Name: Spandan Mukherjee

Experiment 1

Implementation of Token Seperation (Lexical Analyzer)

Code 1

To identify the tokens with the help of c program

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

int isDatatype(char *str) {
    if (strcmp(str, "int") == 0 || strcmp(str, "float") == 0 ||
        strcmp(str, "double") == 0 || strcmp(str, "char") == 0 ||
        strcmp(str, "long") == 0 || strcmp(str, "short") == 0
        ) {
        return 1;
    }
    return 0;
}



int isValid(char *str) {
    if (strcmp(str, "{") == 0 || strcmp(str, "}") == 0 || strcmp(str,",") == 0 ||
        strcmp(str, ";") == 0 || strcmp(str, "[") == 0 || strcmp(str,"]") == 0 ||
        strcmp(str, "(") == 0 || strcmp(str, ")") == 0 || strcmp(str,"]") == 0
        ) {
        return 1;
    }
    return 0;
}




int isOperator(char *str) {
    if (strcmp(str, "+") == 0 || strcmp(str, "-") == 0 ||
        strcmp(str, "*") == 0 || strcmp(str, "/") == 0 || strcmp(str,"==") == 0 ||
        strcmp(str, "=") == 0
        ) {
```

```
        return 1;
    }
    return 0;
}

int isKeyword(char *str) {
    if (strcmp(str, "if") == 0 || strcmp(str, "else") == 0 ||
        strcmp(str, "while") == 0 || strcmp(str, "for") == 0 || strcmp(str, "main") == 0 ||
        strcmp(str, "return") == 0 ||
        strcmp(str, "switch") == 0 || strcmp(str, "typedef") == 0 ||
        strcmp(str, "struct") == 0 || strcmp(str, "static") == 0 ||
        strcmp(str, "goto") == 0 || strcmp(str, "sizeof") == 0 ||
        strcmp(str, "break") == 0 || strcmp(str, "continue") == 0) {
        return 1;
    }
    return 0;
}

int isInteger(char *str) {
    int i;
    for (i = 0; i < strlen(str); i++) {
        if (!isdigit(str[i])) {
            return 0;
        }
    }
    return 1;
}

int main() {

    FILE *fp = fopen("test1.txt", "r");
    if (fp == NULL) {
        printf("Error opening file\n");
        return 1;
    }

    char line[10000];
    int a1,a2,a3,a4,a5,a6;
    while (fgets(line, sizeof(line), fp)) {
        char *token = strtok(line, " ");
        while (token) {
            if (isDatatype(token)) {
                a1++;
                printf("Data type: %s\n", token);
            } else if (isOperator(token)) {
                a2++;
                printf("Operator: %s\n", token);
            } else if (isKeyword(token)) {
                a3++;
                printf("Keyword: %s\n", token);
            } else if (isInteger(token)) {
                a4++;
```

```
            printf("Integer: %s\n", token);
        }
        else if (isValid(token)) {
            a5++;
            printf("Delimiters: %s\n", token);
        }
        else {
            a6++;
            // printf("Identifier: %s\n", token);
        }
        token = strtok(NULL, " ");
    }

  }
  printf("\nTotal tokens are: \n");

  printf("No.of Data type: %d\n", a1);
  printf("No.of Operator: %d\n", a2);
  printf("No.of Keyword: %d\n", a3);
  printf("No.of Integer: %d\n", a4);
  // printf("No.of Identifier: %d\n", a6);
  printf("No.of Delimiter: %d\n", a5);


  fclose(fp);
  return 0;
}
```

2nd File:

Code:

```
#include <stdio.h>
int main()
{
        int num1;
        scanf ( %d , &num1 ) ;
        for (int i = 1 ; i < num1 ; i++ )
        {
                if ( i % 2 == 0 )
                { printf ( i ) ;
                }
        }
return 0;
}
```

Image:

Open test1.txt ~/ Save

```c
#include <stdio.h>
int main()
{
        int num1;
        scanf ( %d , &num1 ) ;
        for (int i = 1 ; i < num1 ; i++ )
        {
                if ( i % 2 == 0 )
                { printf ( i ) ;
                }
        }
return 0;
}
```

Output:

```
spandan@spandan-VirtualBox: ~
spandan@spandan-VirtualBox:~$ gedit cdesign3.c
^C
spandan@spandan-VirtualBox:~$ gcc cdesign3.c
spandan@spandan-VirtualBox:~$ ./a.out
Data type: int
Delimiters: {
Delimiters: (
Delimiters: ,
Delimiters: )
Delimiters: ;
Operator: =
Integer: 1
Delimiters: ;
Delimiters: ;
Delimiters: (
Integer: 2
Operator: ==
Integer: 0
Delimiters: (
Delimiters: )
Keyword: return

Total tokens are:
No.of Data type: 1
No.of Operator: 2
No.of Keyword: 1
No.of Integer: 3
No.of Delimiter: 10
spandan@spandan-VirtualBox:~$ gedit cdesign3.c
```

Code 2:

To write a c program to use switch case, statement to perform addition, substraction, multiplication and division. Then the c program to identify as identifier, keyword, symbols etc.

Code:

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

int isDatatype(char *str) {
    if (strcmp(str, "int") == 0 || strcmp(str, "float") == 0 ||
        strcmp(str, "double") == 0 || strcmp(str, "char") == 0 ||
        strcmp(str, "long") == 0 || strcmp(str, "short") == 0
        ) {
        return 1;
    }
    return 0;
}



int isValid(char *str) {
    if (strcmp(str, "{") == 0 || strcmp(str, "}") == 0 || strcmp(str,",") == 0 ||
        strcmp(str, ";") == 0 || strcmp(str, "[") == 0 || strcmp(str,"]") == 0 ||
        strcmp(str, "(") == 0 || strcmp(str, ")") == 0 || strcmp(str,"]") == 0
        ) {
        return 1;
    }
    return 0;
}




int isOperator(char *str) {
    if (strcmp(str, "+") == 0 || strcmp(str, "-") == 0 ||
        strcmp(str, "*") == 0 || strcmp(str, "/") == 0 || strcmp(str,"==") == 0 ||
        strcmp(str, "=") == 0
        ) {
        return 1;
    }
    return 0;
}

int isKeyword(char *str) {
    if (strcmp(str, "if") == 0 || strcmp(str, "else") == 0 ||
        strcmp(str, "while") == 0 || strcmp(str, "for") == 0 || strcmp(str, "main") == 0 ||
```

```
        strcmp(str, "return") == 0 ||
        strcmp(str, "switch") == 0 || strcmp(str, "typedef") == 0 ||
        strcmp(str, "struct") == 0 || strcmp(str, "static") == 0 ||
        strcmp(str, "goto") == 0 || strcmp(str, "sizeof") == 0 ||
        strcmp(str, "break") == 0 || strcmp(str, "continue") == 0) {
        return 1;
    }
    return 0;
}

int isInteger(char *str) {
    int i;
    for (i = 0; i < strlen(str); i++) {
        if (!isdigit(str[i])) {
            return 0;
        }
    }
    return 1;
}

int main() {

    FILE *fp = fopen("test2.c", "r");
    if (fp == NULL) {
        printf("Error opening file\n");
        return 1;
    }

    char line[10000];
    int a1,a2,a3,a4,a5,a6;
    while (fgets(line, sizeof(line), fp)) {
        char *token = strtok(line, " ");
        while (token) {
            if (isDatatype(token)) {
                a1++;
                printf("Data type: %s\n", token);
            } else if (isOperator(token)) {
                a2++;
                printf("Operator: %s\n", token);
            } else if (isKeyword(token)) {
                a3++;
                printf("Keyword: %s\n", token);
            } else if (isInteger(token)) {
                a4++;
                printf("Integer: %s\n", token);
            }
            else if (isValid(token)) {
                a5++;
                printf("Delimiters: %s\n", token);
            }
            else {
                a6++;
```

```
            // printf("Identifier: %s\n", token);
          }
          token = strtok(NULL, " ");
        }

    }
    printf("\nTotal tokens are: \n");

    printf("No.of Data type: %d\n", a1);
    printf("No.of Operator: %d\n", a2);
    printf("No.of Keyword: %d\n", a3);
    printf("No.of Integer: %d\n", a4);
    // printf("No.of Identifier: %d\n", a6);
    printf("No.of Delimiter: %d\n", a5);


    fclose(fp);
    return 0;
}
```

2nd Code

```
#include <stdio.h>

int main() {

  char op;
  int first, second;
  printf("Enter an operator (+, -, *, /): ");
  scanf("%c", &op);
  printf("Enter two operands: ");
  scanf("%lf %lf", &first, &second);

  switch (op)
  {

    case '+':
      printf("%d + %d = %d", first, second, first + second);
      break;
    case '-':
      printf("%d - %d = %d", first, second, first - second);
      break;
    case '*':
      printf("%d * %d = %d", first, second, first * second);
      break;
    case '/':
      printf("%d / %d = %d", first, second, first / second);
      break;

    default:
      printf("Error! operator is not correct");
```

```
  }

  return 0;
}
```

OUTPUT: