ASSIGNMENTS SOLUTION

COURSE CODE : CSE/PC/B/S/322

COMPILER DESIGN LAB

By

SHAHIR HABIB

002210501006

Department of Computer Scirence & Engineering,

Jadavpur University , Kolkata

January-April, 2025

Assignment3\problem1.lex

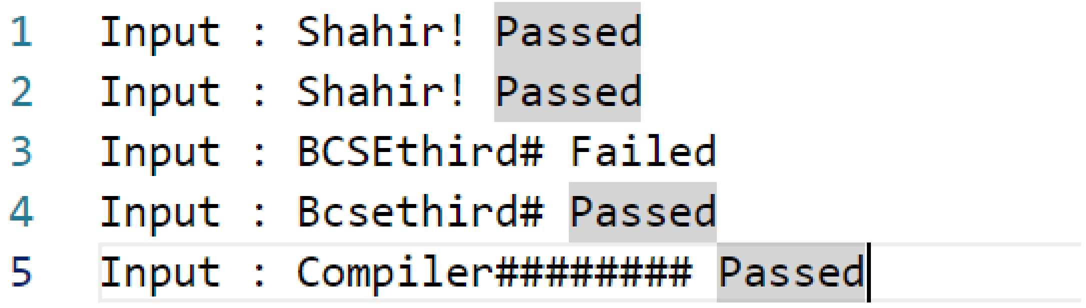
```
#@title Writing Lex program
%%writefile program.l

%{
    #include <stdio.h>

%}

%%
1(0|1|.)*101\n {printf("Input : %s\n Passed",yytext);}
.* {printf("Input : %s Failed \n",yytext);}
%%



void main(int argc, char *argv[]){

    yylex();


}

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

Assignment3\problem2.lex

```
%{
#include <stdio.h>
%}

%%

^[A-Z][a-z0-9]*[!@#$%^&*()_+={}|\:;<>?~`-]$ {
    printf("Input : %s\n Passed",yytext);
}

.|\n
    {printf("Input : %s Failed \n",yytext);}

%%

int main() {
    yylex();
    return 0;
}
```

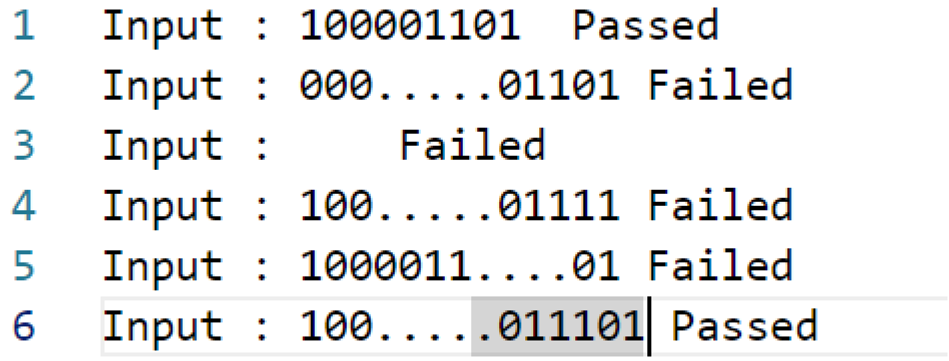
Assignment3\problem3.lex

```
#@title Writing Lex program
%%writefile program.l

%{
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int num1,num2;
char operator;
%}

%%
[0-9]+[+\-*/%^][0-9]+ {

          int i=0;
          char s[100];
          strcpy(s,yytext);
          while(i<strlen(s)){
            num1=(num1*10)+(s[i]-'0');
            i++;
          }

          operator =  s[i];
          while(i<strlen(s)){
            num2=(num2*10)+(s[i]-'0');
            i++;
          }
           switch(operator) {
               case '+': printf("Result: %d\n", num1 + num2); break;
               case '-': printf("Result: %d\n", num1 - num2); break;
               case '*': printf("Result: %d\n", num1 * num2); break;
               case '/':
                   if (num2 != 0)
                       printf("Result: %.2f\n", (double)num1 / num2);
                   else
                       printf("Error: Division by zero!\n");
                   break;
               case '%': printf("Result: %d\n", num1 % num2); break;
               case '^': printf("Result: %.2f\n", pow(num1, num2)); break;
               default: printf("Invalid operator!\n");
           }
          }
.|\n       { printf("Invalid input!\n"); }
%%

int main() {
    printf("Enter your expression (e.g., 5 + 3): ");
    yylex();  // Start lexical analysis
    return 0;
}


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



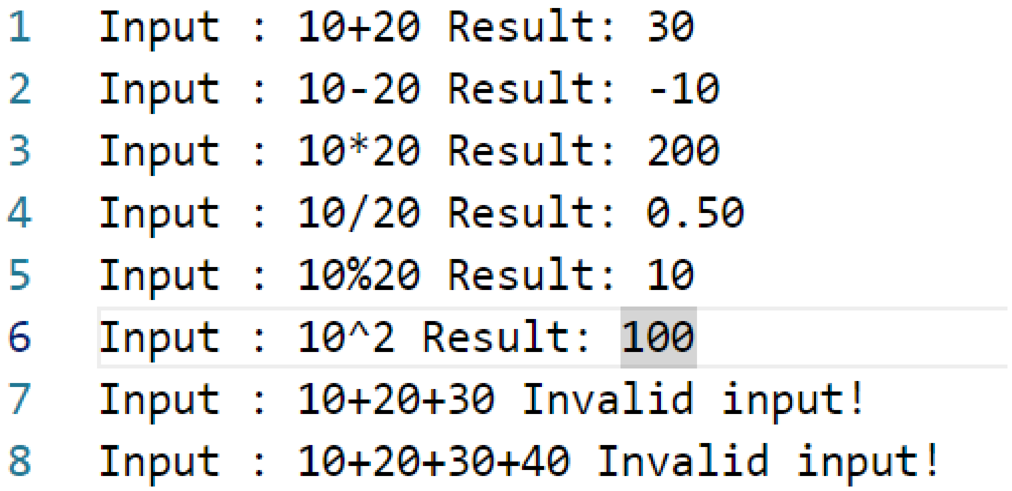
```
Input : Shahir! Passed
Input : Shahir! Passed
Input : BCSEthird# Failed
Input : Bcsethird# Passed
Input : Compiler######## Passed
```

Problem-1



```
Input : 100001101  Passed
Input : 000.....01101 Failed
Input :      Failed
Input : 100.....01111 Failed
Input : 1000011....01 Failed
Input : 100.....011101 Passed
```

Problem-2



```
Input : 10+20 Result: 30
Input : 10-20 Result: -10
Input : 10*20 Result: 200
Input : 10/20 Result: 0.50
Input : 10%20 Result: 10
Input : 10^2 Result: 100
Input : 10+20+30 Invalid input!
Input : 10+20+30+40 Invalid input!
```

Problem-3