Project Report on

Compiler for << Finding Area and Parimeter of Circle and Square Using Gujarati Language>>

Developed by

IT035 GOHIL SHASHANK BHIKHUBHAI 19ITUBS058

IT036 GONDALIYA ZEEL BHAGVANBHAI 19ITUBS153

IT037 GORASIYA KASHYAP KAMLESHBHAI 19ITUEF014

Guided By:

Prof. Nikita P. Desai

Dept. of Information Technology

Department of Information Technology
Faculty of Technology, Dharmsinh Desai University
College Road, Nadiad-387001 2020-2021

DHARMSINH DESAI UNIVERSITY
NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled "**Compiler for Finding Area and Parimeter of Circle and Square Using Gujarati language"** is a bonafied report of the work carried out by

1) Mr. KASHYAP GORASIYA, Student ID No: 19ITUEF014
2) Mr. ZEEL GONDALIYA, Student ID No: 19ITUBS153
3) Mr. SHASHANK SHASHANK Student ID No: 19ITUBS058

of Department of Information Technology, semester VI, under the guidance and supervision for the award of the degree of Bachelor of Technology at Dharmsinh Desai University, Nadiad (Gujarat). They were involved in Project in subject of "**Language Translator**" during academic year 2021-2022.

Prof. N.P. Desai
(Lab Incharge)
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad Date:

Prof. (Dr.)V K Dabhi,
Head , Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad Date:

Index

1.0 INTRODUCTION

1.0.1 Project Details

Language Name: Compiler for Layman friendly Finding Area and Parimeter of Circle and Square Program.

description:

1.0.2 Project Planning

- Valid sentences in language :

  Vartul ni trijya 5 meter chhe , vartul nu xetrafal shu thase?

  Vartul ni trijya 5 meter chhe , vartul ni parimiti shu thase?

  Choras ni lambai 5 meter chhe, choras nu xetrafal shu thase?

  Choras ni lambai 5 meter chhe, choras nu xetrafal shu thase?

List of Students with their Roles/Responsibilities:

Regular Expression , DFA Design,
Algorithm Design and implmentation, Scanner phase Implementation, Grammar rules, YACC implementation, Final Report.

2.0 LEXICAL PHASE DESIGN

2.0.1 Regular Expression:

Keywords:

vartul

chhe

shu

choras

thase

ni trijya

ni lambai

Operators:

ni parimiti	+
nu xetrafal	*

units:

meter

meters

centimeter

centimeters

squaremeter

squarecentimeter

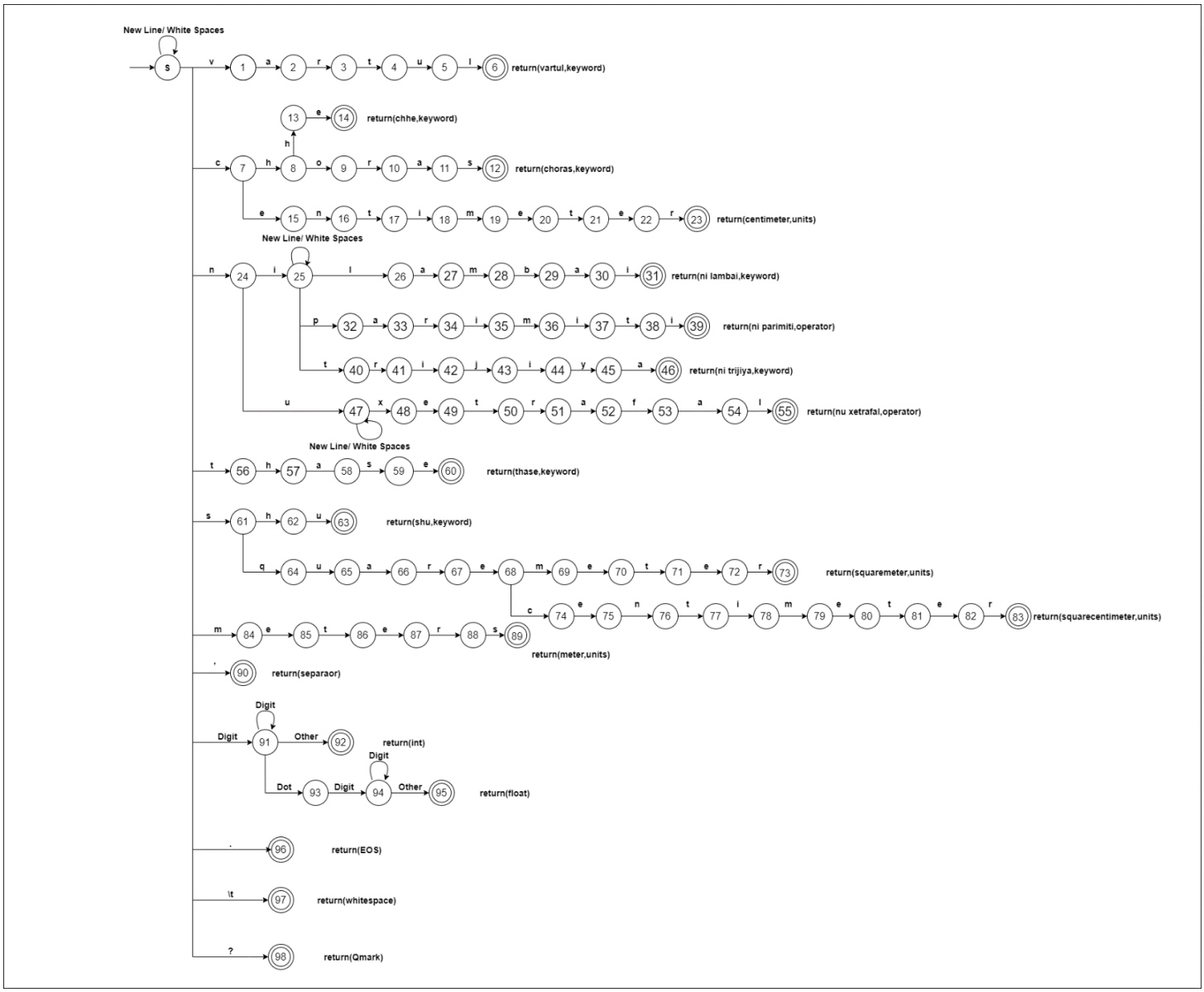
digit:

[0-9]+	int
[0-9]+(. [0-9]+)	float

Punchuation :

Qmark:	?	q
NewLine:	[\n]	n
Eos	.	e
Separater	,	s
White space	[/t]	w

2.0.2 Deterministic Finite Automata design for lexer



2.0.3 Algorithm of lexer

ALGORITHM:

```
lexer
{
    int c = 0;
    bool f = false;
    int len = string.length();
    while not eof do
    {
        state="S";
        while not eof do (c < len)
        {
            if (f)
            {
                f= false;
            }

            char ch = nextchar();

            switch (state)
            {

                case state of "S": {
                    case state of
                    'v':
                        state = "1";
                        ch = nextchar();
                        break;

                    'c':
                        state = "7";
                        ch = nextchar();
                        break;

                    'n':
                        state = "24";
                        ch = nextchar();
                        break;

                    't':
                        state = "56";
                        ch = nextchar();
                        break;
```

```
    's':
        state = "61";
        ch = nextchar();
        break;

    'm':
        state = "84";
        ch = nextchar();
        break;

    ',':
        state = "90";
        ch = nextchar();
        f = true;
        break;

    [0-9]:
        state = "91";
        ch = nextchar();
        f = true;
        break;

    '.':
        state = "96";
        ch = nextchar();
        f = true;
        break;

    '\t':
        state = "97";
        ch = nextchar();
        break;

    '?':
        state = "98";
        ch = nextchar();
        f = true;
        break;

    Default:
        f = true;

    end case
}
case state of "1":
{
    case state of 'a':
        state = "2";
        ch = nextchar();
        break;
```

```
}

case state of "2":
{
    case state of 'r':
        state = "3";
        ch = nextchar();
        break;
}

case state of "3":
{
    case state of 't':
        state = "4";
        ch = nextchar();
        break;
}

case state of "4":
{
    case state of 'u':
        state = "5";
        ch = nextchar();
        break;
}

case state of "5":
{
    case state of 'l':
    state = "6";
    ch = nextchar();
    f = true;
    break;
}



case state of "7":
{
    case state of 'h':
    state = "8";
    ch = nextchar();
    break;

    case state of 'e':
    state = "15";
    ch = nextchar();
    break;
```

```
    'Default':
    f=true;

}

case state of "8":
{
    case state of 'o':
    state = "9";
    ch = nextchar();
    break;

    case state of 'h':
    state = "13";
    ch = nextchar();
    break;


    'Default':
    f=true;

}

case state of "9":
{
    case state of 'r':
    state = "10";
    ch = nextchar();
    break;
}

case state of "10":
{
    case state of 'a':
    state = "11";
    ch = nextchar();
    break;
}

case state of "11":
{
    case state of 's':
    state = "12";
    ch = nextchar();
    f = true;
    break;
}

case state of "13":
{
```

```
    case state of 'e':
    state = "14";
    ch = nextchar();
    f = true;
    break;
}

case state of "15":
{
    case state of 'n':
    state = "16";
    ch = nextchar();
    break;
}

case state of "16":
{
    case state of 't':
    state = "17";
    ch = nextchar();
    break;
}

case state of "17":
{
    case state of 'i':
    state = "18";
    ch = nextchar();
    break;
}

case state of "18":
{
    case state of 'm':
    state = "19";
    ch = nextchar();
    break;
}

case state of "19":
{
    case state of 'e':
    state = "20";
    ch = nextchar();
    break;
}

case state of "20":
{
    case state of 't':
```

```
    state = "21";
    ch = nextchar();
    break;
}

case state of "21":
{
    case state of 'e':
    state = "22";
    ch = nextchar();
    break;
}

case state of "22":
{
    case state of 'r':
    state = "23";
    ch = nextchar();
    f = true;
    break;
}

case state of "24":
{
    case state of 'i':
    state = "25";
    ch = nextchar();
    break;

    case state of 'u':
    state = "47";
    ch = nextchar();
    break;


    'Default':
    f=true;

}

case state of "25":
{
    case state of '\t':
    state = "25";
    ch = nextchar();
    break;

    case state of 'l':
    state = "26";
    ch = nextchar();
```

```
    break;

    case state of 'p':
    state = "32";
    ch = nextchar();
    break;

    case state of 't':
    state = "40";
    ch = nextchar();
    break;


    'Default':
    f=true;

}



case state of "26":
{
    case state of 'a':
    state = "27";
    ch = nextchar();
    break;
}
case state of "27":
{
    case state of 'm':
    state = "28";
    ch = nextchar();
    break;
}
case state of "28":
{
    case state of 'b':
    state = "29";
    ch = nextchar();
    break;
}
case state of "29":
{
    case state of 'a':
    state = "30";
    ch = nextchar();
    break;
}
case state of "30":
{
```

```
    case state of 'i':
    state = "31";
    ch = nextchar();
    f = true;
    break;
}

case state of "32":
{
    case state of 'a':
    state = "33";
    ch = nextchar();
    break;
}
case state of "33":
{
    case state of 'r':
    state = "34";
    ch = nextchar();
    break;
}
case state of "34":
{
    case state of 'i':
    state = "35";
    ch = nextchar();
    break;
}
case state of "35":
{
    case state of 'm':
    state = "36";
    ch = nextchar();
    break;
}
case state of "36":
{
    case state of 'i':
    state = "37";
    ch = nextchar();
    break;
}
case state of "37":
{
    case state of 't':
    state = "38";
    ch = nextchar();
    break;
}
case state of "38":
```

```
{
    case state of 'i':
    state = "39";
    ch = nextchar();
    f=true;
    break;
}
case state of "38":
{
    case state of 'i':
    state = "39";
    ch = nextchar();
    f=true;
    break;
}

case state of "40":
{
    case state of 'r':
    state = "41";
    ch = nextchar();
    break;
}
case state of "41":
{
    case state of 'i':
    state = "42";
    ch = nextchar();
    break;
}
case state of "42":
{
    case state of 'j':
    state = "43";
    ch = nextchar();
    break;
}
case state of "43":
{
    case state of 'i':
    state = "44";
    ch = nextchar();
    break;
}
case state of "44":
{
    case state of 'y':
    state = "45";
    ch = nextchar();
    break;
```

```
}
case state of "45":
{
    case state of 'a':
    state = "46";
    ch = nextchar();
    f=true;
    break;
}

case state of "47":
{
    case state of '\t':
    state = "47";
    ch = nextchar();
    break;

    case state of 'x':
    state = "48";
    ch = nextchar();
    break;
}
case state of "48":
{
    case state of 'e':
    state = "49";
    ch = nextchar();
    break;
}
case state of "49":
{
    case state of 't':
    state = "50";
    ch = nextchar();
    break;
}
case state of "50":
{
    case state of 'r':
    state = "51";
    ch = nextchar();
    break;
}
case state of "51":
{
    case state of 'a':
    state = "52";
    ch = nextchar();
    break;
}
```

```
case state of "52":
{
    case state of 'f':
    state = "53";
    ch = nextchar();
    break;
}
case state of "53":
{
    case state of 'a':
    state = "54";
    ch = nextchar();
    break;
}
case state of "54":
{
    case state of 'l':
    state = "55";
    ch = nextchar();
    f=true;
    break;
}

case state of "56":
{
    case state of 'h':
    state = "57";
    ch = nextchar();
    break;
}
case state of "57":
{
    case state of 'a':
    state = "58";
    ch = nextchar();
    break;
}
case state of "58":
{
    case state of 's':
    state = "59";
    ch = nextchar();
    break;
}
case state of "59":
{
    case state of 'e':
    state = "60";
    ch = nextchar();
    f=true;
```

```
    break;
}


case state of "61":
{
    case state of 'h':
    state = "62";
    ch = nextchar();
    break;

    case state of 'q':
    state = "64";
    ch = nextchar();
    break;

    'Default':
    f=true;

}

case state of "62":
{
    case state of 'u':
    state = "63";
    ch = nextchar();
    break;

    'Default':
    f=true;

}

case state of "64":
{
    case state of 'u':
    state = "65";
    ch = nextchar();
    break;
}
case state of "65":
{
    case state of 'a':
    state = "66";
    ch = nextchar();
    break;
}
case state of "66":
{
    case state of 'r':
```

```
    state = "67";
    ch = nextchar();
    break;
}
case state of "67":
{
    case state of 'e':
    state = "68";
    ch = nextchar();
    break;
}
case state of "68":
{
    case state of 'm':
    state = "69";
    ch = nextchar();
    break;

    case state of 'c':
    state = "74";
    ch = nextchar();
    break;

}
case state of "69":
{
    case state of 'e':
    state = "70";
    ch = nextchar();
    break;
}
case state of "70":
{
    case state of 't':
    state = "71";
    ch = nextchar();
    break;
}
case state of "71":
{
    case state of 'e':
    state = "72";
    ch = nextchar();
    break;
}
case state of "72":
{
    case state of 'r':
    state = "73";
    ch = nextchar();
```

```
    f=true;
    break;
}

case state of "62":
{
    case state of 'u':
    state = "63";
    ch = nextchar();
    break;

    'Default':
    f=true;

}

case state of "74":
{
    case state of 'e':
    state = "75";
    ch = nextchar();
    break;
}
case state of "75":
{
    case state of 'n':
    state = "76";
    ch = nextchar();
    break;
}
case state of "76":
{
    case state of 't':
    state = "77";
    ch = nextchar();
    break;
}
case state of "77":
{
    case state of 'i':
    state = "78";
    ch = nextchar();
    break;
}
case state of "78":
{
    case state of 'm':
    state = "79";
    ch = nextchar();
    break;
```

```
}
case state of "79":
{
    case state of 'e':
    state = "80";
    ch = nextchar();
    break;
}
case state of "80":
{
    case state of 't':
    state = "81";
    ch = nextchar();
    break;
}
case state of "81":
{
    case state of 'e':
    state = "82";
    ch = nextchar();
    break;
}
case state of "82":
{
    case state of 'r':
    state = "83";
    ch = nextchar();
    f=true;
    break;
}

case state of "78":
{
    case state of 'm':
    state = "79";
    ch = nextchar();
    break;

}
case state of "84":
{
    case state of 'e':
    state = "85";
    ch = nextchar();
    break;
}
case state of "85":
{
    case state of 't':
```

```
    state = "86";
    ch = nextchar();
    break;
}
case state of "86":
{
    case state of 'e':
    state = "87";
    ch = nextchar();
    break;
}
case state of "87":
{
    case state of 'r':
    state = "88";
    ch = nextchar();
    break;
}
case state of "88":
{
    case state of 's':
    state = "89";
    ch = nextchar();
    f=true;
    break;
}

case state of "91":
{
    case state of [0-9]:
    state = "91";
    ch = nextchar();
    break;

    '.':
    state = "93";
    ch = nextchar();
    break;

    default:
    state = "92";
    f = true;
}
case state of "93": {
    case state of
    [0-9]:
    state = "94";
    ch = nextchar();
    break;
```

```
                default:
                f = true;
            }
            case state of "94": {
                case state of
                [0-9]:
                ch = nextchar();

                default:
                state = "95";
                f = true;
            }




        }
        }
        case state of

        "6"|"14"|"12"|"23"|"31"|"39"|"46"|"55"|"60"|"63"|"73"|"83"|"89":
        print(" keyword");

        "90":
        print("separator");
        "92":
        print("Int");
        "95":
        print("Float");
        "96":
        print("eos");
        "97":
        print("WhiteSpace");
        "98":
        print("Qmark");
        default:
        print("invalid input");
        ch := nextchar();

        end case;
    }
}
```

2.0.4 Implementation of lexer

Flex Program:

ltproj.l

```
%{
#include<stdio.h>
int kwc=0;
int opc=0;
int unc=0;
int dgc=0;
int inc=0;
int flc=0;
int qmc=0;
int wsc=0;
int nlc=0;
int esc=0;
int spc=0;
int nvc=0;
%}

Keyword "vartul"|"choras"|"shu"|"chhe"|"thase"|"ni
trijya"|"ni lambai"
Operator "ni parimiti"|"nu xetrafal"
```

```
Unit
"meter"|"meters"|"centimeter"|"centimeters"|"squaremeter"|"sq
uarecentimeters"

Digit [0-9]

Integer {Digit}+

Float {Digit}+(.{Digit})

Qmark "?"

WhiteSpace " "

NewLine [\n]

Eos "."

Separater ","


%%
{Keyword} {printf("Valid Keyword-%s\n",yytext); kwc++;}

{Operator} {printf("Valid Operator-%s\n",yytext); opc++;}

{Unit} {printf("Valid Unit-%s\n",yytext); unc++;}

{Digit} {printf("Valid Digit-%s\n",yytext); dgc++;}

{Integer} {printf("Valid Integer-%s\n",yytext); inc++;}

{Float} {printf("Valid Float-%s\n",yytext); flc++;}

{Qmark} {printf("Valid Qmark-%s\n",yytext); qmc++;}

{WhiteSpace} {printf("Valid WhiteSpace-%s\n",yytext); wsc++;}

{NewLine} {printf("Valid NewLine-%s\n",yytext); nlc++;}

{Eos} {printf("Valid Eos-%s\n",yytext); esc++;}

{Separater} {printf("Valid Separater-%s\n",yytext); spc;}

. {printf("%s is Not valid token.\n",yytext); nvc++;}

end   {return 0;}

%%
```

```
int yywrap(){return 1;}

int main()
{
yylex();
printf("Number of Keyword %d\n",kwc);
printf("Number of Operator %d\n",opc);
printf("Number of Unit %d\n",unc);
printf("Number of Digit %d\n",dgc);
printf("Number of Integer %d\n",inc);
printf("Number of Float %d\n",flc);
printf("Number of Qmark %d\n",qmc);
printf("Number of WhiteSpace %d\n",wsc);
printf("Number of Eos %d\n",esc);
printf("Number of Separater %d\n",spc);
printf("Number of Non Valid Token %d\n",nvc);
return 0;
}
```

OUTPUT SCREENSHOT

```
kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT$ ./a.out
vartul ni trijya 5 meter chhe,vartul nu xetrafal shu thase ?
Valid Keyword-vartul
Valid WhiteSpace-
Valid Keyword-ni trijya
Valid WhiteSpace-
Valid Digit-5
Valid WhiteSpace-
Valid Unit-meter
Valid WhiteSpace-
Valid Keyword-chhe
Valid Separater-,
Valid Keyword-vartul
Valid WhiteSpace-
Valid Operator-nu xetrafal
Valid WhiteSpace-
Valid Keyword-shu
Valid WhiteSpace-
Valid Keyword-thase
Valid WhiteSpace-
Valid Qmark-?
Valid NewLine-

end
Number of Keyword 6
Number of Operator 1
Number of Unit 1
Number of Digit 1
Number of Integer 0
Number of Float 0
Number of Qmark 1
Number of WhiteSpace 8
Number of Eos 0
Number of Separater 0
Number of Non Valid Token 0
```

```
kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT$ ./a.out
vartul ni trijya 5 meter chhe , vartul ni parimiti shu thase?
Valid Keyword-vartul
Valid WhiteSpace-
Valid Keyword-ni trijya
Valid WhiteSpace-
Valid Digit-5
Valid WhiteSpace-
Valid Unit-meter
Valid WhiteSpace-
Valid Keyword-chhe
Valid WhiteSpace-
Valid Separater-,
Valid WhiteSpace-
Valid Keyword-vartul
Valid WhiteSpace-
Valid Operator-ni parimiti
Valid WhiteSpace-
Valid Keyword-shu
Valid WhiteSpace-
Valid Keyword-thase
Valid Qmark-?
Valid NewLine-

end
Number of Keyword 6
Number of Operator 1
Number of Unit 1
Number of Digit 1
Number of Integer 0
Number of Float 0
Number of Qmark 1
Number of WhiteSpace 9
Number of Eos 0
Number of Separater 0
```

```
kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT$ ./a.out
choras ni lambai 5 meter chhe, choras nu xetrafal shu thase?
Valid Keyword-choras
Valid WhiteSpace-
Valid Keyword-ni lambai
Valid WhiteSpace-
Valid Digit-5
Valid WhiteSpace-
Valid Unit-meter
Valid WhiteSpace-
Valid Keyword-chhe
Valid Separater-,
Valid WhiteSpace-
Valid Keyword-choras
Valid WhiteSpace-
Valid Operator-nu xetrafal
Valid WhiteSpace-
Valid Keyword-shu
Valid WhiteSpace-
Valid Keyword-thase
Valid Qmark-?
Valid NewLine-

end
Number of Keyword 6
Number of Operator 1
Number of Unit 1
Number of Digit 1
Number of Integer 0
Number of Float 0
Number of Qmark 1
Number of WhiteSpace 8
Number of Eos 0
Number of Separater 0
Number of Non Valid Token 0
```

```
kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT$ ./a.out
choras ni lambai 5 meter chhe, choras nu xetrafal shu thase?
Valid Keyword-choras
Valid WhiteSpace-
Valid Keyword-ni lambai
Valid WhiteSpace-
Valid Digit-5
Valid WhiteSpace-
Valid Unit-meter
Valid WhiteSpace-
Valid Keyword-chhe
Valid Separater-,
Valid WhiteSpace-
Valid Keyword-choras
Valid WhiteSpace-
Valid Operator-nu xetrafal
Valid WhiteSpace-
Valid Keyword-shu
Valid WhiteSpace-
Valid Keyword-thase
Valid Qmark-?
Valid NewLine-

end
Number of Keyword 6
Number of Operator 1
Number of Unit 1
Number of Digit 1
Number of Integer 0
Number of Float 0
Number of Qmark 1
Number of WhiteSpace 8
Number of Eos 0
Number of Separater 0
Number of Non Valid Token 0
```

2.0.7 Implementation of scanner using c language

CODE:

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

int errorPrint(char s[], int len, int initial, int final)
{
    char temp[100];
    for (int k = 0; k < final; k++)
    {
        temp[k] = s[k + initial];
    }
    printf("\nInvalid Input:\t\t\t%s", temp);
    return 0;
}

void scanner(char s[], int len)
{
    int i = 0;
    while (i < len)
    {

        if (s[i] == '?')
        {
            printf("\nQMark:\t\t\t?");
            i++;
        }

        else if (s[i] == '.')
        {
            printf("\nEOS:\t\t\t.");
            i++;
        }

        else if (s[i] == ' ')
        {
            printf("\nWhitespace:\t\t ");
            i++;
        }
        else if (s[i] == ',')
        {
            printf("\nseparator:\t\t,");
            i++;
        }

        else if (s[i] == 'c')///
        {
```

```
            int j = i;
            i++;

            if (s[i] == 'e')//
            {
                i++;
                if (s[i] == 'n' && s[i + 1] == 't' && s[i + 2] == 'i' && s[i +
3] == 'm' && s[i + 4] == 'e' && s[i + 5] == 't' && s[i + 6] == 'e' && s[i + 7]
== 'r')
                {
                    i += 8;


                    printf("\nKeyword:\t\tcentimeter");

                }
                else
                {
                    errorPrint(s, len, j, i);
                }
            }
            else if (s[i] == 'h')//
            {
                i++;

                if ( s[i] == 'h' && s[i+1] == 'e')
                {
                    printf("\nKeyword:\t\tchhe");
                    i += 2;
                }
                else if ( s[i] == 'o' && s[i+1] == 'r' && s[i+2] == 'a' &&
s[i+3] == 's'  )
                {
                    printf("\nKeyword:\t\tchoras");
                    i += 4;
                }
                else
                {
                    errorPrint(s, len, j, i);
                }
            }


        }

        else if (s[i] == 'n')///
        {
            int j = i;
            i++;
```

```
            if (s[i] == 'i')//
            {
                i++;
                if (s[i] == ' ' && s[i + 1] == 'l' && s[i + 2] == 'a' && s[i +
3] == 'm' && s[i + 4] == 'b' && s[i + 5] == 'a' && s[i + 6] == 'i')
                {
                    i += 7;


                    printf("\nKeyword:\t\tni lambai");

                }
                else if (s[i] == ' ' && s[i + 1] == 'p' && s[i + 2] == 'a' &&
s[i + 3] == 'r' && s[i + 4] == 'i' && s[i + 5] == 'm' && s[i + 6] == 'i' &&
s[i + 7] == 't'  && s[i + 8] == 'i')
                {
                    i += 9;


                    printf("\nKeyword:\t\tni parimiti");

                }
                else if (s[i] == ' ' && s[i + 1] == 't' && s[i + 2] == 'r' &&
s[i + 3] == 'i' && s[i + 4] == 'j' && s[i + 5] == 'i' && s[i + 6] == 'y' &&
s[i + 7] == 'a')
                {
                    i += 8;


                    printf("\nKeyword:\t\tni trijiya");

                }
                else
                {
                    errorPrint(s, len, j, i);
                }
            }
            else if (s[i] == 'u')//
            {
                i++;

                if ( s[i] == ' ' && s[i+1] == 'x' && s[i+2] == 'e' && s[i+3]
== 't'   && s[i+4] == 'r' && s[i+5] == 'a' && s[i+6] == 'f'  && s[i+7] == 'a'
&& s[i+8] == 'l'     )
                {
                    printf("\nKeyword:\t\tnu xetrafal");
                    i += 9;
                }
                else
                {
```

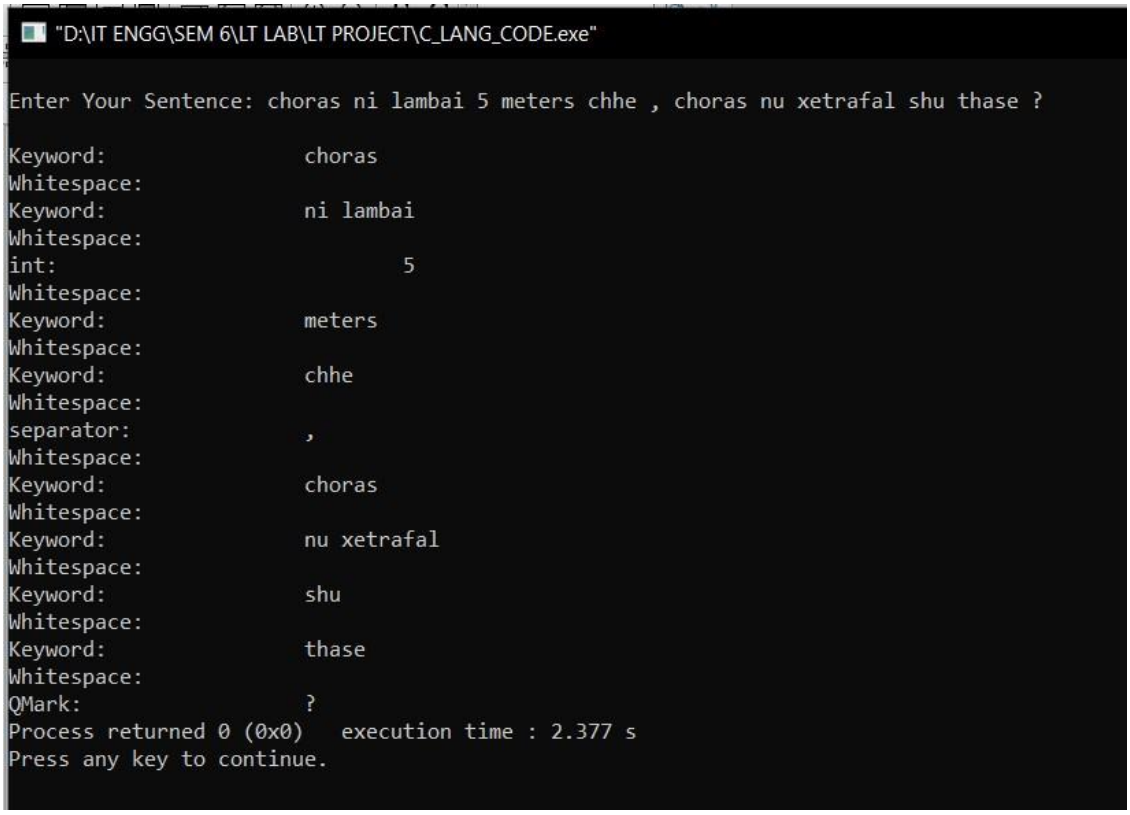
```
                errorPrint(s, len, j, i);
            }
        }


    }




    else if (s[i] == 'v')
    {
        int j = i;
        i++;
        if (s[i] == 'a' && s[i + 1] == 'r' && s[i + 2] == 't' && s[i + 3]
== 'u' && s[i + 4] == 'l')
        {
            i += 5;
            printf("\nKeyword:\t\tvartul");

        }
        else
        {
            errorPrint(s, len, j, i);
        }
    }



    else if (s[i] == 't')
    {
        int j = i;
        i++;
        if (s[i] == 'h' && s[i + 1] == 'a' && s[i + 2] == 's' && s[i + 3]
== 'e')
        {
            printf("\nKeyword:\t\tthase");
            i += 4;
        }
        else
        {
            errorPrint(s, len, j, i);
        }
    }
    else if (s[i] == 'm')
    {
        int j = i;
        i++;
        if (s[i] == 'e' && s[i + 1] == 't' && s[i + 2] == 'e' && s[i + 3]
== 'r'&& s[i + 4] == 's')
```

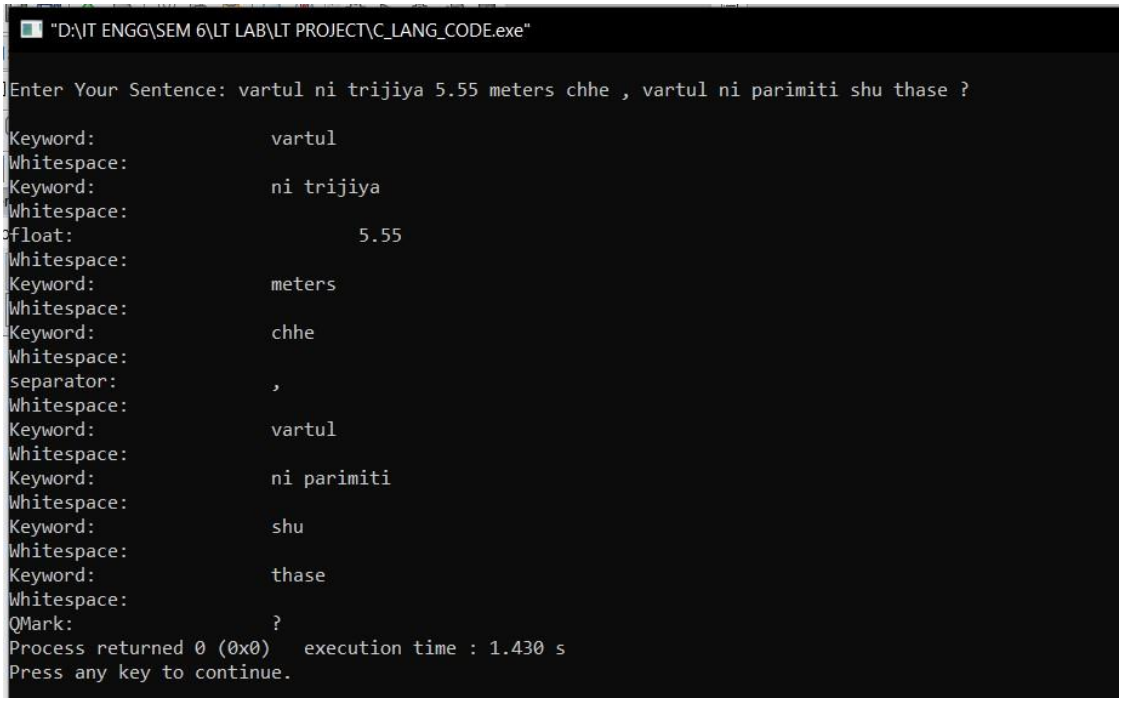
```
    {
        printf("\nKeyword:\t\tmeters");
        i += 5;
    }
    else
    {
        errorPrint(s, len, j, i);
    }
}

else if (s[i] == 's')
{
    int j = i;
    i++;
    if (s[i] == 'h' && s[i + 1] == 'u')
    {
        printf("\nKeyword:\t\tshu");
        i += 2;
    }
    else
    {
        errorPrint(s, len, j, i);
    }
}

else if (s[i] == 't')
{
    int j = i;
    i++;
    if (s[i] == 'o')
    {
        printf("\nKeyword:\t\tto");
        i++;
    }
    else
    {
        errorPrint(s, len, j, i);
    }
}

else if (s[i] == 'i')
{
    int j = i;
    i++;
    if (s[i] == 's')
    {
        printf("\nKeyword:\t\tis");
        i++;
    }
    else
```

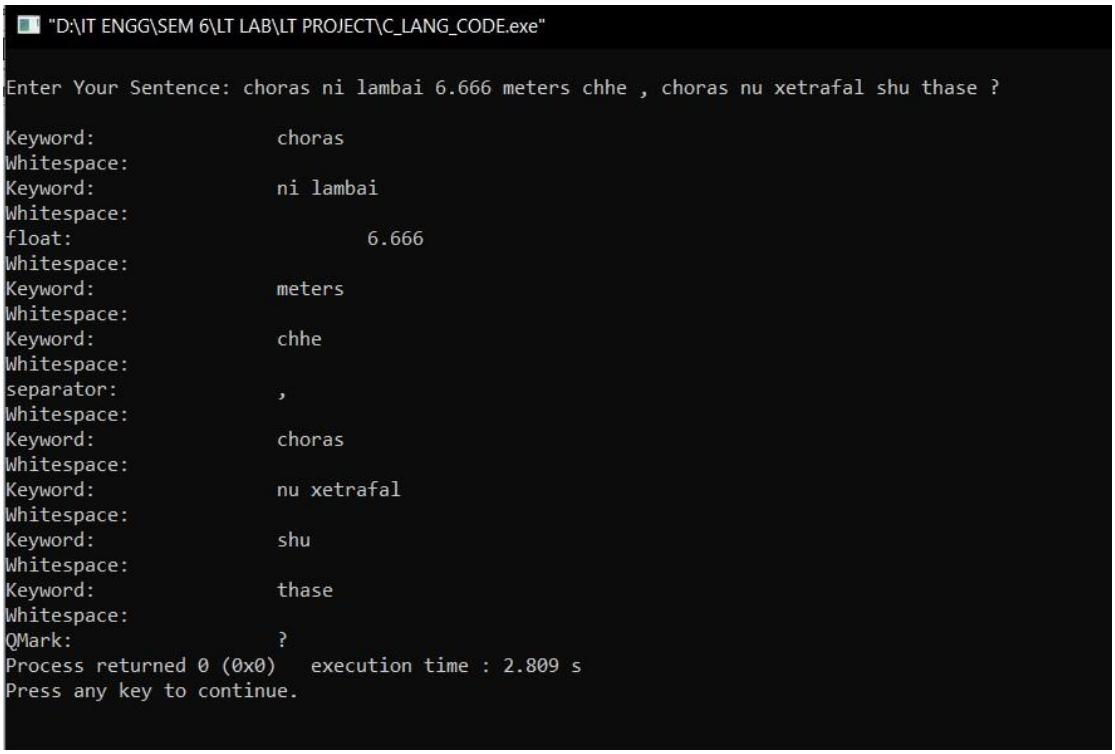
```
            {
                errorPrint(s, len, j, i);
            }
        }
        //-------------------------------------------------------------------------
-----------------------------

        //-------------------------------------------------------------------------
-----------------------------
        else if (isdigit(s[i]) != 0)
        {
            bool isFloat = false;
            int j = i;
            i++;
            while (s[i] != ' ')
            {
                if (isdigit(s[i]) != 0)
                {
                    i++;
                }
                else if (s[i] == '.')
                {
                    i++;
                    while (s[i] != ' ')
                    {
                        if (isdigit(s[i]) != 0)
                        {
                            i++;
                        }
                        if (s[i] == '\0')
                        {
                            break;
                        }
                    }
                    isFloat = true;
                    break;
                }

                if (s[i] == '\0')
                {
                    break;
                }
            }

            if (!isFloat)
            {
                char temp[10];
                for (int k = 0; k < i-j; k++)
                {
                    temp[k] = s[k + j];
```

```
                }
                printf("\nint:\t\t%s", temp);
            }

            if (isFloat)
            {
                char temp[10];
                for (int k = 0; k < i-j; k++)
                {
                    temp[k] = s[k + j];
                }
                printf("\nfloat:\t\t\t%s", temp);
            }
        }
        //------------------------------------------------------------------------
------------------------------

        //------------------------------------------------------------------------
------------------------------
    }
}

int main()
{
    char string[100];
    printf("\nEnter Your Sentence: ");
    gets(string);

    int length = strlen(string);

    scanner(string, length);

    return 0;
}

/*

vartul ni trijiya meters chhe , vartul nu xetrafal thase ? 5.64
vartul ni trijiya 5 meters chhe , vartul nu xetrafal shu thase ?
vartul ni trijiya 5 meters chhe , vartul ni parimiti shu thase ?
choras ni lambai 5 meters chhe , choras nu xetrafal shu thase ?
choras ni lambai 5 meters chhe , choras nu xetrafal shu thase ?

*/
```

2.0.8 Output screenshots of scanner using c






"D:\IT ENGG\SEM 6\LT LAB\LT PROJECT\C_LANG_CODE.exe"
Enter Your Sentence: choras ni lambai 6.666 meters chhe , choras nu xetrafal shu thase ?
Keyword: choras
Whitespace:
Keyword: ni lambai
Whitespace:
float: 6.666
Whitespace:
Keyword: meters
Whitespace:
Keyword: chhe
Whitespace:
separator: ,
Whitespace:
Keyword: choras
Whitespace:
Keyword: nu xetrafal
Whitespace:
Keyword: shu
Whitespace:
Keyword: thase
Whitespace:
QMark: ?
Process returned 0 (0x0) execution time : 2.809 s
Press any key to continue.

3.0 SYNTAX ANALYZER DESIGN

3.0.1 Grammar rules

Choras ni lambai 5 meter chhe , choras nu xetrafal shu thase ?

KW KW DIGIT UNIT KW PUN KW OP KW KW PUN

Non-terminals: S , KW , M , P , O , OP , D , U , Q , q , e , s

Terminals: meter|meters|centimeter|centimeters| ni parimiti|nu xetraf al vartul|choras|shu|chhe|thase|ni trijya|ni lambai | int | float q | e | s | ? | . | ,

S -> KW

KW -> KW KW | KW M | KW P | KW O

M -> D U

O -> OP KW

U -> meter|meters|centimeter|centimeters

OP -> ni parimiti|nu xetrafal

KW -> vartul|choras|shu|chhe|thase|ni trijya|ni lambai

D -> int | float

P -> q | e | s

q -> ?

e -> .

s -> ,

OUTPUT

Fisrt :

First(S):{ vartul,choras,shu,chhe,thase,ni trijya,ni lambai }

First(KW):{ vartul,choras,shu,chhe,thase,ni trijya,ni lambai }

First(M):{ int , float }

First(O):{ ni parimiti , nu xetrafal }

First(U):{ meter,meters,centimeter,centimeters }

First(OP):{ ni parimiti , nu xetrafal }

First(D):{int , float }

First(P):{? . , }

First(q):{?}

First(e):{.}

First(s):{,}

Follow :

Follow(S):{}

Follow(KW):{ vartul ,choras ,shu ,chhe, thase ,ni trijya, ni lambai ni parimiti , nu xetrafal int, float , ? . , }

Follow(M):{ $ }

Follow(O):{xetrafal,vartul,choras,shu,chhe,thase,ni trijya,ni lambai}

Follow(U):{$}

Follow(OP):{ vartul,choras,shu,chhe,thase,ni trijya,ni lambai }

Follow(D):{ meter,meters,centimeter,centimeters }

Follow(P):{$}

Follow(e):{$}

Follow(q):{$}

Follow(s):{$}

3.0.2 Yacc based imlementation of syntax analyzer

CODE :

Ltproj.lex

```
%{
#include<stdio.h>
#include "y.tab.h"
%}

Keyword "vartul"|"choras"|"shu"|"chhe"|"thase"|"ni trijya"|"ni lambai"
Operator "ni parimiti"|"nu xetrafal"
Unit "meter"|"meters"|"centimeter"|"centimeters"|"squarecentimeters"
Digit [0-9]
Integer {Digit}+
Float {Digit}+(.{Digit})
Qmark "?"
WhiteSpace " "
NewLine [\n]
Eos "."
Separater ","

%%
{Keyword} {printf("Valid Keyword :%10s\n",yytext); return KEYWORD;}
{Operator} {printf("Valid Operator :%10s\n",yytext); return OPERATER;}
{Unit} {printf("Valid Unit :%10s\n",yytext); return UNIT;}
{Digit} {printf("Valid Digit :%10s\n",yytext); return DIGIT;}
{Integer} {printf("Valid Integer :%10s\n",yytext); return INTEGER;}
{Float} {printf("Valid Float :%10s\n",yytext); return FLOAT;}
{Qmark} {printf("Valid Qmark :%10s\n",yytext); return QMARK;}
```

```
{WhiteSpace} {printf("Valid WhiteSpace :%10s\n",yytext); return WHITESPACE;}
{NewLine} {printf("Valid NewLine :%10s\n",yytext); return NEWLINE;}
{Eos} {printf("Valid Eos :%10s\n",yytext); return EOS;}
{Separater} {printf("Valid Separater :%10s\n",yytext); return SEPARATER;}
. {printf("%s is Not valid token.\n",yytext); return;}
end   {return 0;}
%%

int yywrap(){return 1;}




/*

vartul ni trijya 5 meter chhe , vartul nu xetrafal shu thase ?
vartul ni trijya 55.5 meter chhe , vartul nu xetrafal shu thase ?
vartul ni trijya 55 meter chhe , vartul ni parimiti shu thase ?
choras ni lambai 0.5 meter chhe , choras nu xetrafal shu thase ?
choras ni lambai 123 centimeter chhe , choras nu xetrafal shu thase ?


vartul ni trijya  meter chhe , vartul nu xetrafal shu thase ?

vartul ni trijya 5 meter chhe , vartul nu xetrafal shu thse ?

vartul ni trijya 5 meter chhe , ? vartul nu xetrafal shu thse ?

vartul ni trijya 5 meter chhe , ? vartul nu xetrafal shu thse ?

vartul ni trijya 5 meter chhe

vartul ni trijya 5 meter chhe , ? vartul nu xetrafal ni parimiti  shu thse ?
```

```
*/
```

Ltptoj.yacc

```
%{
#include <stdio.h>
#include<stdlib.h>
#define YYERROR_VERBOSE 1
void yyerror(char *err);

%}
%token KEYWORD OPERATER UNIT DIGIT INTEGER FLOAT QMARK WHITESPACE NEWLINE EOS
SEPARATER
%%
S : K{ printf("\nThese Sentences is Valid. \n\n"); return 0; };

K : KEYWORD {}
| K WHITESPACE K {}

| K WHITESPACE M {}
| K WHITESPACE P {}
| K WHITESPACE O {}
;

M : D WHITESPACE UNIT {};

O : OPERATER WHITESPACE KEYWORD {};

P : SEPARATER {}
| QMARK {}
```

```
| EOS {}
| NEWLINE {};

D : DIGIT {}
| INTEGER {}
| FLOAT {};



%%
void yyerror(char *err) {
printf("Error: ");
fprintf(stderr, "%s\n", err);
exit(1);
}
int main() {
printf("Enter Sentences:\n");
yyparse();
}
```

3.0.3 OUTPUT:

1.

```
kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT/l-10$ yacc -d ltproj.yacc
ltproj.yacc: warning: 1 shift/reduce conflict [-Wconflicts-sr]
kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT/l-10$ lex ltproj.lex
kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT/l-10$ cc y.tab.c lex.yy.c -o ltproj
        In function ‘       ’:
                warning: implicit declaration of function ‘     ’ [-Wimplicit-function-declaration]
 1247 |       yychar = yylex ();
      |                ^~~~~
                warning: passing argument 1 of ‘       ’ discards ‘     ’ qualifier from pointer target type [-Wdiscarded-qualifiers]
 1491 |         yyerror (yymsgp);
      |                  ^~~~~~
                note: expected ‘      ’ but argument is of type ‘            ’
    5 | void yyerror(char *err);
      |              ~~~~~~^~~
           In function ‘     ’:
                warning: ‘      ’ with no value, in function returning non-void
   30 | . {printf("%s is Not valid token.\n",yytext); return;}
      |                                               ^~~~~~
                note: declared here
  670 | #define YY_DECL int yylex (void)
      |                     ^~~~~
                note: in expansion of macro ‘       ’
  690 | YY_DECL
      | ^~~~~~~
^[[Akashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT/l-10$ ./ltproj
Enter Sentences:
vartul ni trijya 5 meter chhe , vartul nu xetrafal shu thase ?
Valid Keyword :    vartul
Valid WhiteSpace :
Valid Keyword : ni trijya
Valid WhiteSpace :
Valid Digit :         5
Valid WhiteSpace :
Valid Unit :     meter
Valid WhiteSpace :
Valid Keyword :      chhe
Valid WhiteSpace :
Valid Separater :         ,
Valid WhiteSpace :
Valid Keyword :    vartul
Valid WhiteSpace :
Valid Operator :nu xetrafal
Valid WhiteSpace :
Valid Keyword :       shu
Valid WhiteSpace :
Valid Keyword :     thase
Valid WhiteSpace :
Valid Qmark :         ?
Valid NewLine :


These Sentences is Valid.
```

2.

```
kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT/l-10$ ./ltproj
Enter Sentences:
vartul ni trijya 55.5 meter chhe , vartul nu xetrafal shu thase ?
Valid Keyword :    vartul
Valid WhiteSpace :
Valid Keyword : ni trijya
Valid WhiteSpace :
Valid Float :      55.5
Valid WhiteSpace :
Valid Unit :     meter
Valid WhiteSpace :
Valid Keyword :      chhe
Valid WhiteSpace :
Valid Separater :         ,
Valid WhiteSpace :
Valid Keyword :    vartul
Valid WhiteSpace :
Valid Operator :nu xetrafal
Valid WhiteSpace :
Valid Keyword :       shu
Valid WhiteSpace :
Valid Keyword :     thase
Valid WhiteSpace :
Valid Qmark :         ?
Valid NewLine :


These Sentences is Valid.
```

3.

```
kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT/l-10$ ./ltproj
Enter Sentences:
vartul ni trijya 55 meter chhe , vartul ni parimiti shu thase ?
Valid Keyword :    vartul
Valid WhiteSpace :
Valid Keyword : ni trijya
Valid WhiteSpace :
Valid Integer :        55
Valid WhiteSpace :
Valid Unit :     meter
Valid WhiteSpace :
Valid Keyword :      chhe
Valid WhiteSpace :
Valid Separater :         ,
Valid WhiteSpace :
Valid Keyword :    vartul
Valid WhiteSpace :
Valid Operator :ni parimiti
Valid WhiteSpace :
Valid Keyword :       shu
Valid WhiteSpace :
Valid Keyword :     thase
Valid WhiteSpace :
Valid Qmark :         ?
Valid NewLine :


These Sentences is Valid.
```

4.

```
kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT/l-10$ ./ltproj
Enter Sentences:
choras ni lambai 0.5 meter chhe , choras nu xetrafal shu thase ?
Valid Keyword :    choras
Valid WhiteSpace :
Valid Keyword : ni lambai
Valid WhiteSpace :
Valid Float :       0.5
Valid WhiteSpace :
Valid Unit :     meter
Valid WhiteSpace :
Valid Keyword :      chhe
Valid WhiteSpace :
Valid Separater :         ,
Valid WhiteSpace :
Valid Keyword :    choras
Valid WhiteSpace :
Valid Operator :nu xetrafal
Valid WhiteSpace :
Valid Keyword :       shu
Valid WhiteSpace :
Valid Keyword :     thase
Valid WhiteSpace :
Valid Qmark :         ?
Valid NewLine :


These Sentences is Valid.
```

5.

```
kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT/l-10$ ./ltproj
Enter Sentences:
choras ni lambai 123 centimeter chhe , choras nu xetrafal shu thase ?
Valid Keyword :    choras
Valid WhiteSpace :
Valid Keyword : ni lambai
Valid WhiteSpace :
Valid Integer :       123
Valid WhiteSpace :
Valid Unit :centimeter
Valid WhiteSpace :
Valid Keyword :      chhe
Valid WhiteSpace :
Valid Separater :         ,
Valid WhiteSpace :
Valid Keyword :    choras
Valid WhiteSpace :
Valid Operator :nu xetrafal
Valid WhiteSpace :
Valid Keyword :       shu
Valid WhiteSpace :
Valid Keyword :     thase
Valid WhiteSpace :
Valid Qmark :         ?
Valid NewLine :


These Sentences is Valid.
```

6. Not Complete Sentence.

```
kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT/l-10$ ./ltproj
Enter Sentences:
vartul ni trijya  meter chhe , vartul nu xetrafal shu thase ?
Valid Keyword :    vartul
Valid WhiteSpace :
Valid Keyword : ni trijya
Valid WhiteSpace :
Valid WhiteSpace :
syntax error, unexpected WHITESPACE
Error: kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT/l-10$
```

7. Invalid Token.

```
kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT/l-10$ ./ltproj
Enter Sentences:
vartul ni trijya 5 meter chhe , vartul nu xetrafal shu thse ?
Valid Keyword :    vartul
Valid WhiteSpace :
Valid Keyword : ni trijya
Valid WhiteSpace :
Valid Digit :         5
Valid WhiteSpace :
Valid Unit :     meter
Valid WhiteSpace :
Valid Keyword :      chhe
Valid WhiteSpace :
Valid Separater :         ,
Valid WhiteSpace :
Valid Keyword :    vartul
Valid WhiteSpace :
Valid Operator :nu xetrafal
Valid WhiteSpace :
Valid Keyword :       shu
Valid WhiteSpace :
t is Not valid token.
syntax error, unexpected $undefined
Error: kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT/l-10$
```

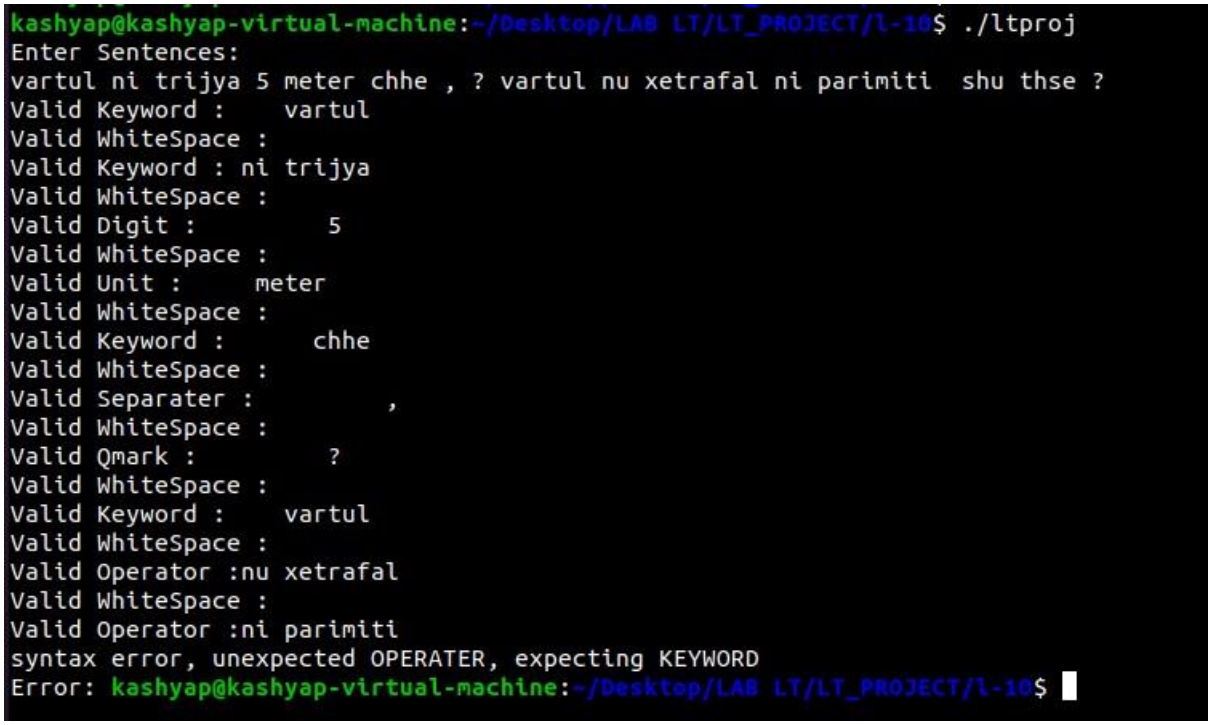
8. Invalid Syntex. Not following Grammer Rule.

```
Error: kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT/l-10$ ./ltproj
Enter Sentences:
vartul ni trijya 5 meter chhe , ? vartul nu xetrafal shu thse ?
Valid Keyword :    vartul
Valid WhiteSpace :
Valid Keyword : ni trijya
Valid WhiteSpace :
Valid Digit :         5
Valid WhiteSpace :
Valid Unit :     meter
Valid WhiteSpace :
Valid Keyword :      chhe
Valid WhiteSpace :
Valid Separater :          ,
Valid WhiteSpace :
Valid Qmark :         ?
Valid WhiteSpace :
Valid Keyword :    vartul
Valid WhiteSpace :
Valid Operator :nu xetrafal
Valid WhiteSpace :
Valid Keyword :       shu
Valid WhiteSpace :
t is Not valid token.
syntax error, unexpected $undefined
Error: kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT/l-10$
```

9. Program is not complete yet (expecting input after) and invalid Token.

```
kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT/l-10$ ./ltproj
Enter Sentences:
vartul ni trijya 5 meter chhe 5k6
Valid Keyword :    vartul
Valid WhiteSpace :
Valid Keyword : ni trijya
Valid WhiteSpace :
Valid Digit :         5
Valid WhiteSpace :
Valid Unit :     meter
Valid WhiteSpace :
Valid Keyword :      chhe
Valid WhiteSpace :
Valid Float :       5k6
Valid NewLine :

syntax error, unexpected NEWLINE, expecting WHITESPACE
Error: kashyap@kashyap-virtual-machine:~/Desktop/LAB LT/LT_PROJECT/l-10$
```

10. Invalid combination.



4.0 CONCLUSION

Until now we have learned many languages in our academic curriculum but we didn’t know the whole complex logic which makes the computer understand those languages, this subject Language Translator makes us understand those complex mechanism.

This project has been implemented from what we have learned in our college curriculum and many rich resources from the web. After doing this project we conclude that we have got more knowledge about how different compilers are working in practical world and also how various types of errors are handled. And for this wonderful subject and project we thank to Prof. Nikita P. Desai for guidance in the project and give chance to know how machine works actually