

# YSC2221 Take-home Exam

Given an arithmetic expression, we would like to check if the parentheses are correct and matching. For example, “1+2-(3+4)” and “((1+3)-5-(6-2))” are correct expressions, but “((1+2)” is incorrect. (We follow our primary or secondary schools definitions.)

## Part a

Write a function to filter out the parentheses from a string input of an arithmetic expression. You don't have to care if the expression is really a valid arithmetic expression at this moment. Your function should **return** a string of only '(' and ')'.

Sample input:

```
test1 = '((4-((1+2)+3)))'
test2 = '(1+2)-(3+4)+(5-6)-(7-8)'
test3 = '((1+2)-(5+7))'
test4 = '((1+2)-(5+7))-('
```

Sample output:

```
>>> filterParentheses(test1)
'((((())))'
>>> filterParentheses(test2)
'()()()()'
>>> filterParentheses(test3)
'()()'
>>> filterParentheses(test4)
'()()()('
```

## Part b

Write a function `checkParentheses(s)` to check if the parentheses are matching. Your function should **return** True if they match in normal mathematical sense, and False otherwise. Sample output:

```
>>> checkParentheses(test1)
True
>>> checkParentheses(test2)
True
>>> checkParentheses(test3)
True
>>> checkParentheses(test4)
False
```

## Part c

Write a new function `checkMoreParentheses(s)` to do the check with two types of blankets, '(' and '['. Sample output:

```
>>> test = '(1+2-3+[4+5]-(5+[6+7]))'
>>> checkMoreParentheses(test)
True
>>> test = '(1+2-3+[4+5]-(5+[6+7]))'
>>> checkMoreParentheses(test)
False
```