CS 358 LAB 2

Arnav Jain - 220002018

Question 1

Code:

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

int main()
{
    char str[100];
    int vowels = 0, consonants = 0, i = 0;

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);

    while (str[i] != '\0')
    {
        char ch = tolower(str[i]);
        if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
        {
            vowels++;
        }
        else if ((ch >= 'a' && ch <= 'z'))
        {
            consonants++;
        }
        i++;
    }

    printf("Vowels: %d\n", vowels);
    printf("Consonants: %d\n", consonants);

    return 0;
}
```

Output:

```
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller/LAB 2$ ./q1
Enter a string: hello
Vowels: 2
Consonants: 3
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller/LAB 2$ ./q1
Enter a string: 12222
Vowels: 0
Consonants: 0
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller/LAB 2$ ./q1
Enter a string: arnav
Vowels: 2
Consonants: 3
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller/LAB 2$
```

Question 2

Code:

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

int isKeyword(char *str)
{
    char *keywords[] = {
        "int", "float", "char", "if", "else", "while", "for", "return", "void", "switch",
        "case", "break", "continue", "default", "struct", "union", "typedef", "long",
        "short", "unsigned", "signed", "enum", "const", "volatile", "goto", "NULL" , NULL};

    for (int i = 0; keywords[i] != NULL; i++)
    {
        if (strcmp(str, keywords[i]) == 0)
        {
            return 1;
        }
    }
    return 0;
}

int main()
{
    char str[100];

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);

    str[strcspn(str, "\n")] = '\0';

    if (isKeyword(str))
    {
        printf("\"%s\" is a keyword.\n", str);
    }
    else
    {
        printf("\"%s\" is not a keyword.\n", str);
    }

    return 0;
}
```

Result:

```
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller/LAB 2$ ./q2
 Enter a string: hello
 "hello" is not a keyword.
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller/LAB 2$ ./q2
 Enter a string: int
 "int" is a keyword.
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller/LAB 2$ ./q2
 Enter a string: intx
 "intx" is not a keyword.
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller/LAB 2$ ./q2
 Enter a string: NULL
 "NULL" is a keyword.
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller/LAB 2$ ./q2
 Enter a string: null
 "null" is not a keyword.
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller/LAB 2$
```

Question 3

Code:

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

#define MAX_LINE_LENGTH 1024

void checkSingleLineComment(char *line) {
    char *commentStart = strstr(line, "//");
    if (commentStart != NULL) {
        printf("Single-line comment: %s\n", commentStart + 2);
    }
}

void handleMultiLineComment(FILE *file) {
    char line[MAX_LINE_LENGTH];
    int isMultiLineComment = 0;

    while (fgets(line, sizeof(line), file)) {
        char *start = strstr(line, "/*");
        char *end = strstr(line, "*/");

        if (isMultiLineComment) {

            printf("Multi-line comment: %s", line);
            if (end != NULL) {
                isMultiLineComment = 0;
            }
        } else if (start != NULL) {
            printf("Multi-line comment: %s", start);
            isMultiLineComment = 1;
            if (end != NULL) {
                isMultiLineComment = 0;
            }
        }
    }
}

int isCommentLine(char *line) {

    if (strstr(line, "//") != NULL) {
        return 1;
    }
    if (strstr(line, "/*") != NULL || strstr(line, "*/") != NULL) {
        return 1;
    }
    return 0;
}
```

```c
     int isCommentLine(char *line) {

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

         FILE *file = fopen(argv[1], "r");
         if (file == NULL) {
             perror("Error opening file");
             return 1;
         }

         char line[MAX_LINE_LENGTH];
         int lineNumber = 0;
         int isInsideMultiLineComment = 0;

         while (fgets(line, sizeof(line), file)) {
             lineNumber++;

             if (isInsideMultiLineComment) {
                 printf("Multi-line comment: %s", line);
                 if (strstr(line, "*/") != NULL) {
                     isInsideMultiLineComment = 0;
                 }
                 continue;
             }           You, 8 minutes ago • lab 2 over
             if (strstr(line, "/*") != NULL) {
                 printf("Multi-line comment: %s", strstr(line, "/*"));
                 isInsideMultiLineComment = 1;
                 if (strstr(line, "*/") != NULL) {
                     isInsideMultiLineComment = 0;
                 }
                 continue;
             }
             if (strstr(line, "//") != NULL) {
                 printf("Single-line comment: %s\n", strstr(line, "//") + 2);
                 continue;
             }
             printf("Line %d is not a comment: %s", lineNumber, line);
         }
         fclose(file);
         return 0;
     }
```

Examples on which this code was tested on:

Result:

To view the code files , refer github

https://github.com/arnavjain2710/Compiller-Techniques/tree/main/LAB%202