

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 71
CD Practical 3

Write a LEX program for the well-defined parentheses

Code :

```
%{  
#include <stdio.h>  
#include <stdlib.h>  
  
#define MAXSTACK 1000  
  
// Stack implementation  
char stack[MAXSTACK];  
int top = -1;  
  
void push(char c) {  
    if (top ≥ MAXSTACK - 1) {  
        printf("Stack overflow\n");  
        exit(1);  
    }  
    stack[++top] = c;  
}  
  
char pop() {  
    if (top = -1) {  
        return '\0'; // Return null character if stack is empty  
    }  
    return stack[top--];  
}
```

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 71

CD Practical 3

```
int isEmpty() {
    return top == -1;
}

void check_balance() {
    if (isEmpty()) {
        printf("\n\tValid parentheses\n");
    } else {
        printf("\n\tNot valid parentheses\n");
    }
    exit(0);
}

int yywrap(void) {
    check_balance();
    return 1;
}

%}

%%

"(" { push('('); }
")" {
    if (isEmpty() || pop() != '(') {
        printf("\n\tNot valid parentheses\n");
        exit(0);
    }
}

"{" { push('{'); }
```

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 71

CD Practical 3

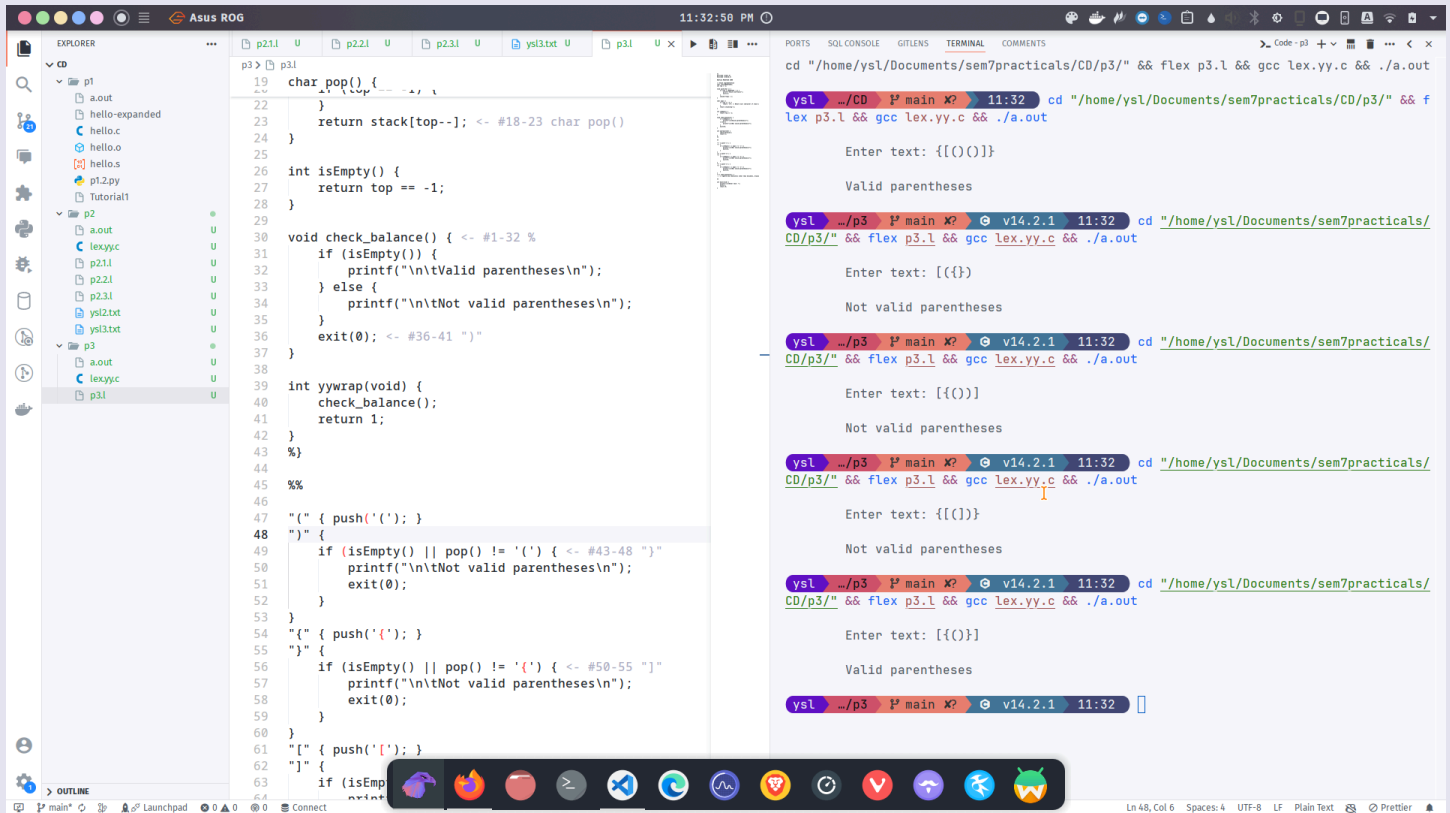
```
"}" {
    if (isEmpty() || pop() != '{') {
        printf("\n\tNot valid parentheses\n");
        exit(0);
    }
}
"[" { push('['); }
"]" {
    if (isEmpty() || pop() != '[') {
        printf("\n\tNot valid parentheses\n");
        exit(0);
    }
}
\n { check_balance(); }
. ; // Ignore any characters other than brackets, braces, and
parentheses

%%

int main(void) {
    printf("\n\tEnter text: ");
    yylex();
    return 0;
}
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 71
CD Practical 3

Output :



The screenshot displays a Visual Studio Code editor window with a C program in the main editor and its execution output in the terminal. The program, located in `p3/p3.c`, implements a stack-based algorithm to validate parentheses. It uses a character array `stack` to store opening parentheses. The `pop()` function removes an element from the stack, and `check_balance()` verifies if the stack is empty or if the closing parenthesis matches the top of the stack. The `yywrap()` function handles the end of input. The terminal shows the program being compiled and run, with test cases for valid and invalid parentheses sequences.

```
19 char pop() {
20     if (top < 0) return -1;
21     return stack[top--]; <- #18-23 char pop()
22 }
23
24 int isEmpty() {
25     return top == -1;
26 }
27
28 void check_balance() { <- #1-32 %
29     if (isEmpty()) {
30         printf("\n\tValid parentheses\n");
31     } else {
32         printf("\n\tNot valid parentheses\n");
33     }
34     exit(0); <- #36-41 %)
35 }
36
37 int yywrap(void) {
38     check_balance();
39     return 1;
40 }
41
42 %
43
44 %
45
46 {" { push('('); }
47 "}" {
48     if (isEmpty() || pop() != '(') { <- #43-48 %}
49         printf("\n\tNot valid parentheses\n");
50         exit(0);
51     }
52 } {" { push('('); }
53 "}" {
54     if (isEmpty() || pop() != '(') { <- #50-55 %}
55         printf("\n\tNot valid parentheses\n");
56         exit(0);
57     }
58 } {" { push('('); }
59 "}" {
60     if (isEmpty() || pop() != '(') { <- #50-55 %}
61         printf("\n\tNot valid parentheses\n");
62         exit(0);
63     }
```

Terminal Output:

```
cd "/home/ysl/Documents/sem7practicals/CD/p3/" && flex p3.l && gcc lex.yy.c && ./a.out
yyl main X? 11:32 cd "/home/ysl/Documents/sem7practicals/CD/p3/" && f
lex p3.l && gcc lex.yy.c && ./a.out
Enter text: {[(){}]}
Valid parentheses
yyl main X? 11:32 cd "/home/ysl/Documents/sem7practicals/CD/p3/" && flex p3.l && gcc lex.yy.c && ./a.out
Enter text: [(())]
Not valid parentheses
yyl main X? 11:32 cd "/home/ysl/Documents/sem7practicals/CD/p3/" && flex p3.l && gcc lex.yy.c && ./a.out
Enter text: {[(){}]}
Not valid parentheses
yyl main X? 11:32 cd "/home/ysl/Documents/sem7practicals/CD/p3/" && flex p3.l && gcc lex.yy.c && ./a.out
Enter text: {[(){}]}
Valid parentheses
```