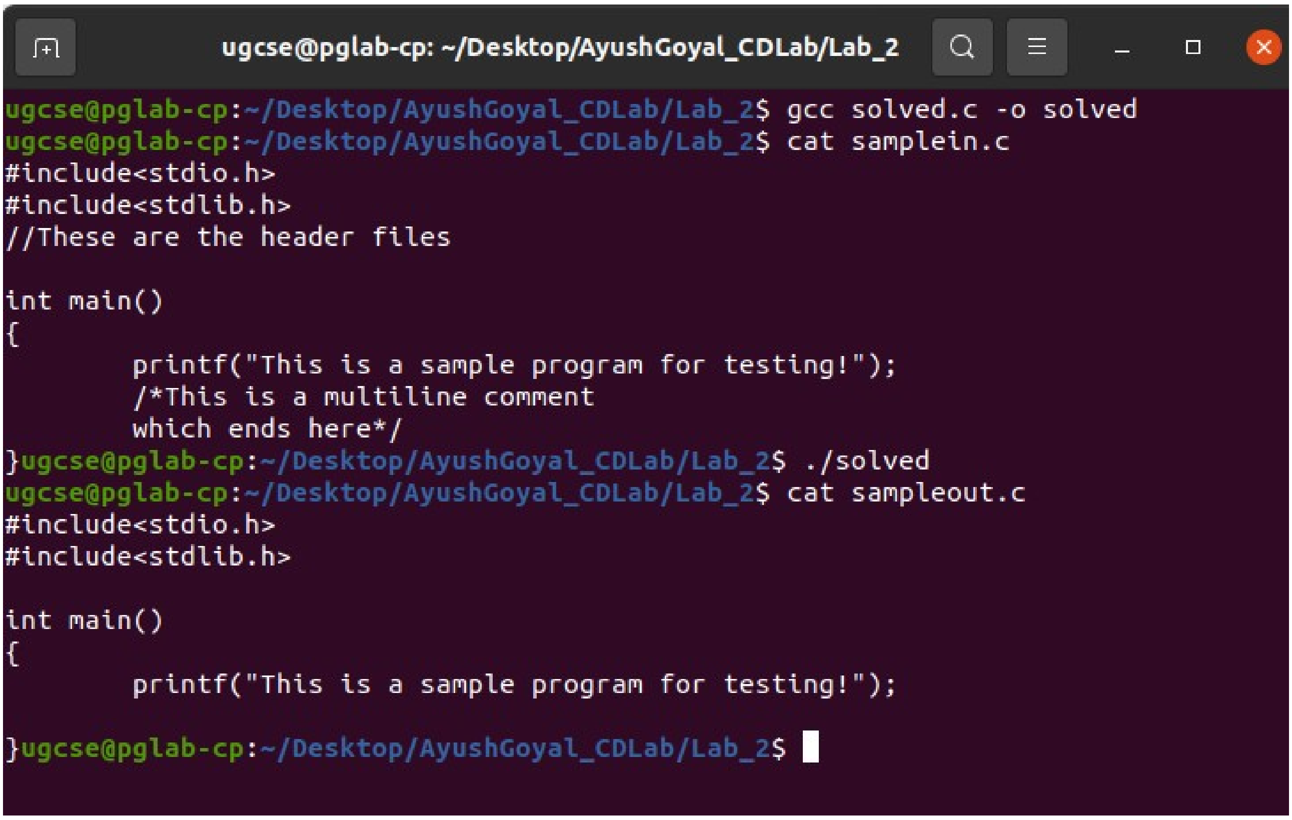
```
#include<stdio.h>
#include<stdlib.h>

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

We create a sample c file called “samplein.c” where we keep both single and multiline comments and we execute our program and get the output in the file “sampleout.c” as shown:

Output:



```
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ gcc solved.c -o solved
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ cat samplein.c
#include<stdio.h>
#include<stdlib.h>
//These are the header files

int main()
{
        printf("This is a sample program for testing!");
        /*This is a multiline comment
        which ends here*/
}ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ ./solved
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ cat sampleout.c
#include<stdio.h>
#include<stdlib.h>

int main()
{
        printf("This is a sample program for testing!");

}ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ 
```

Lab Exercises:

Write a C program:

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

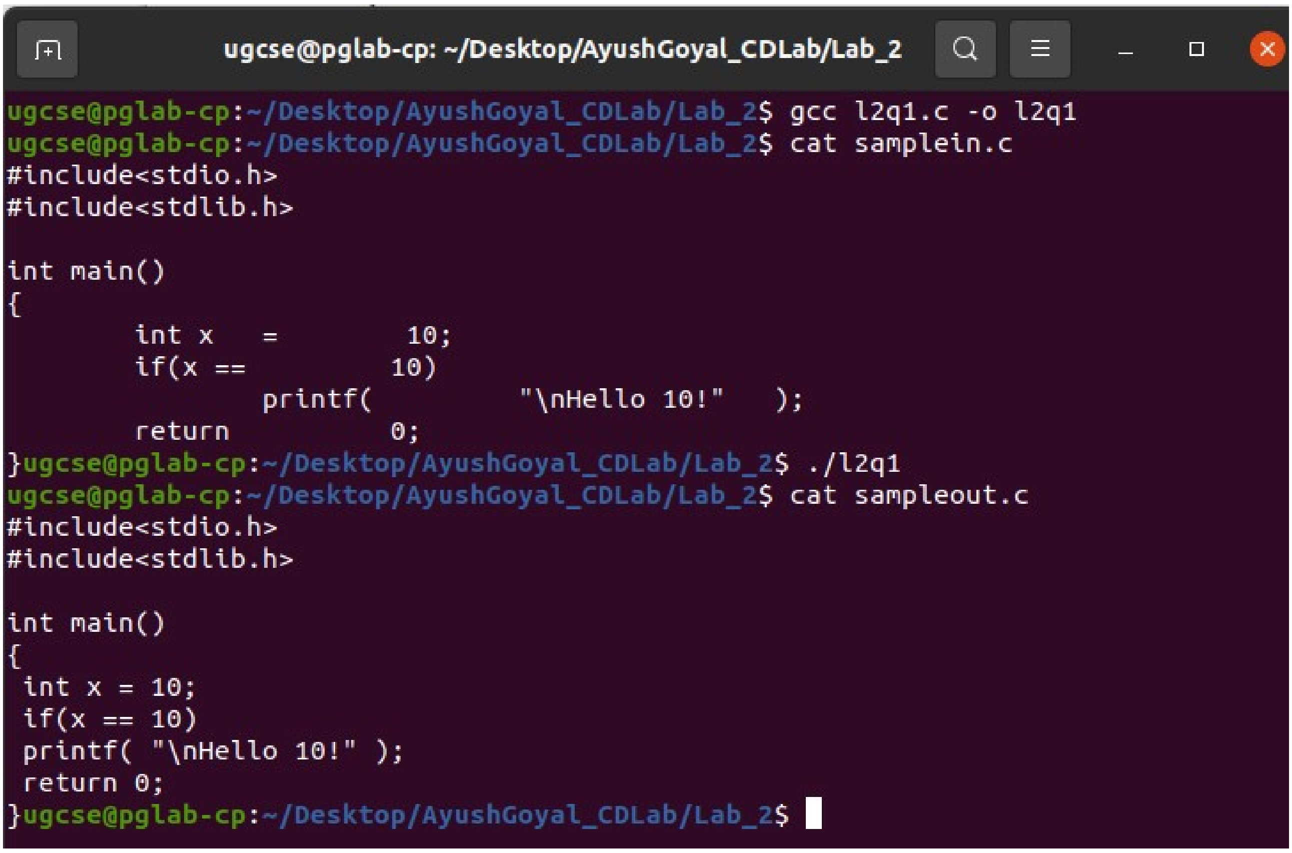
Code:

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

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

We use a new “samplein.c” file to test the current program and we store the output in “sampleout.c” file. The output is as shown below:

Output:



```
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ gcc l2q1.c -o l2q1
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ cat samplein.c
#include<stdio.h>
#include<stdlib.h>

int main()
{
        int x   =        10;
        if(x ==         10)
                printf(         "\nHello 10!"    );
        return          0;
}ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ ./l2q1
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ cat sampleout.c
#include<stdio.h>
#include<stdlib.h>

int main()
{
 int x = 10;
 if(x == 10)
 printf( "\nHello 10!" );
 return 0;
}ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ 
```

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

Code:

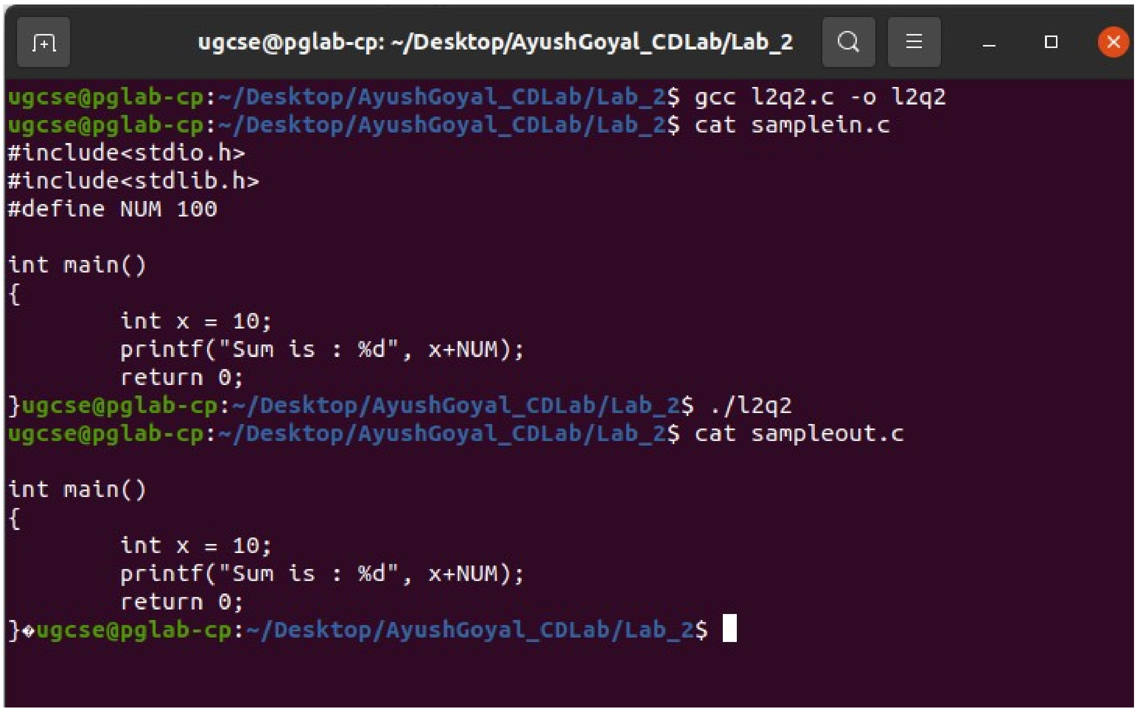
```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

int main(){
	FILE *finp = fopen("samplein.c","r");
	FILE *fout = fopen("sampleout.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;
}
```

We create a new “samplein.c” file to test this program the contents of which is given below. We execute the program and get the required output in “sampleout.c”. We can see all the preprocessor directives have been removed.

Output:



```
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ gcc l2q2.c -o l2q2
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ cat samplein.c
#include<stdio.h>
#include<stdlib.h>
#define NUM 100

int main()
{
        int x = 10;
        printf("Sum is : %d", x+NUM);
        return 0;
}ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ ./l2q2
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ cat sampleout.c

int main()
{
        int x = 10;
        printf("Sum is : %d", x+NUM);
        return 0;
}�ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ 
```

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

Code:

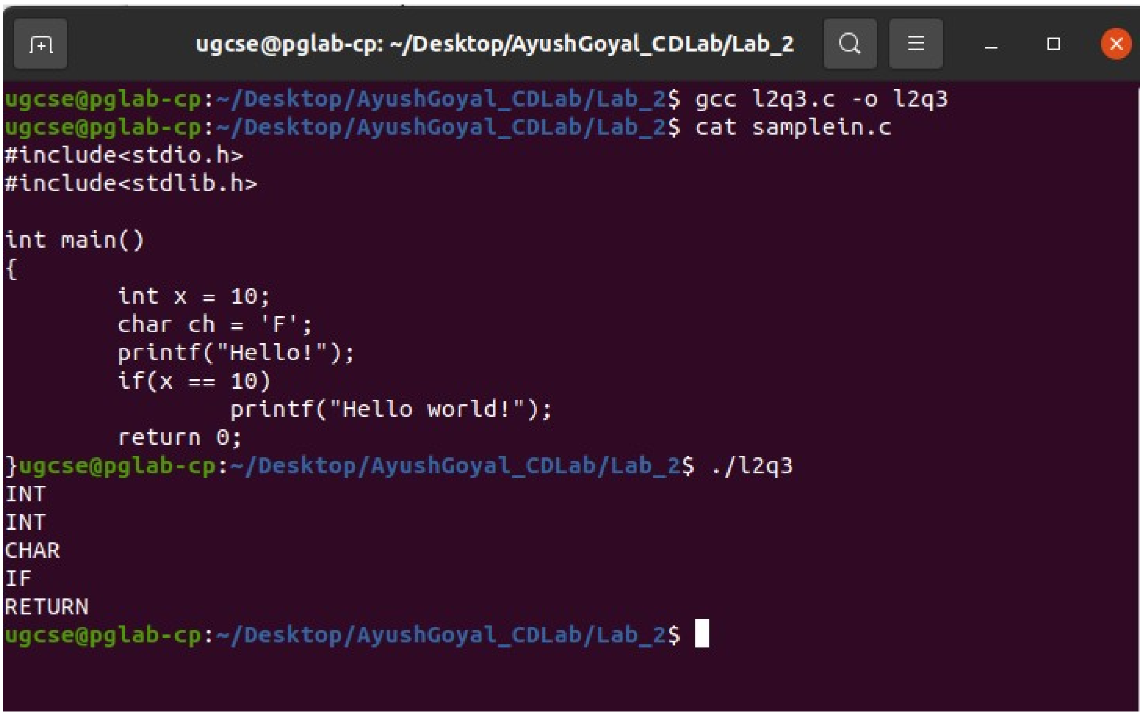
```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<ctype.h>

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* fin = fopen("samplein.c","r");
        if(fin == NULL){
                printf("Cannot find file!\n");
                exit(0);
        }
        char buffer[100], c = 0;
        int i = 0;
        while(c != EOF){
                if(isalpha(c) != 0){
                        while(isalpha(c) != 0){
                                c = fgetc(fin);
                                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(fin);
                        if(isalpha(c) != 0)
                                buffer[i++] = c;
                }
        }
        fclose(fin);
        return 0;
}
```

Output:



```
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ gcc l2q3.c -o l2q3
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ cat samplein.c
#include<stdio.h>
#include<stdlib.h>

int main()
{
        int x = 10;
        char ch = 'F';
        printf("Hello!");
        if(x == 10)
                printf("Hello world!");
        return 0;
}ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ ./l2q3
INT
INT
CHAR
IF
RETURN
ugcse@pglab-cp:~/Desktop/AyushGoyal_CDLab/Lab_2$ 
```

THE END