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 | stack<char> st |
| เพิ่มตัวอักษรเข้าไปใน Stack  (ไม่มีการคืนค่ากลับมา) | char ch1 = '('  st.push(c) |
| เอาตัวอักษรบนสุดออกจาก Stack  (ไม่มีการคืนค่ากลับมา) | st.pop() |
| ดูตัวบนสุดของ Stack | char ch2 = st.top() |
| เช็คว่า Stack ว่างหรือไม่  (คืนค่าเป็น Boolean) | st.empty() |
| ดึงค่าจำนวนข้อมูลใน Stack | int stack\_size = st.size() |

**การใช้งาน STL Queue**

|  |  |
| --- | --- |
| การสร้าง Queue Object ที่เก็บค่าประเภท Character | queue<char> qe |
| เพิ่มตัวอักษรเข้าไปใน Queue  (ไม่มีการคืนค่ากลับมา) | char ch1 = '('  qe.push(c) |
| เอาตัวอักษรแรกสุดออกจาก Queue  (ไม่มีการคืนค่ากลับมา) | qe.pop() |
| ดูตัวเก่าที่สุดของ Queue | char ch2 = qe.front() |
| เช็คว่า Queue ว่างหรือไม่  (คืนค่าเป็น Boolean) | qe.empty() |
| ดึงค่าจำนวนข้อมูลใน Queue | int queue\_size = qe.size() |

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