Project Report on
Compiler for
"Binary to Decimal Conversion"

Developed by

IT154 - GAURAV TELI - 20ITUBS007
IT157 - DHRUV THUMMAR - 21ITUOD007
IT159 - VACHA PATEL -20ITUON054
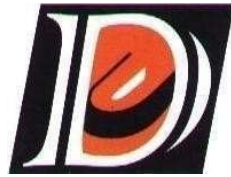**IT160 - VARANA NAVADIYA -20ITUON139**

Guided By:
Prof.Nilamba Vala
Dept. of Information Technology

Department of Information Technology
Faculty of Technology, Dharmsinh Desai University
College Road, Nadiad-387001
2022-2023

DHARMSINH DESAI UNIVERSITY
NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled "**Compiler for Binary to Decimal Conversion**" is a bonafide report of the work carried out by

1. IT154 - GAURAV TELI - 20ITUBS007
2. IT157 - DHRUV THUMMAR - 21ITUOD007
3. IT159 - VACHA PATEL -20ITUON054
4. IT160 - VARANA NAVADIYA -20ITUON139

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 2022-2023.

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

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

Date:

Date:

1.   Project Details

Language Name: Binary to Decimal Conversion
Language description:

This Program takes input in the form of 'T' and 'F' which represent 1 and 0 respectively and convert it to decimal.

Example of valid program in this language is -
Given string is : "TT"
Output : 3.

2.   Project Planning

List of Students with their Roles/Responsibilities:

IT154 GAURAV TELI   : Regular Expression , DFA Design.
IT157 DHRUV THUMMAR : Algorithm Design and implementation.
**IT159 VACHA PATEL**   **:** Scanner phase Implementation.
IT160 VARANA NAVADIYA: Grammar rules, YACC implementation.

3. Algorithm Design

Steps:

1. The lexer reads input from standard input and recognizes tokens 'F'and 'T' based on regular expression patterns defined in the Lex file.

2. The Yacc parser uses a grammar to determine whether the input is valid according to the language definition.

3. When the input is valid, the Yacc parser executes code defined in the grammar rules.

4. If a parsing error occurs, the yyerror() function is called to print an error message to standard output.

5. When parsing is complete, the driver code in the main function terminates the program.

4. Finite Automata :
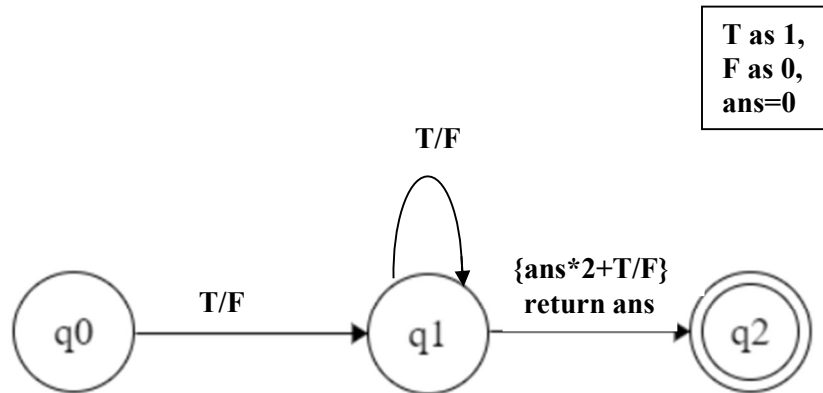
5.   Code Implementation

1) ltproject.l

```
%{
        /* Definition section */
        #include<stdio.h>
        #include<stdlib.h>
        #include"y.tab.h"
        extern int yylval;
%}

/* Rule Section */
 %%
        F  {yylval=0;return ZERO;}
        T  {yylval=1;return ONE;}

        [ \t] {;}
        \n return 0;
        . return yytext[0];
%%


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

2) Parser.y

```
%{
    /* Definition section */
    #include<stdio.h>
    #include<stdlib.h>
    void yyerror(char *s);
%}
%token ZERO ONE

    /* Rule Section */
    %%
    N: L {printf("\n%d", $$);}
    L: L B {$$=$1*2+$2;}
    | B {$$=$1;}
    B:ZERO {$$=$1;}
    |ONE {$$=$1;};
%%

//driver code
int main()
{
        while(yyparse());
}

yyerror(char *s)
{
        fprintf(stdout, "\n%s", s);
}
```

6.   Output :

7.   Github Link:

https://github.com/dhruv2210/LT_Project-BinarytoDecimal-.git

8.   CONCLUSION

This project has been implemented from what we have learned in our college curriculum and some 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 arehandled.