

Lab4 Parentheses Checker

Write a program to read a text file and print whether or not the parentheses are balanced in the expression. (use **stack**)

1) The **data file**, lab4.txt should contain the following data:

1. (A + B) - { C + D } - [F + G]
2. (A * (B + (C * D + E)))
3. ((A + B)
4.) A + B (- C
5. (A + B)) - (C + D
6. (A+B)}
7. (A + B })
8. (A + B) - (C + D]}
9. { [A + B) - [(C -D)] }

2) Output

- | | |
|------------------------------------|------------------------------------|
| 1. (A + B) - { C + D } - [F + G] | Valid |
| 2. (A * (B + (C * D + E))) | Valid |
| 3. ((A + B) | Unbalanced parentheses |
| 4.) A + B (- C | Unbalanced parentheses |
| 5. (A + B)) - (C + D | Unbalanced parentheses |
| 6. (A+B)} | Unbalanced parentheses |
| 7. (A + B }) | Mismatched parentheses are (and } |
| 8. (A + B) - (C + D]} | Mismatched parentheses are (and] |
| 9. { [A + B) - [(C -D)] } | Mismatched parentheses are [and) |

Total: valid: 2 Unbalanced: 4 Mismatched: 3

- Invalid 의 경우 아래 두가지를 구분할 것
 - Parentheses 의 개수가 틀린 경우 (위 예제 3,4,5,6)
 - Parentheses 의 종류가 틀린 경우 (위 예제 7,8,9)
종류가 틀린 경우 “틀린 parentheses”를 명시할 것.

<< Lab4: Parentheses Checker >>

1. (A + B) - { C + D } - [F + G]
The Expression is Valid

2. (A * (B + (C * D + E)))
The Expression is Valid

3. ((A + B)
The Expression has unbalanced parentheses

4.) A + B (- C
The Expression has unbalanced parentheses

5. (A + B)) - (C + D
The Expression has unbalanced parentheses

6. (A+B)}
The Expression has unbalanced parentheses

7. (A + B })
The Mismatched Parenthes in the Expression are (and }

8. (A + B) - (C + D]}
The Mismatched Parenthes in the Expression are (and]

9. { [A + B) - [(C -D)] }
The Mismatched Parenthes in the Expression are [and)

Total: Balanced :2 Unbalanced :4 Mimatched : 3

계속하려면 아무 키나 누르십시오 . . .

Algorithm 참조:

```
void main() {  
    open data file // check file open error  
    while (infile.getline(buffer, 80)) {  
        validity = check (buffer );  
        if (validity is true) print “valid”  
    }  
    Print Total Number of each Parentheses;  
}
```

```
int check(exp) {  
    for(i=0; i<strlen(exp); i++){  
        if(exp[i]=='(' || exp[i]=='{' || exp[i]=='[')    push(exp[i]);  
        if(exp[i]==')' || exp[i]=='}' || exp[i]==']')  
            if(stack empty) {    print("UnBalanced ");  
            else { temp=pop();  
                    if(!match(temp, exp[i]))  
                        print("Mismatched"    temp, “and “ exp[i])  
            }  
        }  
    }  
    if(stack empty)    return true  
    else    return false // stack 에 parentheses 남아있는경우
```