ASSIGNMENT 01

122CS0967

CODE

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

// Function prototypes
int check_header(const char *line);
int check_body(const char *line);
int check_arithmetic_operation(const char *line);
int check_logic_operation(const char *line);
int check_results(const char *line);
int check_print(const char *line);

int main(int argc, char *argv[]) {
  if (argc != 2) {
    fprintf(stderr, "Usage: %s <input_file>\n", argv[0]);
    return 1;
  }

  FILE *file = fopen(argv[1], "r");
  if (!file) {
```

```
    perror("Error opening file");
    return 1;
}

char line[256];
int header_valid = 0;
int body_valid = 0;
int arithmetic_valid = 0;
int logic_valid = 0;
int results_valid = 0;
int print_valid = 0;
int state = 0;

while (fgets(line, sizeof(line), file))
    line[strcspn(line, "\n")] = '\0';

    switch (state) {
        case 0:
            if (strstr(line, "Begin_Header") == line) {
                header_valid = check_header(line);
                if (header_valid) state = 1;
            }
            break;
        case 1:
            if (strcmp(line, "Body{") == 0) {
                body_valid = 1;
```

```
            state = 2;
        }
        break;
    case 2:
        if (strstr(line, "ArithmaticOperation:") == line) {
            arithmetic_valid = check_arithmetic_operation(line);
        } else if (strstr(line, "LogicOperation:") == line) {
            logic_valid = check_logic_operation(line);
        } else if (strcmp(line, "Results(") == 0) {
            results_valid = 1;
            state = 3;
        }
        break;
    case 3:
        if (strstr(line, "Print(") == line) {
            print_valid = check_print(line);
        } else if (strcmp(line, "}") == 0) {
            state = 4;
        }
        break;
    case 4:
        break;
  }
}

fclose(file);
```

```
    if (header_valid && body_valid && arithmetic_valid && logic_valid && results_valid &&
print_valid) {
        printf("The input file is valid.\n");
    } else {
        printf("The input file is invalid.\n");
    }

    return 0;
}

int check_header(const char *line) {
    // Expected format: "Begin_Header <Name_HeaderFile> End_Header"
    const char *format = "Begin_Header %*s End_Header";
    return sscanf(line, format) == 1;
}

int check_arithmetic_operation(const char *line) {
    // Expected format: "ArithmaticOperation: result = operand_1 + operand_2;"
    const char *format = "ArithmaticOperation: %*s = %*s + %*s;";
    return sscanf(line, format) == 1;
}

int check_logic_operation(const char *line) {
    // Expected format: "LogicOperation: Output = input_1 & input_2;"
    const char *format = "LogicOperation: %*s = %*s & %*s;";
```

```
    return sscanf(line, format) == 1;
}

int check_print(const char *line) {
    // Expected format: "Print(result);" or "Print(output);"
    const char *format = "Print(%*s);";
    return sscanf(line, format) == 1;
}
```

INPUT FORMAT

```
Begin_Header lib.h End_Header
Body{ArithmaticOperation: result = operand_1 + operand_2;
LogicOperation: Output = input_1 & input_2;
Results(Print(result), Print(output);
}
```