Ganpat University | Institute of Computer Technology

॥ विद्यया समाजोत्कर्षः ॥

Name: Tushar Panchal

En.No: 21162101014

Sub: CD(Compiler Design)

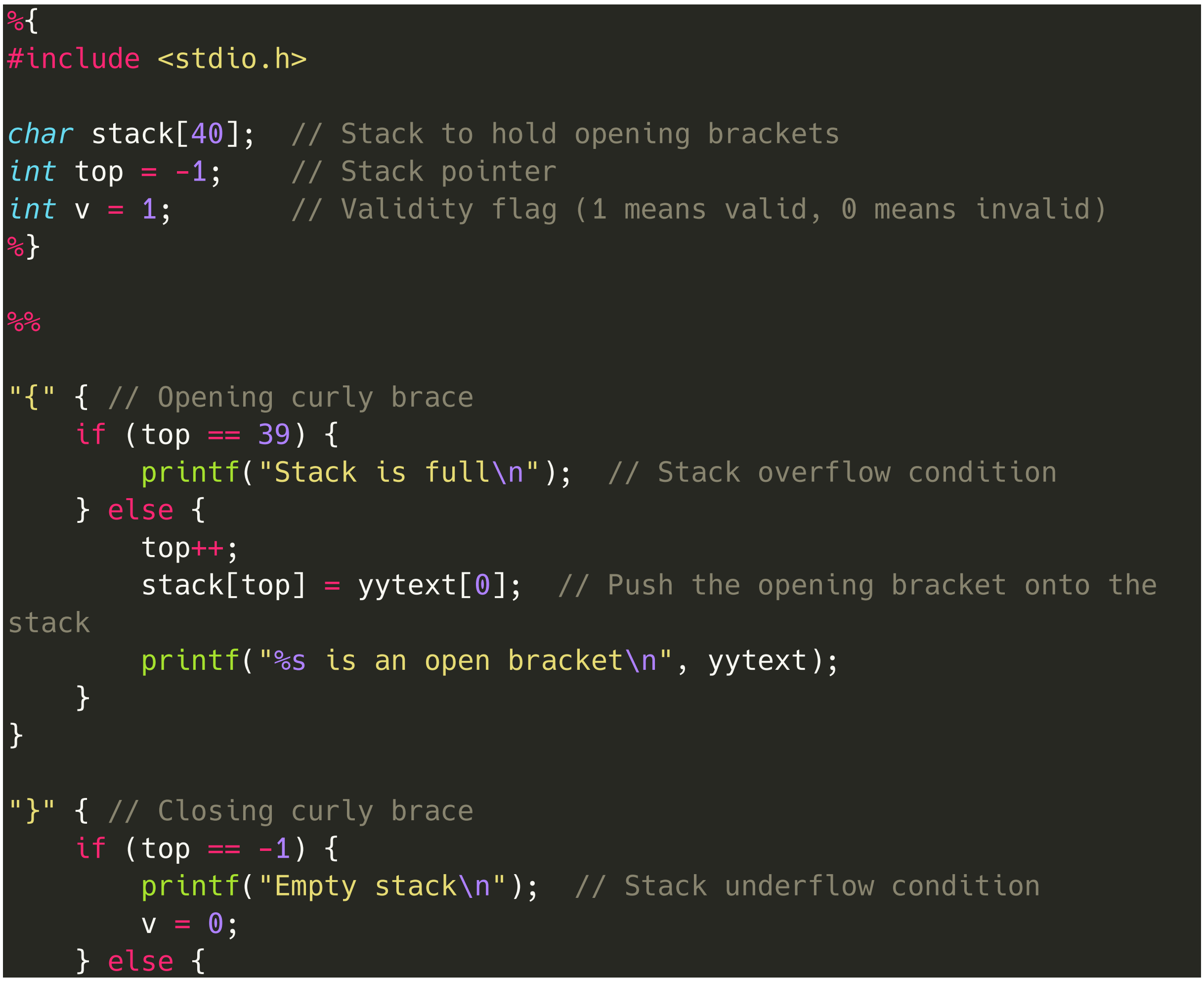
Branch: CBA

Batch:71

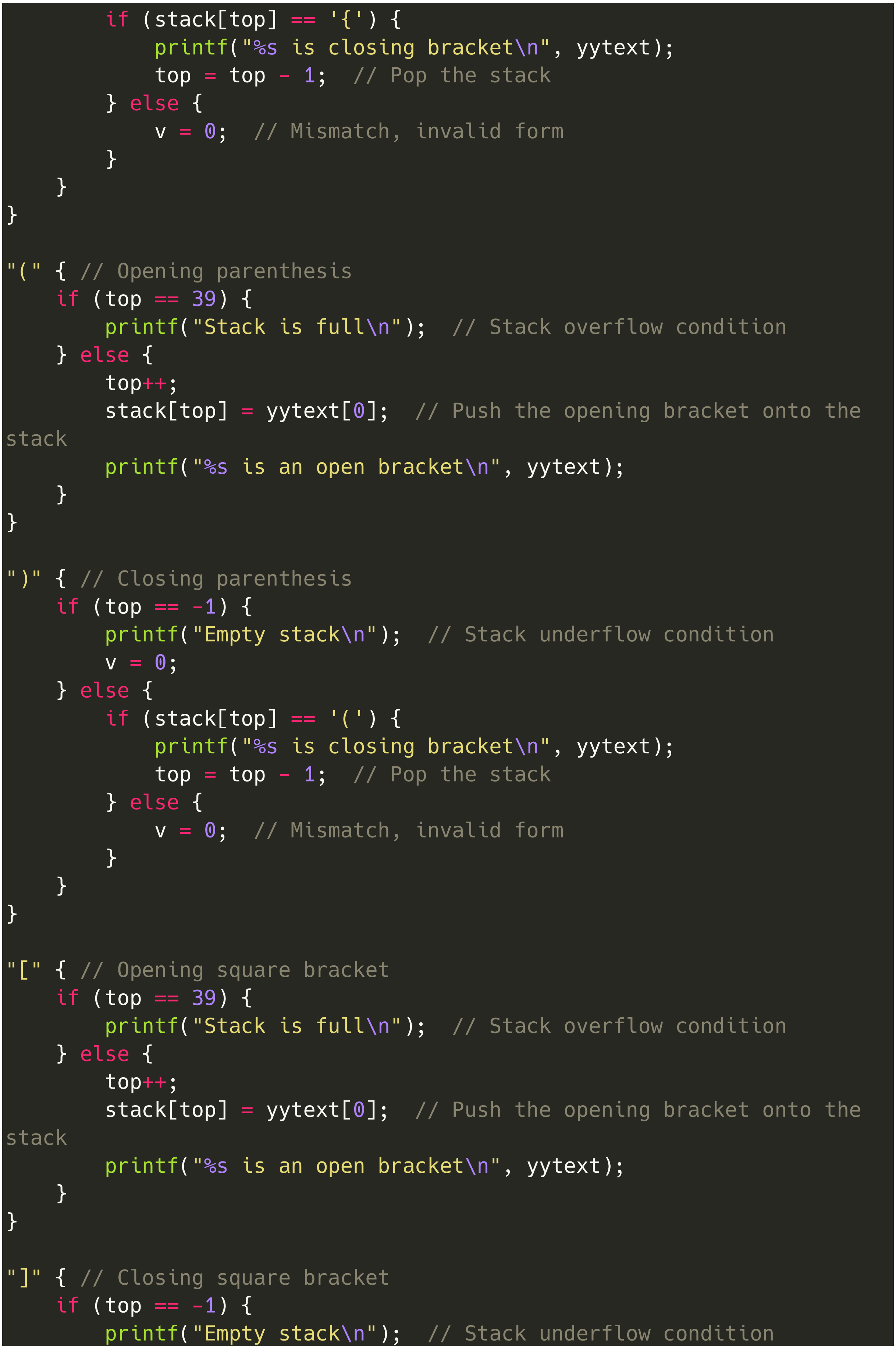
---------------------------PRACTICAL 03----------------------------

Write a LEX program for the well-defined parentheses.

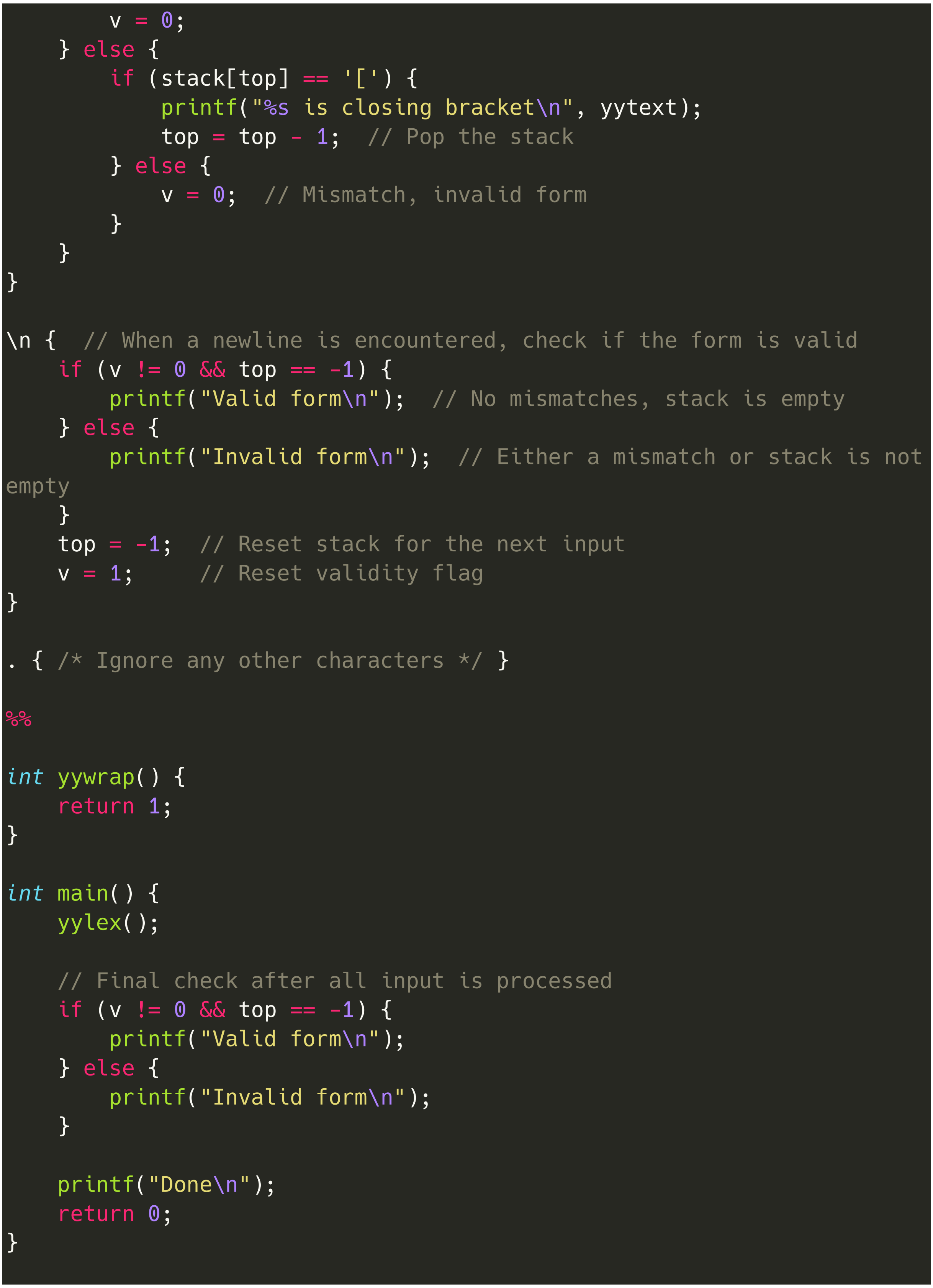
Source Code :



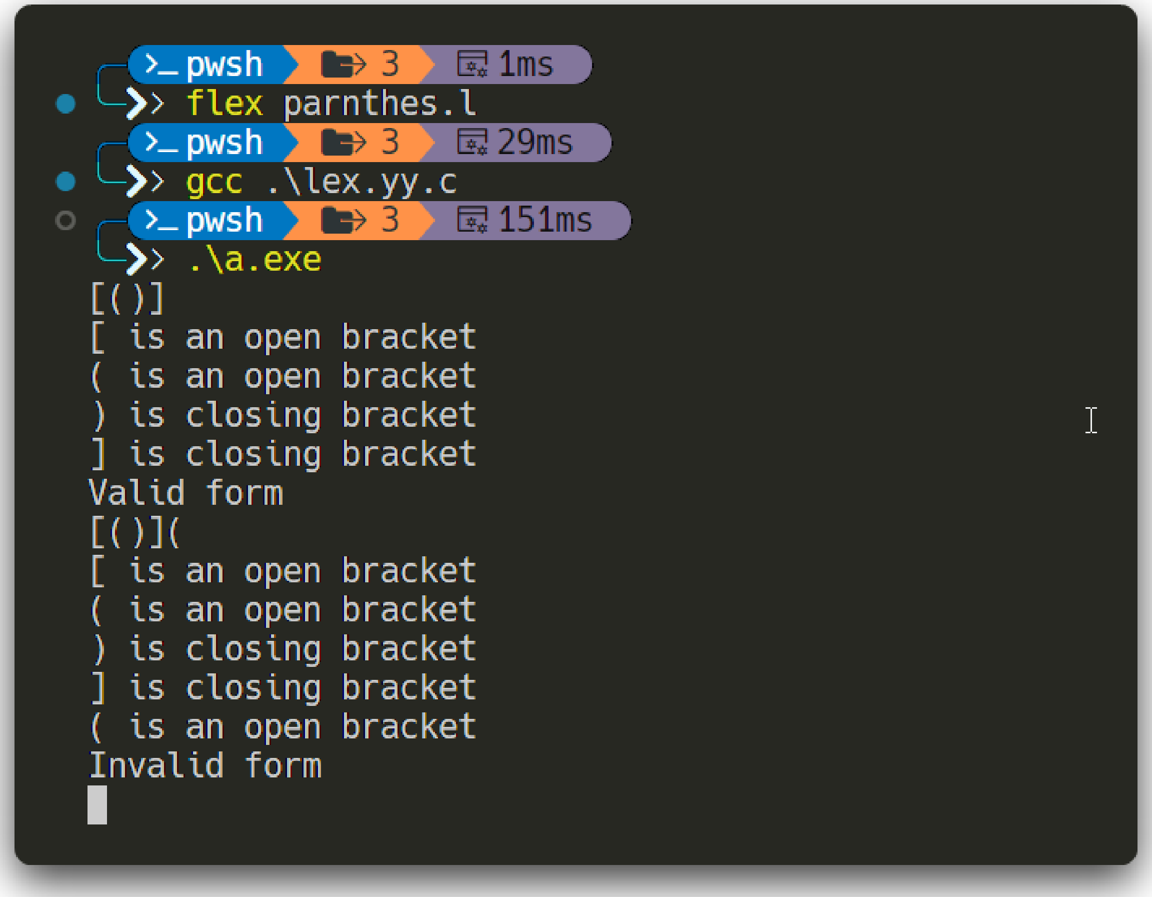
```
%{
#include <stdio.h>

char stack[40];  // Stack to hold opening brackets
int top = -1;    // Stack pointer
int v = 1;       // Validity flag (1 means valid, 0 means invalid)
%}

%%

"{" { // Opening curly brace
    if (top == 39) {
        printf("Stack is full\n");  // Stack overflow condition
    } else {
        top++;
        stack[top] = yytext[0];  // Push the opening bracket onto the
stack
        printf("%s is an open bracket\n", yytext);
    }
}

"}" { // Closing curly brace
    if (top == -1) {
        printf("Empty stack\n");  // Stack underflow condition
        v = 0;
    } else {
```



```
        if (stack[top] == '{') {
            printf("%s is closing bracket\n", yytext);
            top = top - 1;  // Pop the stack
        } else {
            v = 0;  // Mismatch, invalid form
        }
    }
}

"(" { // Opening parenthesis
    if (top == 39) {
        printf("Stack is full\n");  // Stack overflow condition
    } else {
        top++;
        stack[top] = yytext[0];  // Push the opening bracket onto the
stack
        printf("%s is an open bracket\n", yytext);
    }
}

")" { // Closing parenthesis
    if (top == -1) {
        printf("Empty stack\n");  // Stack underflow condition
        v = 0;
    } else {
        if (stack[top] == '(') {
            printf("%s is closing bracket\n", yytext);
            top = top - 1;  // Pop the stack
        } else {
            v = 0;  // Mismatch, invalid form
        }
    }
}

"[" { // Opening square bracket
    if (top == 39) {
        printf("Stack is full\n");  // Stack overflow condition
    } else {
        top++;
        stack[top] = yytext[0];  // Push the opening bracket onto the
stack
        printf("%s is an open bracket\n", yytext);
    }
}

"]" { // Closing square bracket
    if (top == -1) {
        printf("Empty stack\n");  // Stack underflow condition
```



```
        v = 0;
    } else {
        if (stack[top] == '[') {
            printf("%s is closing bracket\n", yytext);
            top = top - 1;  // Pop the stack
        } else {
            v = 0;  // Mismatch, invalid form
        }
    }
}

\n {  // When a newline is encountered, check if the form is valid
    if (v != 0 && top == -1) {
        printf("Valid form\n");  // No mismatches, stack is empty
    } else {
        printf("Invalid form\n");  // Either a mismatch or stack is not empty
    }
    top = -1;  // Reset stack for the next input
    v = 1;     // Reset validity flag
}

. { /* Ignore any other characters */ }

%%

int yywrap() {
    return 1;
}

int main() {
    yylex();

    // Final check after all input is processed
    if (v != 0 && top == -1) {
        printf("Valid form\n");
    } else {
        printf("Invalid form\n");
    }

    printf("Done\n");
    return 0;
}
```



```
>_ pwsh  3  1ms
>> flex parnthes.l
>_ pwsh  3  29ms
>> gcc .\lex.yy.c
>_ pwsh  3  151ms
>> .\a.exe
[()]
[ is an open bracket
( is an open bracket
) is closing bracket
] is closing bracket
Valid form
[()](
[ is an open bracket
( is an open bracket
) is closing bracket
] is closing bracket
( is an open bracket
Invalid form
```