LAB ASIGNMENT - 8

COMPILER DESIGN LAB

NAME – ADITYARAJ SHRIVASTAVA
REG. NO.- 23BCE1968

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

char* trim_leading_spaces(char *str) {
    while(isspace(*str)) str++;
    return str;
}

int starts_with(const char *str, const char *prefix) {
    while (*prefix) {
        if (*prefix++ != *str++)
            return 0;
    }
    return 1;
}

int check_for_loop(const char *code) {
    const char *ptr = code;
    while ((ptr = strstr(ptr, "for")) != NULL) {
        if ((ptr == code || !isalnum(*(ptr - 1))) && !isalnum(*(ptr + 3))) {
            const char *p = ptr + 3;
            while (*p && isspace(*p)) p++;
            if (*p == '(') {
                const char *start = p;
                while (*p && *p != ')') p++;
                if (*p == ')') {
                    int semicolon_count = 0;
                    for (const char *q = start; q < p; q++) {
                        if (*q == ';')
                            semicolon_count++;
                    }
                    if (semicolon_count == 2) return 1;
                }
            }
        }
        ptr += 3;
    }
    return 0;
}

int check_while_loop(const char *code) {
    const char *ptr = code;
    while ((ptr = strstr(ptr, "while")) != NULL) {
```

```
        if ((ptr == code || !isalnum(*(ptr - 1))) && !isalnum(*(ptr + 5))) {
            const char *p = ptr + 5;
            while (*p && isspace(*p)) p++;
            if (*p == '(') {
                while (*p && *p != ')') p++;
                if (*p == ')') return 1;
            }
        }
        ptr += 5;
    }
    return 0;
}

int check_if_else_if(const char *code) {
    if (strstr(code, "else if") == NULL) return 0;
    const char *pos = code;
    while ((pos = strstr(pos, "if")) != NULL) {
        const char *paren_open = strchr(pos, '(');
        const char *paren_close = strchr(pos, ')');
        if (paren_open == NULL || paren_close == NULL || paren_close <
paren_open)
            return 0;
        pos += 2;
    }
    return 1;
}

int check_if_else(const char *code) {
    if (strstr(code, "if") == NULL || strstr(code, "else") == NULL)
        return 0;
    const char *if_pos = strstr(code, "if");
    const char *else_pos = strstr(code, "else");
    if (if_pos > else_pos)
        return 0;
    const char *paren_open = strchr(if_pos, '(');
    const char *paren_close = strchr(if_pos, ')');
    if (paren_open == NULL || paren_close == NULL || paren_close < paren_open)
        return 0;
    return 1;
}

int check_switch_case(const char *code) {
    if (strstr(code, "switch") == NULL || strstr(code, "case") == NULL)
        return 0;
    if (strstr(code, "{") == NULL || strstr(code, "}") == NULL)
```

```
        return 0;
    const char *switch_pos = strstr(code, "switch");
    const char *paren_open = strchr(switch_pos, '(');
    const char *paren_close = strchr(switch_pos, ')');
    if (paren_open == NULL || paren_close == NULL || paren_close < paren_open)
        return 0;
    return 1;
}

char* read_file(const char* filename) {
    FILE *file = fopen(filename, "r");
    if (!file) {
        printf("Cannot open file.\n");
        exit(1);
    }
    fseek(file, 0, SEEK_END);
    long length = ftell(file);
    rewind(file);
    char *buffer = malloc(length + 1);
    if (!buffer) {
        printf("Memory error.\n");
        exit(1);
    }
    fread(buffer, 1, length, file);
    buffer[length] = '\0';
    fclose(file);
    return buffer;
}

int main(int argc, char *argv[]) {
    if (argc != 2) {
        printf("Usage: %s filename\n", argv[0]);
        return 1;
    }
    char *code = read_file(argv[1]);

    int found = 0;
    if (check_for_loop(code)) {
        printf("Valid for loop syntax\n");
        found = 1;
    }
    if (check_while_loop(code)) {
        printf("Valid while loop syntax\n");
        found = 1;
    }
```

```
    if (check_if_else_if(code)) {
        printf("Valid if-else-if syntax\n");
        found = 1;
    }
    else if (check_if_else(code)) {
        printf("Valid if-else syntax\n");
        found = 1;
    }
    if (check_switch_case(code)) {
        printf("Valid switch-case syntax\n");
        found = 1;
    }
    if (!found) {
        printf("Syntax not recognized or invalid\n");
    }
    free(code);
    return 0;
}
```

OUTPUT:

1:

input.txt

```
1   for (i = 0; i < 10; i++) { sum += i; }
2
3   switch (grade) {
4       case 1: printf("A"); break;
5       case 2: printf("B"); break;
6       default: printf("F");
7   }
8
9   if (x > 0) { y = 1; }
10  else if (x < 0) { y = -1; }
11  else { y = 0; }
```

```
PS C:\Users\scope1\Desktop\23bce1968> ./a input.txt
Valid for loop syntax
Valid if-else-if syntax
Valid switch-case syntax
PS C:\Users\scope1\Desktop\23bce1968>
```

2:

```
for i = 0; i < 10; i++
    printf("%d", i);
END
```

```
PS C:\Users\scope1\Desktop\23bce1968> ./a input.txt
Syntax not recognized or invalid
PS C:\Users\scope1\Desktop\23bce1968>
```