|  |  |
| --- | --- |
| A blue logo with a black background  Description automatically generated | **AIR UNIVERSITY** |
| **DEPARTMENT OF COMPUTER SCIENCE** |
| **Assignment 1** |

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

**Subject: Compiler Construction Semester: VIII**

**Objective: Decimal to Binary**

**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** |  |  |  |  |  |

**Code(s):**

**decimal\_to\_binary.l**

%{

#include <stdio.h>

#include "decimal\_to\_binary.tab.h"

%}

%%

[0-9]+ { yylval = atoi(yytext); return NUM; } // Recognize decimal numbers and return NUM token

\n    { return END; } // Recognize end of input and return END token

.     { ; } // Ignore any other characters

%%

/\*\*

 \* This function is called when the end of input is reached.

 \* It returns 1 to indicate that there is no more input to process.

 \*/

int yywrap() {

    return 1;

}

**decimal\_to\_binary.y**

%{

#include <stdio.h>

#include <stdlib.h>

// function to convert decimal to binary

void dec\_to\_bin(int n) {

    int binary[32]; // array to store binary number

    int i = 0; // counter for binary array

    while (n > 0) { // convert decimal to binary

        binary[i] = n % 2;  // store remainder in binary array

        n = n / 2; // divide number by 2

        i++; // increment counter

    }

    printf("Binary Equivalent: "); // print binary equivalent

    for (int j = i - 1; j >= 0; j--) { // print binary number

        printf("%d", binary[j]); // print binary digit

    }

    printf("\n");

}

int yylex(void);

void yyerror(const char \*s);

%}

%token NUM

%token END

// rules

%%

input: NUM { printf("Binary Equivalent: "); dec\_to\_bin($1); }  // convert decimal to binary

     | input NUM { dec\_to\_bin($2); } // convert decimal to binary

     | input END { printf("\n"); } // print newline

     ;

%%

void yyerror(const char \*s) { // error handling function

    fprintf(stderr, "%s\n", s);

}

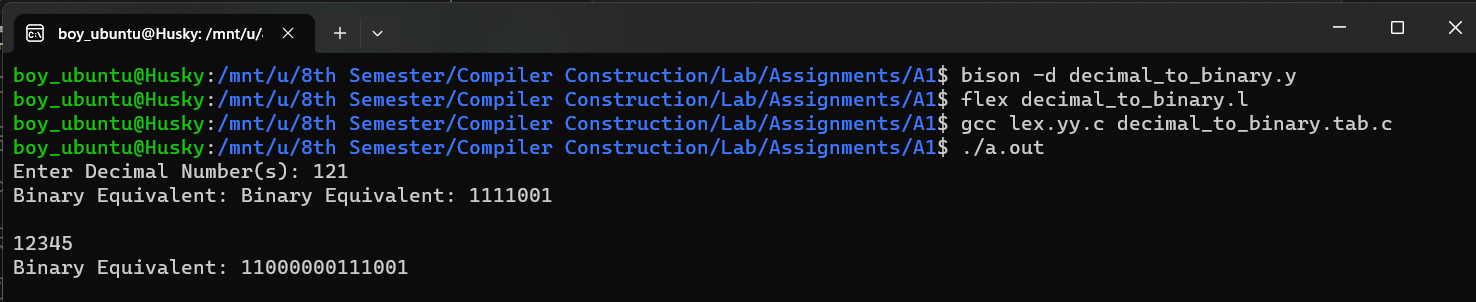
int main() {

    printf("Enter Decimal Number(s): ");

    yyparse();

    return 0;

}

**Output:  
**