

CD LAB - WEEK 2

Pranay Goel

220905524

CSE B

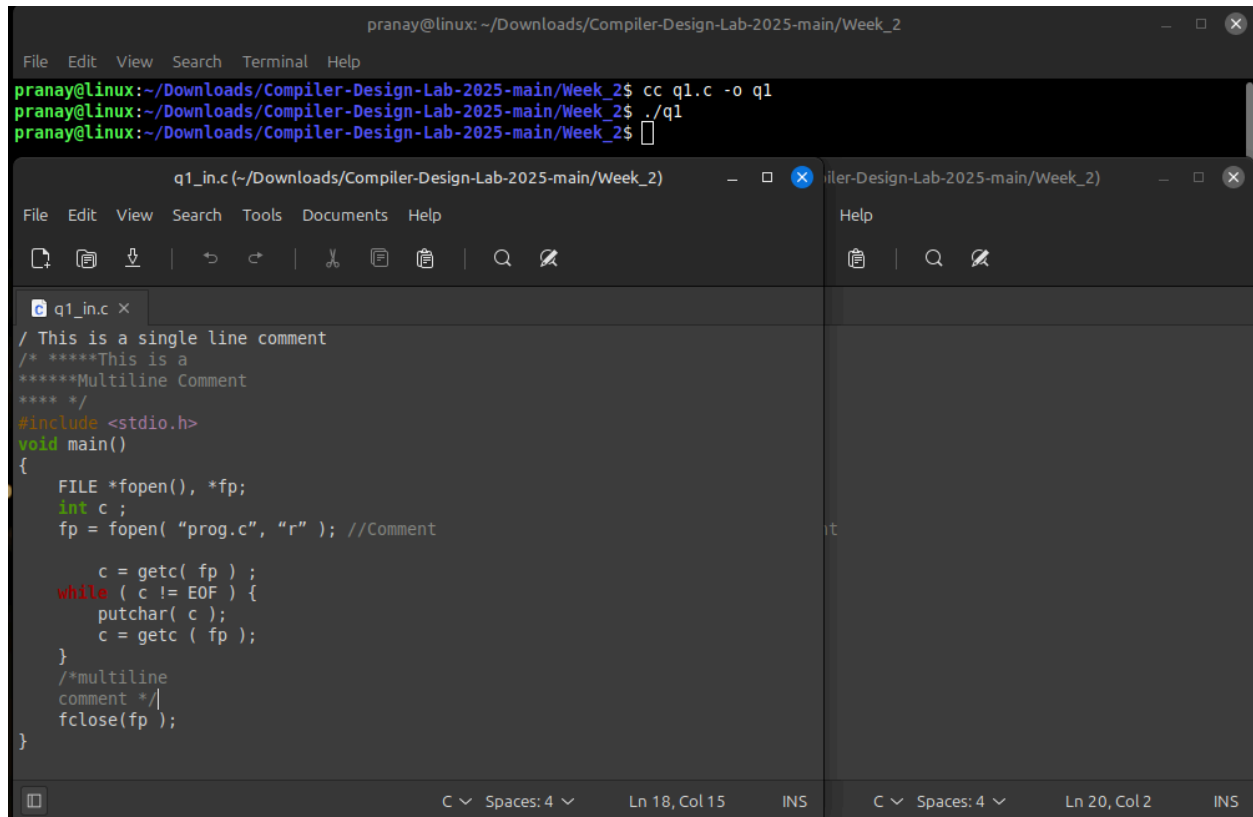
Write a 'C' program

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

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

int main(){
    FILE *fa, *fb;
    int c;
    fa = fopen("q1_in.c", "r");
    if (fa == NULL) {
        printf("Cannot open file \n");
        exit(0);
    }
    fb = fopen("q1_out.c", "w");

    c = getc(fa);
    while (c != EOF) {
        if(c == '\t' || c == ' ') {
            putc(' ', fb);
            while(c == ' ' || c == '\t')
                c = getc(fa);
            putc(c, fb);
        }
        else putc(c, fb);
        c = getc(fa);
    }
    fclose(fa);
    fclose(fb);
    return 0;
}
```



The screenshot shows a Linux terminal window and a code editor. The terminal window at the top displays the following commands and output:

```
pranay@linux: ~/Downloads/Compiler-Design-Lab-2025-main/Week_2
pranay@linux:~/Downloads/Compiler-Design-Lab-2025-main/Week_2$ cc q1.c -o q1
pranay@linux:~/Downloads/Compiler-Design-Lab-2025-main/Week_2$ ./q1
```

The code editor below shows the contents of the file `q1_in.c`:

```
q1_in.c x
/ 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 );
}
```

The status bar at the bottom of the editor shows the current cursor position as Ln 18, Col 15.

2. to discard preprocessor directives from the given input 'C' file.

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

```
#define MAX_BUFFER_SIZE 1024
```

```
int isPreprocessorDirective(const char *line) {
    if (strncmp(line, "#include", 8) == 0 ||
        strncmp(line, "#define", 7) == 0 ||
        strncmp(line, "#ifdef", 6) == 0 ||
        strncmp(line, "#ifndef", 7) == 0 ||
        strncmp(line, "#endif", 6) == 0) {
        return 1;
    }
    return 0;
}
```

```
int main(int argc, char *argv[]) {
    FILE *fa, *fb;
```

```

char c;
char buffer[MAX_BUFFER_SIZE];
int bufferIndex = 0;

fa = fopen("q2_in.c", "r");
if (fa == NULL) {
    printf("Error opening input file.\n");
    return 1;
}

fb = fopen("q2_out.c", "w");
if (fb == NULL) {
    printf("Error opening output file.\n");
    fclose(fa);
    return 1;
}

while ((c = getc(fa)) != EOF) {
    if (c == '#'){
        bufferIndex = 0;
        buffer[bufferIndex++] = c;
        while ((c = getc(fa)) != EOF && c != '\n' && bufferIndex < MAX_BUFFER_SIZE - 1){
            buffer[bufferIndex++] = c;
        }
        buffer[bufferIndex] = '\0'; // Null-terminate the buffer

        if (!isPreprocessorDirective(buffer)){
            // If not a directive write the # followed by the content to the output file
            for (int i = 0; i < bufferIndex; i++) {
                putc(buffer[i], fb);
            }
        }

        if (c != EOF && c == '\n'){
            putc(c, fb);
        }
    }
    else{
        putc(c, fb);
    }
}

fclose(fa);
fclose(fb);

```

```

printf("Preprocessor directives removed and output written to q2_out.c\n");
}

```

The screenshot shows a terminal window and two code editors. The terminal window shows the command `cc q2.c -o q2` and the output `Preprocessor directives removed and output written to q2_out.c`. The two code editors show the source code `q2_in.c` and the preprocessed output `q2_out.c`. The source code `q2_in.c` includes `<stdio.h>`, defines `hello`, and has a `main` function that opens a file, reads characters, and prints them. The preprocessed output `q2_out.c` shows the same code but with the preprocessor directives removed.

```

pranay@linux: ~/Downloads/Compiler-Design-Lab-2025-main/Week_2
File Edit View Search Terminal Help
pranay@linux:~/Downloads/Compiler-Design-Lab-2025-main/Week_2$ cc q2.c -o q2
pranay@linux:~/Downloads/Compiler-Design-Lab-2025-main/Week_2$ ./q2
Preprocessor directives removed and output written to q2_out.c
pranay@linux:~/Downloads/Compiler-Design-Lab-2025-main/Week_2$

q2_in.c (~Downloads/Compiler-Des
File Edit View Search Tools Documents Help
#include <stdio.h>
#define hello
#komal
#innnclude<./>
void main()
{
    FILE *fopen(), *fp;
    int c ;
    fp = fopen( "prog.c", "r" ); //Comment
    printf("\n#def");

    c = getc( fp ) ;
    while ( c != EOF ) {
        putchar( c );
        c = getc ( fp );
    }
    /*multiline
    comment */
    fclose(fp );
}

q2_out.c (~Downloads/Compiler-Design-Lab-2025-main/Week_2)
File Edit View Search Tools Documents Help
#komal
#innnclude<./>
void main()
{
    FILE *fopen(), *fp;
    int c ;
    fp = fopen( "prog.c", "r" ); //Comment
    printf("\n#def");

    c = getc( fp ) ;
    while ( c != EOF ) {
        putchar( c );
        c = getc ( fp );
    }
    /*multiline
    comment */
    fclose(fp );
}

```

3. that takes a C program as input, recognizes all the keywords and prints them in uppercase.

```

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

```

```

#define MAX_BUFFER_SIZE 100
#define NUM_KEYWORDS 7

```

```

void toUpperCase(char* str) {
    for (int i = 0; str[i] != '\0'; i++) {
        str[i] = toupper(str[i]);
    }
}

```

```

int isKeyword(const char* word, char* keywords[], int num_keywords) {
    for (int i = 0; i < num_keywords; i++) {
        if (strcmp(word, keywords[i]) == 0)

```

```

        return 1;
    }
    return 0;
}

int main() {
    FILE *fa, *fb;
    char c;
    char buffer[MAX_BUFFER_SIZE];
    int bufferIndex = 0;
    char *keywords[] = {"include", "define", "if", "else", "while", "for", "return"};

    fa = fopen("q3_in.c", "r");
    if (fa == NULL) {
        printf("Error opening input file.\n");
        return 1;
    }

    fb = fopen("q3_out.c", "w");
    if (fb == NULL) {
        printf("Error opening output file.\n");
        fclose(fa);
        return 1;
    }

    while ((c = getc(fa)) != EOF) {
        if (isalpha(c) || c == '_') {
            buffer[bufferIndex++] = c;

            // Make sure the buffer doesn't overflow
            if (bufferIndex >= MAX_BUFFER_SIZE) {
                bufferIndex = MAX_BUFFER_SIZE - 1;
            }
        }
        else {
            if (bufferIndex > 0) {
                buffer[bufferIndex] = '\0';
                if (isKeyword(buffer, keywords, NUM_KEYWORDS)) {
                    toUpperCase(buffer);
                }
                for (int i = 0; buffer[i] != '\0'; i++) {
                    putc(buffer[i], fb);
                }
                bufferIndex = 0;
            }
        }
    }
}

```

```

    }
    putc(c, fb);
}
}

// If there's any word left in the buffer ie file ends with word
if (bufferIndex > 0) {
    buffer[bufferIndex] = '\0';
    if (isKeyword(buffer, keywords, NUM_KEYWORDS)) {
        toUpperCase(buffer);
    }
    for (int i = 0; buffer[i] != '\0'; i++) {
        putc(buffer[i], fb);
    }
}

fclose(fa);
fclose(fb);
printf("Keywords converted to uppcase and written to q3_out.c\n");
}

```

The screenshot shows a Linux terminal window and two code editors. The terminal window at the top shows the compilation and execution of a C program. The code editors below show the source code for the program.

Terminal Window:

```

pranay@linux: ~/Downloads/Compiler-Design-Lab-2025-main/Week_2
pranay@linux:~/Downloads/Compiler-Design-Lab-2025-main/Week_2$ cc q3.c -o q3
pranay@linux:~/Downloads/Compiler-Design-Lab-2025-main/Week_2$ ./q3
Keywords converted to uppcase and written to q3_out.c
pranay@linux:~/Downloads/Compiler-Design-Lab-2025-main/Week_2$

```

Code Editor 1: *q3_in.c

```

#include <stdio.h>
#define hello
void main()
{
    FILE *fopen(), *fp;
    int c ;
    fp = fopen( "prog.c", "r" ); //Comment
    printf("\n#define");

    c = getc( fp ) ;
    while ( c != EOF ) {
        putchar( c );
        c = getc ( fp );
    }
    for(int i=0;i<100;i++);
    fo
    r
    /*multiline
    comment */
    fclose(fp );
}

```

Code Editor 2: *q3_out.c

```

#include <stdio.h>
#define hello
void main()
{
    FILE *fopen(), *fp;
    int c ;
    fp = fopen( "prog.c", "r" ); //Comment
    printf("\n#define");

    c = getc( fp ) ;
    WHILE ( c != EOF ) {
        putchar( c );
        c = getc ( fp );
    }
    FOR(int i=0;i<100;i++);
    fo
    r
    /*multiline
    comment */
    fclose(fp );
}

```

The status bar at the bottom of the code editors shows "C", "Spaces: 4", "Ln 21, Col 2", and "INS".