Parul University

FACULTY OF ENGINEERING AND TECHNOLOGY

Compiler Design Laboratory (303105350)

6th SEMESTER (3rd YEAR)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

Laboratory Manual

PARUL UNIVERSITY

FACULTY OF ENGINEERING & TECHNOLOGY
PARUL INSTITUTE OF ENGINEERING & TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

STUDENT DETAILS

NAME	:	THAKAR AARYA RUSHIKESH
ENROLLMENT NO	:	2303031050636
SUBJECT	:	COMPILER DESIGN LABORATORY
SUBJECT CODE	:	303105350
SEMESTER	:	6TH

Parul University

PARUL UNIVERSITY

FACULTY OF ENGINEERING AND TECHNOLOGY

BACHELOR OF TECHNOLOGY

CERTIFICATE

This is to certify that MR THAKAR AARYA RUSHIKESH with Enrollment no. 2303031050636 has Successfully completed his/her Laboratory Experiments in the Compiler Design (203105350) from the Department of Computer Science and Engineering during the academic year 2025-2026.

Date of Submission: ....

Staff In charge: .... **Head of Department: ....**

INDEX

10	Implement following programs using Lex. a. Write a Lex program to print out all numbers from the given file. b. Write a Lex program to printout all HTML tags in file. c. Write a Lex program which adds line numbers to the given file and display the same onto the standard output.						

Practical - 1

Aim:- Project Definition and objective of the specified module and perform requirement engineering process.

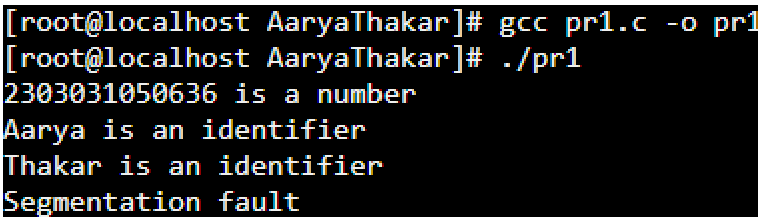
main.c

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

void keyw(char *p);

int i = 0, id = 0, kw = 0, num = 0, op = 0, sp = 0, ar = 0, count = 0, new_line = 0;

char keys[32][10] = {
    "auto","break","case","char","const","continue","default","do",
    "double","else","enum","extern","float","for","goto","if","int",
    "long","register","return","short","signed","sizeof","static",
    "struct","switch","typedef","union","unsigned","void","volatile","while"
};

int main()
{
    char ch, str[25];
    char seps[20] = " \t\n,;(){}[]#\"<>";
    char oper[] = "!%^&*-+=~|.<>/?";
    int j;

    FILE *f1 = fopen("sample1.txt", "r");
    if (f1 == NULL)
    {
        printf("File not found\n");
        exit(0);
    }

    while ((ch = fgetc(f1)) != EOF)
    {
        for (j = 0; j <= 14; j++)
        {
            if (ch == oper[j])
            {
                printf("%c is an operator\n", ch);
                op++;
                count++;

                str[i] = '\0';
                keyw(str);
            }
        }
        if (ch == '\n')
            new_line++;
        for (j = 0; j <= 14; j++)
        {
            if (i == -1)
                break;

            if (ch == seps[j])
            {
                if (ch == '#')
                {
                    while (ch != '>')
                    {
                        printf("%c", ch);
                        ch = fgetc(f1);
                    }
                    printf("%c is a header file\n", ch);

                    i = -1;
                    break;
                }
                if (ch == '"')
                {
                    do {
                        ch = fgetc(f1);
                        printf("%c", ch);
                    } while (ch != '"');

                    printf("\b is an argument\n");
                    ar++;
                    count++;
                    i = -1;
                    break;
                }
                if (ch == ',' || ch == ';' || ch == '(' || ch == ')' ||
                    ch == '{' || ch == '}' || ch == '[' || ch == ']')
                {
                    printf("%c is a Separator\n", ch);
                    sp++;
                    count++;
                }

                str[i] = '\0';
                keyw(str);
            }
        }
        if (i != -1)
        {
            str[i] = ch;
            i++;
        }
        else
            i = 0;
    }
    printf("\nKeywords: %d", kw);
    printf("\nIdentifiers: %d", id);
    printf("\nOperators: %d", op);
    printf("\nNumbers: %d", num);
    printf("\nSeparators: %d", sp);
    printf("\nArguments: %d", ar);
    printf("\nTotal Tokens: %d", count);
    printf("\nNumber of Lines: %d\n", new_line);

    return 0;
}
void keyw(char *p)
{
    int k, flag = 0;
    for (k = 0; k <= 31; k++)
    {
        if (strcmp(keys[k], p) == 0)
        {
            printf("%s is a keyword\n", p);
            kw++;
            count++;
            flag = 1;
            break;
        }
    }

    if (flag == 0)
    {
        if (isdigit(p[0]))
        {
            printf("%s is a number\n", p);
            num++;
            count++;
        }
        else
        {
            if (p[0] != '\0')
            {
                printf("%s is an identifier\n", p);
                id++;
                count++;
            }
        }
    }

    i = -1;
}
```

Sample1.txt

2303031050636 Aarya Thakar

Output:



```
[root@localhost AaryaThakar]# gcc pr1.c -o pr1
[root@localhost AaryaThakar]# ./pr1
2303031050636 is a number
Aarya is an identifier
Thakar is an identifier
Segmentation fault
```

Practical - 2

Aim:- Program to count digits, vowels and symbols in C.

main.c

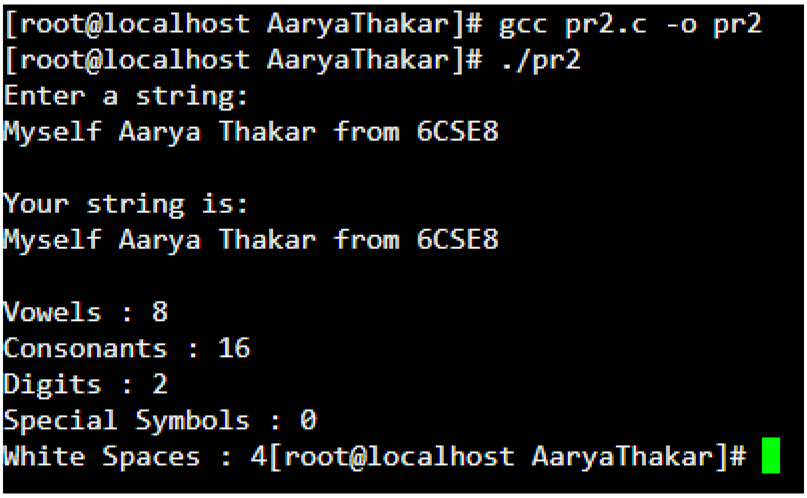
```
#include <stdio.h>
#include <string.h>

int main()
{
   char str[100];
   int i;
   int vowels = 0, consonant = 0, digit = 0, symbols = 0, spaces = 0;

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

   printf("\nYour string is:\n");
   printf("%s", str);

   for (i = 0; str[i] != '\0'; i++)
   {
      if (str[i] == 'a' || str[i] == 'A' ||
         str[i] == 'e' || str[i] == 'E' ||
         str[i] == 'i' || str[i] == 'I' ||
         str[i] == 'o' || str[i] == 'O' ||
         str[i] == 'u' || str[i] == 'U')
      {
         vowels++;
      }
      else if ((str[i] >= 'a' && str[i] <= 'z') ||
              (str[i] >= 'A' && str[i] <= 'Z'))
      {
         consonant++;
      }
      else if (str[i] >= '0' && str[i] <= '9')
      {
         digit++;
      }
      else if (str[i] == ' ')
      {
         spaces++;
      }
      else if (str[i] != '\n')   // ignore newline from fgets
      {
         symbols++;
      }
   }

   printf("\nVowels : %d", vowels);
   printf("\nConsonants : %d", consonant);
   printf("\nDigits : %d", digit);
```

```
    printf("\nSpecial Symbols : %d", symbols);
    printf("\nWhite Spaces : %d", spaces);

    return 0;
}
```

input.txt

MySelf Aarya Thakar from 6CSE8

Output:



```
[root@localhost AaryaThakar]# gcc pr2.c -o pr2
[root@localhost AaryaThakar]# ./pr2
Enter a string:
Myself Aarya Thakar from 6CSE8

Your string is:
Myself Aarya Thakar from 6CSE8

Vowels : 8
Consonants : 16
Digits : 2
Special Symbols : 0
White Spaces : 4[root@localhost AaryaThakar]#
```

Practical - 3

Aim:- Program to check validation of User Name and Password in C.

main.c

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int main() {
    char username[25], password[20];
    int i, flag;
    FILE *fp, *fp1;
    fp = fopen("user.txt", "r");
    if (fp == NULL) {
        printf("User file not opened\n");
        exit(1);
    }
    fgets(username, 25, fp);
    username[strcspn(username, "\n")] = '\0';  // remove newline
    fclose(fp);
    printf("Your Username is: %s\n", username);
    if (strcmp(username, "AaryaThakar") == 0 ||
        strcmp(username, "ThakarAarya") == 0 ||
        strcmp(username, "AaryaRThakar") == 0 ||
        strcmp(username, "RAaryaThakar") == 0) {
        printf("This Username already exists.\n");
        return 0;
    }
    if (username[0] < 'A' || username[0] > 'Z') {
        printf("First character must be capital.\n");
        return 0;
    }
    for (i = 0; username[i] != '\0'; i++) {
        if (username[i] >= '0' && username[i] <= '9') {
            printf("Digits are not allowed in username.\n");
            return 0;
        }
    }
    for (i = 0; username[i] != '\0'; i++) {
        if (strchr("~!@#$%^&*", username[i])) {
            printf("Special symbols not allowed in username.\n");
            return 0;
        }
    }
    fp1 = fopen("pass.txt", "r");
    if (fp1 == NULL) {
        printf("Password file not opened\n");
        exit(1);
    }
    fgets(password, 20, fp1);
    password[strcspn(password, "\n")] = '\0';
    fclose(fp1);
    printf("Your Password is: %s\n", password);
    if (strlen(password) < 8 || strlen(password) > 15) {
        printf("Password length must be 8 to 15 characters.\n");
        return 0;
    }
    flag = 0;
    for (i = 0; password[i] != '\0'; i++) {
        if (password[i] >= '0' && password[i] <= '9') {
            flag = 1;
            break;
        }
    }
    if (!flag) {
        printf("Password must contain at least one digit.\n");
        return 0;
    }
    flag = 0;
    for (i = 0; password[i] != '\0'; i++) {
        if (strchr("~!@#$%^&*", password[i])) {
            flag = 1;
            break;
        }
    }
    if (!flag) {
        printf("Password must contain at least one special symbol.\n");
        return 0;
    }
    for (i = 0; password[i] != '\0'; i++) {
        if (password[i] == ' ') {
            printf("Password should not contain spaces.\n");
            return 0;
        }
    }
    printf("\nUsername and Password created successfully.");
    return 0;
}
```

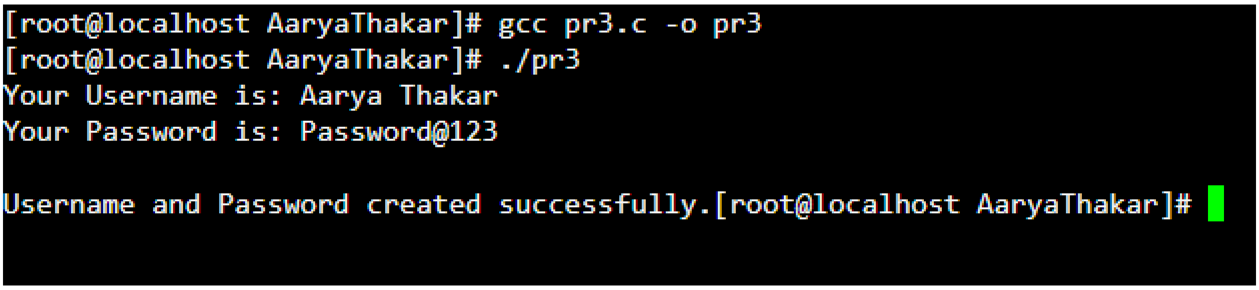
user.txt

Aarya Thakar

pass.txt

Password@123

Output



```
[root@localhost AaryaThakar]# gcc pr3.c -o pr3
[root@localhost AaryaThakar]# ./pr3
Your Username is: Aarya Thakar
Your Password is: Password@123

Username and Password created successfully.[root@localhost AaryaThakar]#
```