|  |  |
| --- | --- |
| A blue logo with a black background  Description automatically generated | **AIR UNIVERSITY** |
| **DEPARTMENT OF COMPUTER SCIENCE** |
| **Lab Task 4** |

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

**Subject: Compiler Construction Semester: VIII**

**Objective: Bison and Flex.**

**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** |  |  |  |  |  |

Task 1:

hello.l

/\* //Modified Bison File: marin.jb.free.fr/bison   \*/

%{

#include "hello.tab.h"

int yyerror(char \*errormsg);

%}

%%

("hi"|"oi")"\n"       { return HI;  }

("tchau"|"bye")"\n"   { return BYE; }

.                     { yyerror("Unknown char");  }

%%

int main(void)

{

   yyparse();

   return 0;

}

int yywrap(void)

{

   return 0;

}

int yyerror(char \*errormsg)

{

    fprintf(stderr, "%s\n", errormsg);

    exit(1);

}

hello.y

%{

#include <stdio.h>

#include <stdlib.h>

int yylex(void);

int yyerror(const char \*s);

%}

%token HI BYE

%%

program:

         hi bye

        ;

hi:

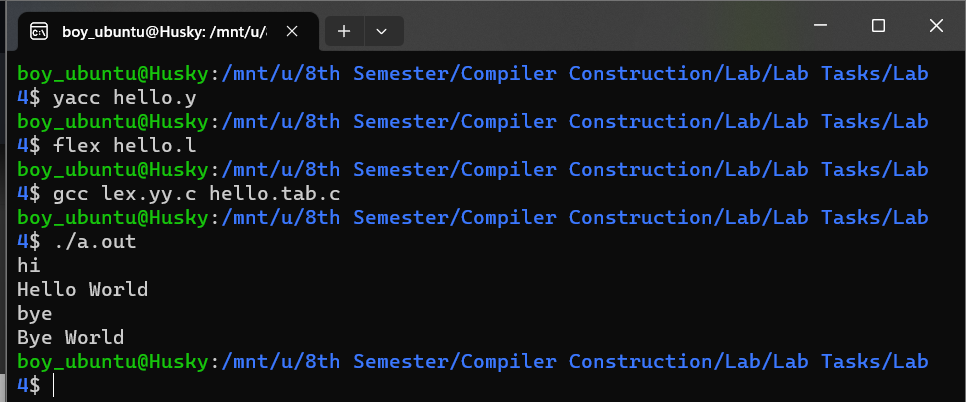
        HI     { printf("Hello World\n");   }

        ;

bye:

        BYE    { printf("Bye World\n"); exit(0); }

         ;

Output:  


calc.l

%{

#include "calc1.tab.h"

extern int yylval; // Make sure yylval is declared as an integer

#include <stdlib.h> // Include stdlib.h for the atoi function

%}

%%

[0-9]+       { yylval = atoi(yytext); return NUMBER; } // Convert matched string to integer

[a-zA-Z]+    {// Ignore letters}

[\t]+        ;  // Ignore tabs

[\n]         { return '\n'; } // Return newline character

.            { return yytext[0]; } // Return any other character

%%

int yywrap(void)

{

   return 0;

}

calc.y

%{

#include <stdlib.h>

#include <stdio.h>

int yylex(void);

void yyerror(char \*errormsg);

%}

%token NUMBER ALPHA

%left '+' '-'

%left '\*' '/'

%left '(' ')'

%%

grammer: expr '\n' {

    printf("\n Arithmetic Expression is valid\n");

    printf("\n Expression Result : %d\n", $1); // Changed $$ to $1 to reflect the result

    exit(0);

}

expr: expr '+' expr { $$ = $1 + $3; }

    | expr '\*' expr { $$ = $1 \* $3; }

    | expr '/' expr { $$ = $1 / $3; }

    | expr '-' expr { $$ = $1 - $3; }

    | '(' expr ')'  { $$ = $2; }

    | NUMBER        { $$ = $1; }

    | ALPHA         { /\* Handle alpha tokens appropriately \*/ }

    ;

%%

int main(void)

{

    printf("Enter the Arithmetic Expression\n");

    yyparse();

    return 0;

}

void yyerror(char \*errormsg)

{

    printf("Arithmetic expression is Invalid\n");

    exit(1);

}

Output:  
  
A computer screen with blue text

Description automatically generated

Problem 1:

task1.l:  
%{

#include "calc1.tab.h"

extern int yylval; // Make sure yylval is declared as an integer

#include <stdlib.h> // Include stdlib.h for the atoi function

%}

%%

[0-9]+       { yylval = atoi(yytext); return NUMBER; } // Convert matched string to integer

[a-zA-Z]+    {

                if (strstr(yytext, "azbz") != NULL) {

                    return ALPHA;

                } else {

                    /\* Ignore other letters \*/

                }

            }

[\t]+        ;  // Ignore tabs

[\n]         { return '\n'; } // Return newline character

.            { return yytext[0]; } // Return any other character

%%

int yywrap(void)

{

   return 0;

}

task1.y

%{

#include <stdlib.h>

#include <stdio.h>

int yylex(void);

void yyerror(char \*errormsg);

%}

%token NUMBER ALPHA

%left '+' '-'

%left '\*' '/'

%left '(' ')'

%%

grammer: expr '\n' {

    printf("\n Arithmetic Expression is valid\n");

    printf("\n Expression Result : %d\n", $1);

    exit(0);

}

expr: expr '+' expr { $$ = $1 + $3; }

    | expr '\*' expr { $$ = $1 \* $3; }

    | expr '/' expr { $$ = $1 / $3; }

    | expr '-' expr { $$ = $1 - $3; }

    | '(' expr ')'  { $$ = $2; }

    | NUMBER        { $$ = $1; }

    | ALPHA         { printf("\nAZBZ found in given string.\n");return 0;}

    ;

%%

int main(void)

{

    printf("Enter the Arithmetic Expression OR a STRING\n");

    yyparse();

    return 0;

}

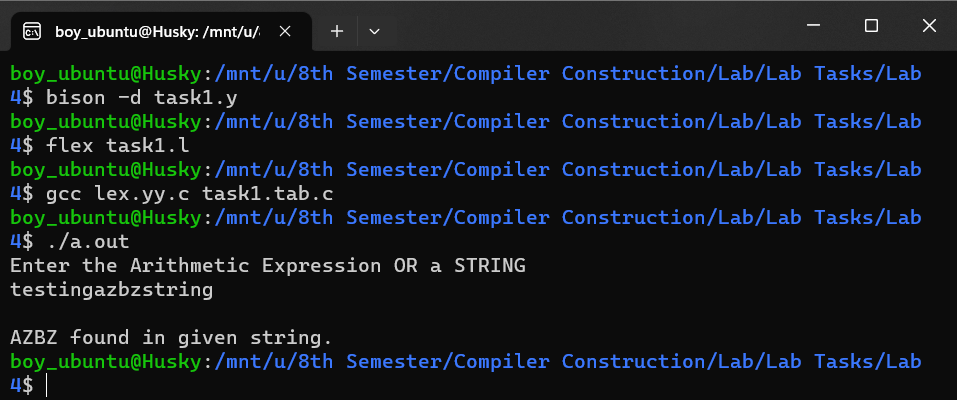
void yyerror(char \*errormsg)

{

    printf("Arithmetic expression is Invalid\n");

    exit(1);

}

Output:  


TASK 2:

task2.l

%{

#include "calc1.tab.h"

extern int yylval;

#include <stdlib.h>

%}

%%

[0-9]+       { yylval = atoi(yytext); return NUMBER; }

[\t]+        {}

[\n]         { return '\n'; }

.            { return yytext[0];}

%%

int yywrap(void)

{

   return 0;

}

task2.y

%{

#include <stdlib.h>

#include <stdio.h>

int yylex(void);

void yyerror(char \*errormsg);

%}

%token NUMBER

%left '+' '-'

%left '\*' '/'

%left '(' ')'

%%

grammer: expr '\n' {

    printf("\n Result : %d\n", $1);

    exit(0);

}

expr: expr '+' expr { $$ = $1 + $3; }

    | expr '\*' expr { $$ = $1 \* $3; }

    | expr '/' expr { $$ = $1 / $3; }

    | expr '-' expr { $$ = $1 - $3; }

    | '(' expr ')'  { $$ = $2; }

    | NUMBER        { $$ = $1; }

    ;

%%

int main(void)

{

    printf("\nEnter the Arithmetic Expression: ");

    yyparse();

    return 0;

}

void yyerror(char \*errormsg)

{

    printf("Arithmetic expression is Invalid\n");

    exit(1);

}

Output:  
A screenshot of a computer

Description automatically generated