

Practice Exercise: Stack

5 กุมภาพันธ์ 2568

แก้ String ให้ถูกต้อง

String ที่ถูกต้องคือ String ที่ไม่มีพยัญชนะสองตัวติดกันซึ่งมีเงื่อนไขต่อไปนี้

- $0 \leq i \leq s.length-2$
- $s[i]$ เป็นพยัญชนะพิมพ์เล็ก และ $s[i+1]$ เป็นพยัญชนะพิมพ์ใหญ่ตัวเดียวกัน หรือ
- $s[i]$ เป็นพยัญชนะพิมพ์ใหญ่ และ $s[i+1]$ เป็นพยัญชนะพิมพ์เล็กตัวเดียวกัน

วิธีแก้ String คือ ลบตัวอักษรคู่ที่ผิดออก

หมายเหตุ: String ว่าง ("") ถือเป็น String ที่ถูกต้อง

ข้อมูลนำเข้า (Input)

บรรทัดที่ 1	String ที่ต้องการแก้ไข
-------------	------------------------

ข้อมูลส่งออก (Output)

บรรทัดที่ 1	String ที่ถูกแก้แล้ว
-------------	----------------------

ตัวอย่างข้อมูลนำเข้า ส่งออก (Examples of Input & Output)

Input	Output
kmuuUtt	kmutt
WeLoVeEeCPeeE	WeLoVeCPE
abBAcC	//Empty string

หอคอยแห่งฮานอย

พีปิ่นได้เดินทางไปเซินเจิ้นเพื่อไปขอซื้อเกมหอคอยแห่งฮานอยจำนวน 1000 อันระหว่างเดินทางกลับพีปิ่นไปเล่นเกมใกล้รุ่งจึงอาจทำให้พีปิ่นไม่สามารถเล่นเกมได้ ให้น้องเขียนโปรแกรมเพื่อเล่นเกมฮานอยและหาจำนวนที่ขยับแผ่นดิสก์ในการเล่นแทนพีปิ่น

เกมหอคอยแห่งฮานอยคือเกมที่ประกอบด้วยแท่งสามแท่งที่มีดิสก์ที่มีขนาดต่างกันเป็นจำนวน n วัตถุประสงค์ของเกมนี้คือการย้ายดิสก์ทั้งหมดจากอันที่อยู่ซ้ายสุดไปยังอันขวาสุด โดยจะต้องทำตามกฎสามข้อนี้

1. สามารถย้ายดิสก์ได้ครั้งละหนึ่งดิสก์เท่านั้น
2. เฉพาะดิสก์บนสุดจากสแต็กหนึ่งเท่านั้นที่สามารถย้ายไปยังด้านบนของสแต็กอื่นหรือแท่งเปล่าได้
3. ไม่สามารถวางดิสก์ขนาดใหญ่ไว้ด้านบนของดิสก์ขนาดเล็กได้

ข้อมูลนำเข้า (Input)

บรรทัดที่ 1	Input จำนวนเต็ม x แทนดิสก์ในหอคอยฮานอย
-------------	--

ข้อมูลส่งออก (Output)

บรรทัดที่ 1 เป็นไปต้นไป	ขั้นตอนที่ใช้ในการดำเนินการแก้หอคอยแห่งฮานอยตามจำนวนดิสก์ที่กำหนด disks
บรรทัดสุดท้าย	จำนวนขั้นตอนที่ใช้ในการแก้หอคอยแห่งฮานอย

ตัวอย่างข้อมูลนำเข้า ส่งออก (Examples of Input & Output)

Input	Output
3	Move disk 1 from A to C Move disk 2 from A to B Move disk 1 from C to B Move disk 3 from A to C Move disk 1 from B to A Move disk 2 from B to C Move disk 1 from A to C Total moves: 7

เช็ควงเล็บ

พีธีต่อมีความเป๊ะสูงในทุกๆเรื่องโดยเฉพาะในการเขียนโปรแกรม การใช้วงเล็บทุกรูปแบบต้องมีการเปิดปิดวงเล็บ เพราะถ้าเปิดวงเล็บแล้วไม่ปิดก็จะรันไม่ได้ ให้เขียนโปรแกรมช่วยพีธีต่อในการเช็ควงเล็บที่พีธีป็นเขียนมาถูกต้องไหม

โดยที่วงเล็บมีดังนี้: (), {}, []

ข้อมูลนำเข้า (Input)

บรรทัดที่ 1	สตริงวงเล็บ
-------------	-------------

ข้อมูลส่งออก (Output)

บรรทัดที่ 1	ผลลัพธ์ของฟังก์ชัน True / False
-------------	---------------------------------

ตัวอย่างข้อมูลนำเข้า ส่งออก (Examples of Input & Output)

Input	Output
{[()]}	True
{{{tt[\$((qsc)	False

Infix to Postfix

พีธีมได้รับโจทย์คณิตศาสตร์จากดอกเตอร์ทวิชย์ นันทวิสุทธิวงศ์ แต่พีธีมคิดแบบ Infix ไม่เป็น จึงต้องการให้น้องๆช่วยแปลงโจทย์คณิตศาสตร์จาก Postfix เป็น Infix เพื่อให้พีธีมทำการบ้านของดอกเตอร์ทวิชย์ นันทวิสุทธิวงศ์ได้

โดยพีธีมได้ไปหา Pseudocode มาได้ดังนี้

1. Scan the infix expression from left to right
2. If the scanned character is an operand, put it in the postfix expression
 - If the precedence and associativity of the scanned operator are greater than the precedence and associativity of the operator in the stack [or the stack is empty or the stack contains a '('], then push it in the stack. ['^' operator is right associative and other operators like '+', '-', '*', and '/' are left-associative].
 - Check especially for a condition when the operator at the top of the stack and the scanned operator both are '^'. In this condition, the precedence of the scanned operator is higher due to its right associativity. So it will be pushed into the operator stack.
 - In all the other cases when the top of the operator stack is the same as the scanned operator, then pop the operator from the stack because of left associativity due to which the scanned operator has less precedence.
 - Else, Pop all the operators from the stack which are greater than or equal to in precedence than that of the scanned operator
 - After doing that Push the scanned operator to the stack. (If you encounter parenthesis while popping then stop there and push the scanned operator in the stack.)
3. If the scanned character is a ')', push it to the stack.
4. If the scanned character is a ')", pop the stack and output it until a '(' is encountered, and discard both the parenthesis.
5. Repeat steps 2-5 until the infix expression is scanned.
6. Once the scanning is over, Pop the stack and add the operators in the postfix expression until it is not empty.
7. Finally, print the postfix expression.

Infix to Postfix Conversion			
Expression = A + B * C / D - F + A ^ E			
Scanned Symbol	Stack	Output	Reason
A		A	Step 2
+	+	A	Step 3.1
B	+	AB	Step 2
*	+	AB	Step 3.1
C	+	ABC	Step 2
/	+/	ABC*	Step 3.2 / prec. is equal to * so not higher, so going to step 3.2
D	+/	ABC*D	Step 2
-	-	ABC*D/*	Step 3.2 / will be popped, added to o/p & then + popped & o/p - will be pushed
F	-	ABC*D/*F	Step 2
+	+	ABC*D/*F-	Step 3.2 - will be popped, added to o/p and then + to stack
A	+	ABC*D/*F-A	Step 2
^	^	ABC*D/*F-A	Step 2
E	^	ABC*D/*F-AE	Step 2
(empty)		ABC*D/*F-AE^+	Step 8

Figure 1: การแปลงจาก Infix เป็น Postfix

ข้อมูลนำเข้า (Input)

บรรทัดที่ 1	สตริงของโจทย์คณิตศาสตร์ในรูปแบบ Infix
-------------	---------------------------------------

ข้อมูลส่งออก (Output)

บรรทัดที่ 1	สตริงของโจทย์คณิตศาสตร์ในรูปแบบ Postfix
-------------	---

ตัวอย่างข้อมูลนำเข้า ส่งออก (Examples of Input & Output)

Input	Output
a+b	ab+
a+	a+
C+D*(A+B)	CDAB+*+