

# Stack and Queue STL

ในโจทย์ข้อนี้คุณจะได้ใช้งาน Stack และ Queue ที่ C++ มีมาให้ใน STL หน้าที่คุณคือเขียนฟังก์ชัน `isBalanced(string s)` ในไฟล์ `main.cpp` โดยฟังก์ชันนี้จะคืนค่า `จริง` ถ้าวงเล็บในข้อความ `s` สมดุลตามนิยามในสไลด์ 251Ch4Stacks\_2.pdf และคืนค่า `เท็จ` ถ้าไม่สมดุล

## การใช้งาน STL Stack

การสร้าง Stack Object ที่เก็บค่าประเภท Character	<code>stack&lt;char&gt; st</code>
เพิ่มตัวอักษรเข้าไปใน Stack (ไม่มีการคืนค่ากลับมา)	<code>char ch1 = '(' st.push(c)</code>
เอาตัวอักษรบนสุดออกจาก Stack (ไม่มีการคืนค่ากลับมา)	<code>st.pop()</code>
ดูตัวบนสุดของ Stack	<code>char ch2 = st.top()</code>
เช็ค Stack ว่างหรือไม่ (คืนค่าเป็น Boolean)	<code>st.empty()</code>
ดึงค่าจำนวนข้อมูลใน Stack	<code>int stack_size = st.size()</code>

## การใช้งาน STL Queue

การสร้าง Queue Object ที่เก็บค่าประเภท Character	<code>queue&lt;char&gt; qe</code>
เพิ่มตัวอักษรเข้าไปใน Queue (ไม่มีการคืนค่ากลับมา)	<code>char ch1 = '(' qe.push(c)</code>
เอาตัวอักษรแรกสุดออกจาก Queue (ไม่มีการคืนค่ากลับมา)	<code>qe.pop()</code>
ดูตัวเก่าที่สุดของ Queue	<code>char ch2 = qe.front()</code>
เช็ค Queue ว่างหรือไม่ (คืนค่าเป็น Boolean)	<code>qe.empty()</code>
ดึงค่าจำนวนข้อมูลใน Queue	<code>int queue_size = qe.size()</code>

ในการบ้านนี้ดาวน์โหลดไฟล์การบ้านได้ที่ <https://github.com/CS-CMU/cs252student>

**ข้อมูลเข้าและออกสำหรับไฟล์ main.cpp ที่แจกให้**

ข้อมูลเข้ารับเป็น string ตัวเดียว ประกอบไปด้วยวงเล็บ 6 แบบได้แก่ ( ) [ ] { } จะรับข้อมูลจนกว่าจะเจอ -1 ถึงจะหยุดรัน ส่วนข้อมูลออกจะพิมพ์คำว่า Parentheses are balanced หรือ Parentheses are not balanced

ตัวอย่างข้อมูลเข้า	ตัวอย่างข้อมูลออก
(( ))	Parentheses are balanced
{()}[]	Parentheses are balanced
-1	
{()}[]	Parentheses are balanced
-1	
{()}[]	Parentheses are not balanced
-1	
{}) ( []	Parentheses are not balanced
-1	