ACD Lab 4

By: Chikkeri Chinmaya
211IT017

1) Code:

```
%{
#include <stdio.h>
int keyword_count = 0;
int identifier_count = 0;
int constant_count = 0;
int operator_count = 0;
int punctuation_count = 0;
%}

digit       [0-9]
letter      [a-zA-Z]
id          ({letter}({letter}|{digit})*)
constant    ({digit}+)

%%

"if"            { keyword_count++; }
"else"          { keyword_count++; }
"while"         { keyword_count++; }
"for"           { keyword_count++; }
"return"        { keyword_count++; }
"int"           { keyword_count++; }
"float"         { keyword_count++; }
"double"        { keyword_count++; }
"char"          { keyword_count++; }
"bool"          { keyword_count++; }
{id}            { identifier_count++; }
{constant}      { constant_count++; }
[+\-*/%=]       { operator_count++; }
```

```
[(){}\[\];,]    { punctuation_count++; }

%%

int main(int argc, char **argv) {
    if (argc != 2) {
        printf("Usage: %s input_file\n", argv[0]);
        return 1;
    }
    FILE *fp = fopen(argv[1], "r");
    if (!fp) {
        printf("Error: cannot open file '%s'\n", argv[1]);
        return 1;
    }
    yyin = fp;
    yylex();
    fclose(fp);
    printf("number of keywords: %d\n", keyword_count);
    printf("number of identifiers: %d\n", identifier_count);
    printf("number of constants: %d\n", constant_count);
    printf("number of operators: %d\n", operator_count);
    printf("number of punctuations: %d\n", punctuation_count);
    printf("total number of tokens: %d\n", keyword_count +
identifier_count + constant_count + operator_count +
punctuation_count);
    return 0;
}
```

p1.l inputfile.txt ×

inputfile.txt

```
1    int float a1 25 b hello 1b 56
```

```
vislinux@LAPTOP-KH7OSAJQ:~/LEX PROGRAMS/Lab4/Problem1$ ./a.out a.txt
Number of keywords: 2
Number of identifiers: 3
Number of numbers: 2
Number of operators: 0
Number of punctuations: 0
Total number of tokens: 7
```

2)

Code :

```
%{
#include <stdio.h>
#include <string.h>
#include <stdlib.h> // Required for malloc
%}

%option noyywrap

%%

[a-zA-Z]   { /* do nothing */ }
[\n]       { /* do nothing */ }
.           { printf("-1\n"); }

%%

int main(int argc, char **argv) {
    int N = 0, count = 0;
    char *K = NULL, *comment = NULL;

    scanf("%d", &N);
    getchar(); // This consumes the newline character left in the
buffer from previous scanf()
```

```
    K = (char*)malloc(11 * sizeof(char)); // Allocate memory for
K

    if(K == NULL) { // Check that the allocation was successful
        printf("Memory allocation failed for K");
        return -1;
    }

    fgets(K, 11, stdin); // Use fgets to read input, this ensures
we don't read more characters than we're expecting
    K[strcspn(K, "\n")] = 0; // This removes the newline
character from the end of the string

    for (int i = 0; i < N; i++) {
        comment = (char*)malloc(256 * sizeof(char)); // Allocate
memory for comment

        if(comment == NULL) { // Check that the allocation was
successful
            printf("Memory allocation failed for comment");
            return -1;
        }

        fgets(comment, 256, stdin); // Use fgets to read input, this
ensures we don't read more characters than we're expecting
        comment[strcspn(comment, "\n")] = 0; // This removes
the newline character from the end of the string

         int i=0;
         while(comment[i]!='\0')
         {

if(comment[i]=='#'||comment[i]==','||comment[i]=='@'||comment
[i]=='!'||comment[i]=='%'||comment[i]=='&')
```

```
            {
            printf("-1\n");
            return 0;
            }
            i++;
        }

        if (strstr(comment, K) != NULL) {
            count++;
        }

        free(comment); // Deallocate the memory used by
comment
    }

    printf("%d\n", count);

    free(K); // Deallocate the memory used by K

    return 0;
}
```

```
vislinux@LAPTOP-KH7OSAJQ:~/LEX PROGRAMS/Lab4/P2$ gcc lex.yy.c -ll
vislinux@LAPTOP-KH7OSAJQ:~/LEX PROGRAMS/Lab4/P2$ ./a.out
2
good
The_video_is_good
Informative
1
```

```
vislinux@LAPTOP-KH7OSAJQ:~/LEX PROGRAMS/Lab4/P2$ ./a.out
4
helpful
Most_expensive_topic_now_a_days
It_was_really_helpful
This_is_very_helpful_video
Productive_talk
2
```

```
vislinux@LAPTOP-KH7OSAJQ:~/LEX PROGRAMS/Lab4/P2$ ./a.out
2
usefull
#Most_wanted_and_usefull_video
Thanks a lot...
-1
vislinux@LAPTOP-KH7OSAJQ:~/LEX PROGRAMS/Lab4/P2$
```