Lab 02: Preliminary Scanning Applications

Ishaan Vatus, CSE-D, Roll No. 10, 220905043

CODE:

```c
/*
 * lexer.h
 * function signatures
 */
#ifndef LEXER_H
#define LEXER_H
#define TAB 9
#define SPACE 32
void remove_whitespace(char *input_file, char *output_file);
void remove_preprocessor(char *input_file, char *output_file);
void regurge_keywords(char *input_file, char *output_file);
#endif
```

```c
#include <stdbool.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "lexer.h"

void remove_whitespace(char *input_file, char *output_file)
{
    char ch;
    bool flag = false;
    FILE *fp = fopen(input_file, "r");
    FILE *op = fopen(output_file, "w");
    while ((ch = getc(fp)) != EOF) {
        if ((ch == SPACE || ch == TAB) && !flag) {
            putc(SPACE, op);
            flag = true;
        }
        else if ((ch == SPACE || ch == TAB) && flag)
            continue;
        else {
            putc(ch, op);
            flag = false;
        }
    }
    fclose(op);
    fclose(fp);
}
void remove_preprocessor(char *input_file, char *output_file)
```

```
{
    FILE *fp = fopen(input_file, "r");
    FILE *op = fopen(output_file, "w");
    char ch;
    while ((ch = getc(fp)) != EOF) {
        while (ch == TAB || ch == SPACE)
            ch = getc(fp);
        if (ch == '#') {
            while (ch != '\n')
                ch = getc(fp);
            continue;
        }
        else {
            putc(ch, op);
            while (ch != '\n') {
                ch = getc(fp);
                putc(ch, op);

            }
        }
    }
    fclose(fp);
    fclose(op);
}
void regurge_keywords(char *input_file, char *output_file)
{
    int keyword_count = 54;
    char *keywords[] = {"alignas","alignof","auto","bool","break","case","char","const",
    "constexpr","continue","default","do","double","else","enum","extern","false","float",
    "for","goto","if","inline","int","long","nullptr","register","restrict","return","short"
    "signed","sizeof","static","static_assert","struct","switch","thread_local","true","type
    "typeof","typeof_unqual","union","unsigned","void","volatile","while","_Atomic","_BitInt
    "_Complex","_Decimal128","_Decimal32","_Decimal64","_Generic","_Imaginary","_Noreturn"};
    FILE *fp = fopen(input_file, "r");
    FILE *op = fopen(output_file, "w");
    int text_len = 0;
    char ch;
    while ((ch = getc(fp)) != EOF)
        text_len++;
    rewind(fp);
    char *text = malloc(text_len*sizeof(char));
    fread(text, sizeof(char), text_len, fp);
    for (int index = 0; index < keyword_count; index++) {
        int pattern_len = strlen(keywords[index]);
        for (int i = 0; i < text_len - pattern_len; i++) {
            int matches;
```

```
            for (matches = 0; matches < pattern_len; matches++) {
                if (text[i+matches] != keywords[index][matches])
                    break;
            }
            if (matches == pattern_len) {
                fwrite(keywords[index], sizeof(char), pattern_len, op);
                fwrite("\n", sizeof(char), 1, op);
            }

        }
    }
    fclose(fp);
    fclose(op);
}
```