|  |  |
| --- | --- |
|  | **AIR UNIVERSITY** |
| **DEPARTMENT OF COMPUTER SCIENCE** |
| **Lab Task 5** |

**Student Name: Hamza Umer Farooq Reg. No: 200789**

**Subject: Compiler Construction Semester: VIII**

**Objective: Binary to decimal and total 0s and 1s same / start & end with 0**

**ASSESSMENT:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | **Excellent**  **(5)** | **Good**  **(4)** | **Average**  **(3)** | **Satisfactory**  **(2)** | **Unsatisfactory (1)** |
| **Ability to Conduct**  Task |  |  |  |  |  |
| **Ability to assimilate the results** |  |  |  |  |  |
| **Effective use of theorems/postulates/formulas** |  |  |  |  |  |

Total Marks:

Obtained Marks:

**REPORT ASSESSMENT:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | **Excellent**  **(5)** | **Good**  **(4)** | **Average**  **(3)** | **Satisfactory**  **(2)** | **Unsatisfactory**  **(1)** |
| **Data presentation** |  |  |  |  |  |
| **Experimental results** |  |  |  |  |  |
| **Conclusion** |  |  |  |  |  |

Question 1:

**binary.l:**

%{

#include <stdio.h>

#include <stdlib.h>

#include "binary.tab.h"

extern int yylval;

%}

%%

0 {yylval = 0;return ZERO;}

1 {yylval = 1;return ONE;}

[ \t] {;}

\n return 0;

. return yytext[0];

%%

int yywrap(){

return 1;

}

binary.y:

%{

#include <stdio.h>

#include <stdlib.h>

void yyerror(const char \*s);

int yylex(void);

int yywrap(void);

%}

%token ZERO ONE

%%

N: L {printf("\nDecimal Number is: %d\n", $1);}

L: L B {$$ = $1 \* 2 + $2;}

 | B {$$ = $1;}

B: ZERO { $$ = 0; }

 | ONE { $$ = 1; }

%%

void yyerror(const char \*s) {

    fprintf(stderr, "%s\n", s);

}

int main() {

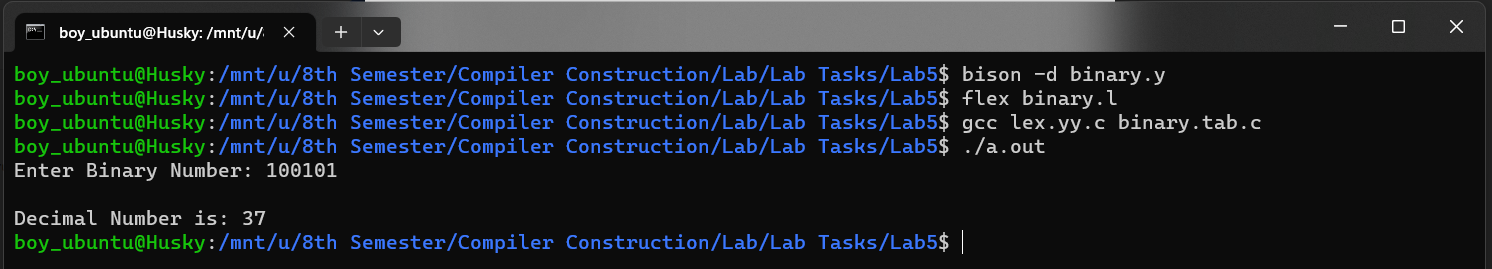
    printf("Enter Binary Number: ");

    yyparse();

    return 0;

}

**Output:**



Question 2:  
  
q2.y:

%{

#include <stdio.h>

int start\_end\_zero = 0;

int start\_end\_one = 0;

int equal\_zero\_one = 1;

%}

%token DIGIT

%%

input : sequence {

            if(start\_end\_zero || start\_end\_one)

                printf("Rule 1 matched:\n");

            if(equal\_zero\_one)

                printf("Rule 2 matched:\n");

            else

        }

      | { printf("Empty sequence\n"); }

      ;

sequence : '0' sequence\_body '0' { printf("Matched sequence starting and ending with 0.\n"); start\_end\_zero = 1; }

         | '1' sequence\_body '1' { printf("Matched sequence starting and ending with 1.\n"); start\_end\_one = 1; }

         ;

sequence\_body : DIGIT { printf("Matched digit: %d\n", $1); if($1 == 0) equal\_zero\_one++; else if($1 == 1) equal\_zero\_one--; }

              | /\* empty \*/ { printf("No digits found\n"); }

              ;

%%

int main() {

    yyparse();

    return 0;

}

int yyerror(const char \*s) {

    printf("%s\n", s);

    return 0;

}

q2.l:

%{

#include "q2.tab.h"

%}

%%

[0-1]   { yylval = yytext[0] - '0'; printf("Token: %s\n", yytext); return DIGIT; }

\n      { /\* skip newlines \*/ }

.       { /\* skip other characters \*/ }

%%

int yywrap() {

    return 1;

}

Output:

