

CHAPTER 3 STACK

ผศ.ดร.สิดดา อินทรโสธรฉันท

สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น

พฤติกรรมพื้นฐานของข้อมูล

ในการจัดการข้อมูลจะมีรูปแบบการนำข้อมูลเข้าและออกหลัก ๆ อยู่ 2 แบบ คือ

แบบที่ 1 ข้อมูลที่เข้าก่อนออกก่อน (First – In – First – Out : FIFO) โครงสร้างที่มีลักษณะการทำงานแบบนี้ คือ คิว (Queue) ซึ่งจะมีทางเข้าออกแยกกันเหมือนท่อที่เข้าทางหนึ่งออกอีกทางหนึ่ง

แบบที่ 2 ข้อมูลที่เข้าทีหลังออกก่อน (Last – In – First – Out : LIFO) โครงสร้างที่มีลักษณะการทำงานแบบนี้คือ สแต็ก (Stack) ซึ่งจะมีทางเข้าออกทางเดียวเหมือนกล่องเมื่อใส่ของลงไปแล้ว การนำของออกจะนำของที่ใส่อยู่บนสุดออกก่อนนั่นเอง

คุณสมบัติของ STACK

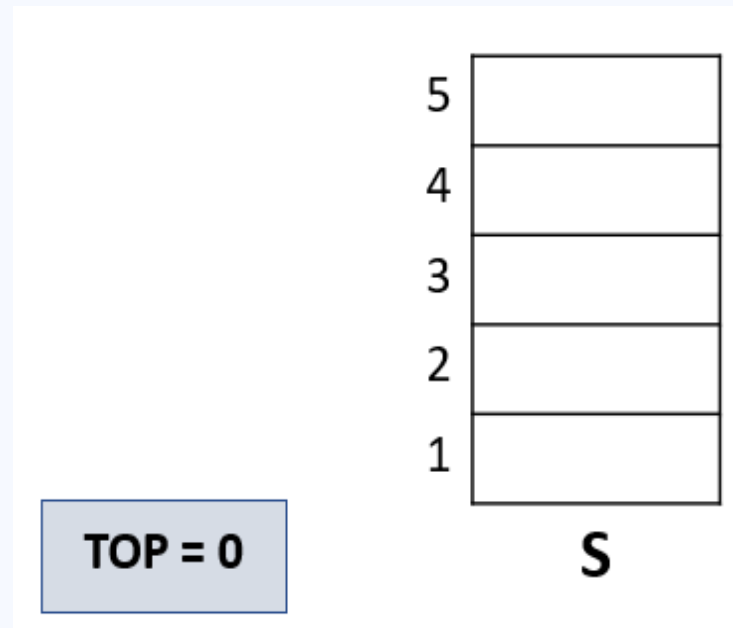
- หลักการทำงานของสแต็กจะเป็นการทำงานแบบข้อมูลที่เข้าทีหลังออกก่อน (LIFO)
- จากพฤติกรรมของสแต็กจะเห็นว่า สแต็กมีลักษณะคล้ายกล่องใส่ของที่มีทางเข้าออกทางเดียว การนำข้อมูลเข้าและออกจึงเกิดขึ้นที่ตำแหน่งข้อมูลที่เก็บล่าสุดหรือข้อมูลที่อยู่บนสุดนั่นเอง
- การทำงานจะเป็นการเข้าออกทางเดียว
- การนำข้อมูลเข้าและออกจะกระทำในตำแหน่งที่มีการนำข้อมูลเข้าล่าสุด หรือ ตำแหน่งที่เรียกว่า “top” of the stack

คุณสมบัติของ STACK

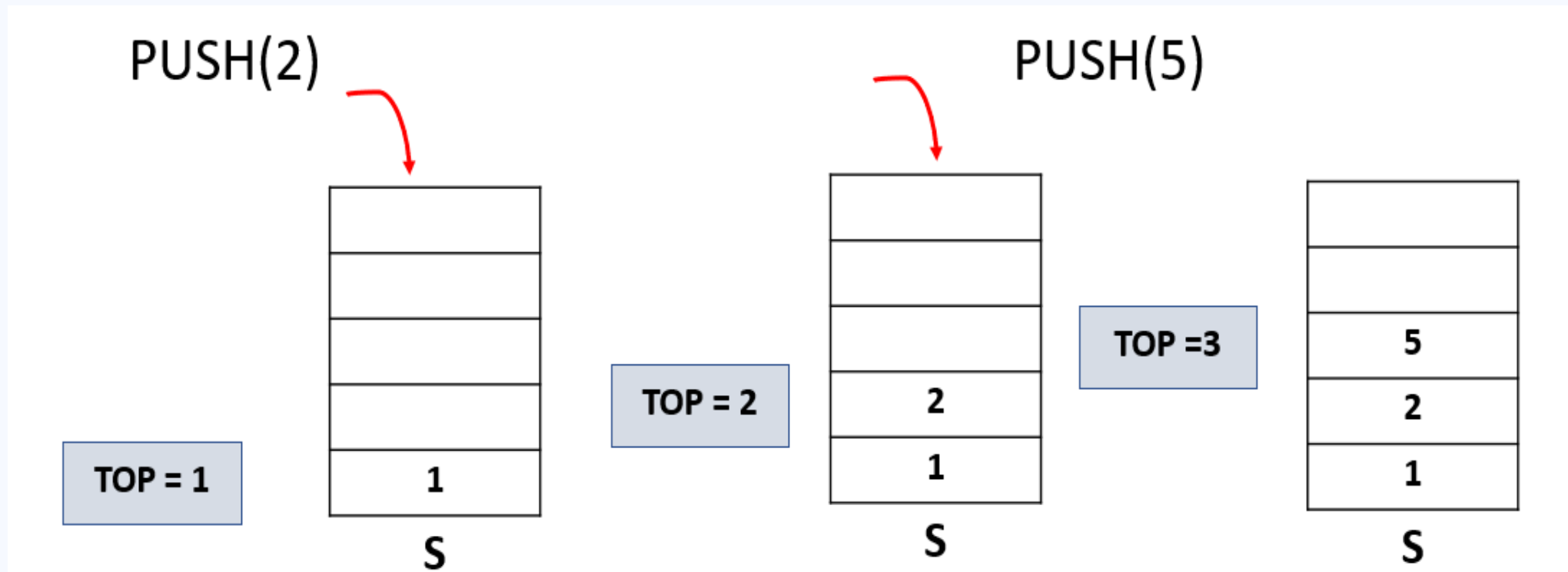
- การทำงานพื้นฐานของสแต็ก จะเรียกว่า “pushing” และ “popping”
- การ “pop” จะหมายถึงการนำข้อมูลในตำแหน่ง top ออกจากสแต็ก
- การนำข้อมูลใหม่ใส่เข้า Stack เรียกว่าการ “push” ข้อมูล โดยข้อมูลใหม่ที่ใส่เข้าไปก็จะไปต่อท้ายข้อมูลตำแหน่ง top และกลายเป็น top แทนค่าเดิม

คุณสมบัติของ STACK

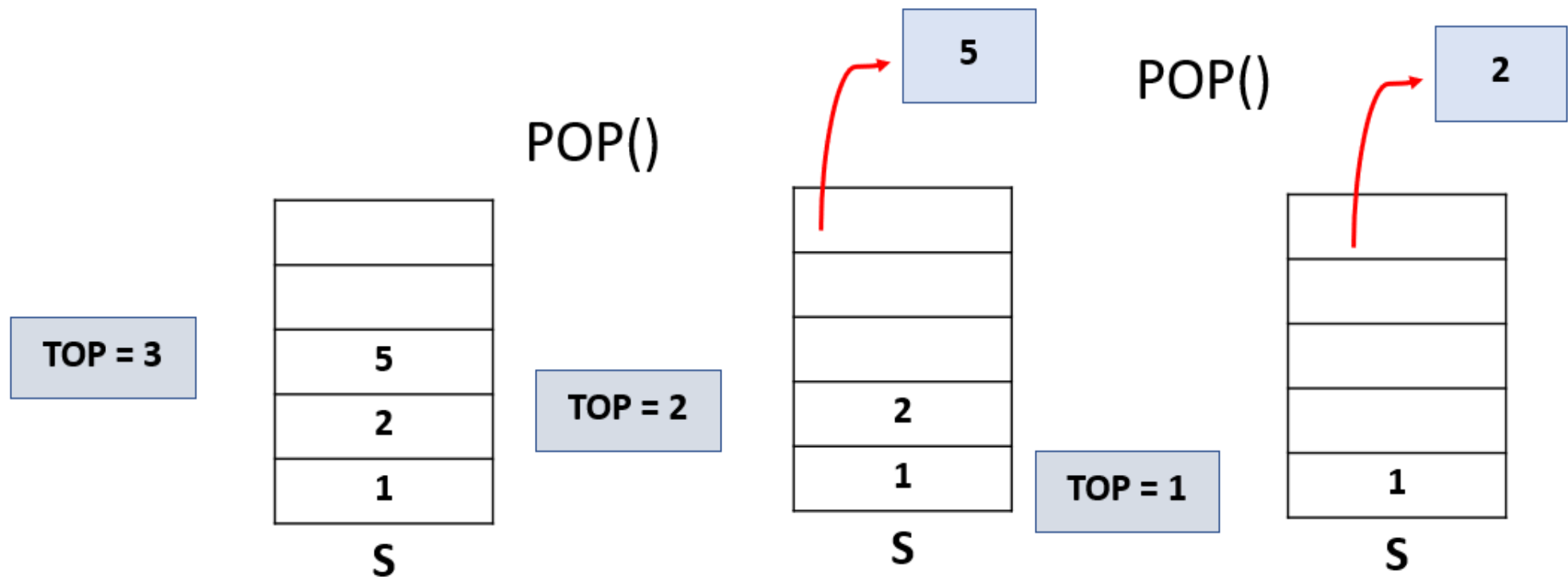
- สแต็ก S สามารถเก็บข้อมูลได้ 5 จำนวน ในตำแหน่งที่ 1 - 5 ในสถานะเริ่มต้นที่สแต็กยังไม่มีข้อมูลหรือสแต็กว่าง ตัวแปร TOP ที่ทำหน้าที่เก็บ index ของตำแหน่งข้อมูลตัวบนสุดจะมีค่าเป็น 0



คุณสมบัติของ STACK



คุณสมบัติของ STACK



การทำงานของ STACK บนโครงสร้างอาเรย์

- ในการสร้างอาเรย์จะต้องมีการกำหนดขนาดไว้
- ในการใช้งานหากต้องการใส่ข้อมูลเพิ่มลงไปจะต้องมีการตรวจสอบก่อนว่ามีพื้นที่ว่างเหลือหรือไม่ โดยสามารถตรวจสอบได้จากตัวแปร TOP
- หากตัวแปร TOP อยู่ตำแหน่งสุดท้ายของอาเรย์แล้วนั้นหมายถึง มีข้อมูลเต็มแล้ว
- หากมีข้อมูลเต็มแล้วจะไม่สามารถนำข้อมูลเข้าได้อีก

การทำงานของ STACK บนโครงสร้างอาร์เรย์

- การpushจะมีพารามิเตอร์ที่ต้องส่งมา 1 ตัว คือ ข้อมูลที่ต้องการเก็บลงสแต็ก
- ตัวแปรที่เกี่ยวข้องดังนี้
 - item จะเป็นข้อมูลที่ต้องการนำใส่สแต็ก
 - S เป็นอาร์เรย์ที่มีการทำงานแบบสแต็ก
 - ตัวแปร TOP ทำหน้าที่เก็บ index ของตำแหน่งของอาร์เรย์ที่มีข้อมูลตัวสุดท้ายในอาร์เรย์

การทำงานของ STACK บนโครงสร้างอาเรย์

PUSH (item)

1. ตรวจสอบว่าอาเรย์ S มีที่ว่างหรือไม่ โดยตรวจสอบว่า TOP อยู่ตำแหน่งสุดท้ายของอาเรย์ S หรือยัง
2. ถ้ายังมีที่ว่างให้ทำดังนี้
 - 2.1 ให้เพิ่มค่า TOP เพิ่ม 1 : $TOP = TOP + 1$
 - 2.2 ทำการใส่ข้อมูลลงตำแหน่งดังกล่าว : $S[TOP] = item$
2. ถ้าไม่มีที่ว่าง ไม่สามารถใส่ข้อมูลลงไปได้

การทำงานของ STACK บนโครงสร้างอาเรย์

- การป้อนจะไม่มีค่าพารามิเตอร์ใด ๆ มา เนื่องจากเป็นการนำข้อมูลตัวสุดท้ายของสแต็กออกเสมอ
- ตัวแปรที่เกี่ยวข้องดังนี้
 - item ทำหน้าที่เก็บข้อมูลที่นำออกจากสแต็ก
 - S เป็นอาเรย์ที่ใช้เก็บข้อมูลของ สแต็ก
 - ตัวแปร TOP ทำหน้าที่เก็บ index ของตำแหน่งของอาเรย์ที่มีข้อมูลตัวสุดท้ายในอาเรย์

การทำงานของ STACK บนโครงสร้างอาร์เรย์

POP() :

1. ตรวจสอบว่าในอาร์เรย์ S มีข้อมูลหรือไม่ โดยดูจากตัวแปร TOP ว่ามีค่ามากกว่าเท่ากับ 0 หรือไม่
2. ถ้ามี ให้ทำการคัดลอกข้อมูลที่ตำแหน่ง TOP ไว้ : $item = S[TOP]$
3. ปรับลดค่า TOP ลง 1 ตำแหน่ง : $TOP = TOP - 1$
4. ส่งค่าที่คัดลอกไว้กลับไป : $return\ item$

การประยุกต์ใช้ STACK

- โปรแกรม Internet Web browsers ใช้สแต็กในการเก็บแอดเดรสของเว็บไซต์ที่ผู้ใช้งานเข้าเยี่ยมชม เมื่อผู้ใช้งานเยี่ยมชมเว็บไซต์ใหม่ก็จะทำการpushแอดเดรสของเว็บไซต์ลงสแต็ก ทำให้ผู้ใช้สามารถย้อนกลับไปดูเว็บไซต์ก่อนหน้าได้ ซึ่งโปรแกรมจะไปทำการป๊อปข้อมูลแอดเดรสจากสแต็กที่เก็บข้อมูลไว้
- โปรแกรม Text editors ก็มีการใช้สแต็กในการทำงานของปุ่ม undo ที่ช่วยยกเลิกการทำงานกลับไปสถานะก่อนหน้าได้ โดยโปรแกรมจะทำการเก็บข้อความที่มีการเปลี่ยนแปลงลงในสแต็ก เมื่อผู้ใช้งานกดปุ่ม undo ก็จะมีการป๊อปข้อมูลออกมาเพื่อปรับเอกสารให้กลับไปสถานะก่อนหน้า

การประยุกต์ใช้ STACK

- การเขียนโปรแกรมแบบรีเคอร์ซีฟ ซึ่งจะมีการเรียกใช้ฟังก์ชันของตัวเองแบบซ้ำ ๆ ทุกครั้งที่มีการเรียกใช้จะทำการpushข้อมูลลงสแต็กจนเมื่อสิ้นสุดการเรียกซ้ำ ก็จะทำงานย้อนกลับโดยการป๊อปเอาข้อมูลจากสแต็กแล้วนำคำตอบที่ได้ก่อนแทนลงไป ทำเช่นนี้ซ้ำ ๆ จนป๊อปข้อมูลออกจากสแต็กหมด
- การคำนวณค่าของนิพจน์ต่าง ๆ ซึ่งในการคำนวณจะทำตามลำดับความสำคัญของเครื่องหมายทำให้การคำนวณไม่สามารถคำนวณเรียงตามลำดับของเครื่องหมายที่ปรากฏในนิพจน์ได้ ทำให้ต้องมีการจัดรูปนิพจน์ใหม่เพื่อง่ายต่อการทำงาน โดยการเปลี่ยนรูปนี้จะเป็นการเปลี่ยนรูปแบบนิพจน์จากอินฟิกซ์ (infix) ไปเป็นรูปแบบที่เรียกว่าโพสต์ฟิกซ์ (postfix)

การแปลงนิพจน์อินฟิกซ์เป็นนิพจน์โพสต์ฟิกซ์

- นิพจน์แบบอินฟิกซ์เป็นรูปแบบนิพจน์ที่มีโอเปอเรเตอร์จะอยู่ตรงกลางระหว่างตัวถูกดำเนินการ
- นิพจน์แบบโพสต์ฟิกซ์จะเป็นรูปแบบนิพจน์ที่มีโอเปอเรเตอร์อยู่ข้างหลังตัวถูกดำเนินการ โดยมีการพิจารณาลำดับการทำงานของเครื่องหมายด้วย

การแปลงนิพจน์อินฟิกซ์เป็นนิพจน์โพสต์ฟิกซ์

อินฟิกซ์	โพสต์ฟิกซ์
$A + B$	$A B +$
$X + Y * Z$	$X Y Z * +$
$(A + B) * C$	$A B + C *$

นิพจน์ในรูปแบบโพสต์ฟิกซ์ตำแหน่งของโอเปอเรเตอร์ที่ทำงานก่อนจะปรากฏก่อน ดังนั้นในการทำงานก็จะทำตามลำดับของโอเปอเรเตอร์ที่ปรากฏ

การแปลงนิพจน์อินฟิกซ์เป็นนิพจน์โพสต์ฟิกซ์

1. ถ้าข้อมูลเข้า (input) เป็นตัวถูกดำเนินการ (operand)ให้นำออกไปเป็นผลลัพธ์ (output)
2. ถ้าข้อมูลเข้าเป็นโอเปอเรเตอร์ (operator) ให้ดำเนินการดังนี้
 - 2.1 ถ้าสแต็กว่าง ให้pushโอเปอเรเตอร์นั้นลงในสแต็ก
 - 2.2 ถ้าสแต็กไม่ว่าง ให้เปรียบเทียบ โอเปอเรเตอร์ที่เข้ามา กับ โอเปอเรเตอร์ที่อยู่ในตำแหน่ง TOP ของสแต็ก

การแปลงนิพจน์อินฟิกซ์เป็นนิพจน์โพสต์ฟิกซ์

2.2.1 ถ้าโอเปอเรเตอร์ที่เข้ามามีความสำคัญมากกว่า โอเปอเรเตอร์ที่ตำแหน่ง TOP ของสแต็กให้พুষ
โอเปอเรเตอร์นั้นลงสแต็ก

2.2.2 ถ้าโอเปอเรเตอร์ที่เข้ามามีความสำคัญน้อยกว่าหรือเท่ากับ โอเปอเรเตอร์ที่อยู่ในตำแหน่ง TOP ของสแต็ก ให้ ป๊อปโอเปอเรเตอร์นั้นออกไปเป็นผลลัพธ์ แล้วทำการเปรียบเทียบโอเปอเรเตอร์ที่เข้ามากับโอเปอเรเตอร์ที่ตำแหน่ง TOP ต่อไป จะหยุดจนกว่า โอเปอเรเตอร์ที่เข้ามามีความสำคัญมากกว่า โอเปอเรเตอร์ที่ตำแหน่ง TOP ของสแต็ก แล้วจึง พুষโอเปอเรเตอร์ที่เข้ามานั้นลงสแต็ก

การแปลงนิพจน์อินฟิกซ์เป็นนิพจน์โพสต์ฟิกซ์

3. ถ้าข้อมูลเข้าเป็นวงเล็บเปิด ให้พูลงสแต็ก
4. ถ้าข้อมูลเข้าเป็นวงเล็บปิด ให้ป๊อปข้อมูลออกจากสแต็กไปเป็นผลลัพธ์จนกว่าจะถึงวงเล็บเปิด จากนั้นทิ้งวงเล็บเปิดและปิดไป
5. ถ้าข้อมูลเข้าหมดให้ป๊อปข้อมูลออกจากสแต็กไปเป็นผลลัพธ์จนกว่าสแต็กจะว่าง

การแปลงนิพจน์อินฟิक्सเป็นนิพจน์โพสต์ฟิक्स

- ในการคำนวณค่าของนิพจน์จะมีการทำงานตามลำดับความสำคัญของโอเปอเรเตอร์ ซึ่งลำดับความสำคัญจะเป็นตัวบอกว่าเครื่องหมายใดต้องทำก่อน

ความสำคัญ	ลำดับการทำงาน	เครื่องหมายโอเปอเรเตอร์	ลำดับการทำงาน
มาก ↓ น้อย	1	วงเล็บ	จากซ้ายไปขวา
	2	ยกกำลัง	จากขวาไปซ้าย
	3	คูณ (*) หาร (/)	จากซ้ายไปขวา
	4	บวก (+) ลบ (-)	จากซ้ายไปขวา

ตัวอย่างการแปลงนิพจน์อินฟิกซ์เป็นนิพจน์โพสต์ฟิกซ์

ตัวอย่างที่ 1 จากนิพจน์ $A + B * C$ จงทำการแปลงไปเป็นนิพจน์โพสต์ฟิกซ์

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
A	“A” เป็นตัวถูกดำเนินการจึงนำออกไปเป็นผลลัพธ์		A
+	“+” เป็นโอเปอเรเตอร์ จะพิจารณาข้อมูลในสแต็ก ซึ่งว่าง จึงทำการpushลงสแต็ก	+	A

ตัวอย่างการแปลงนิพจน์อินฟิกซ์เป็นนิพจน์โพสต์ฟิกซ์

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
B	“B” เป็นตัวถูกดำเนินการจึงนำออกไปเป็นผลลัพธ์	+	A B
*	“*” เป็นโอเปอเรเตอร์ จะพิจารณาข้อมูลในสแต็ก ซึ่ง “*” <u>มีความสำคัญมากกว่า</u> จึงpush “*” ลงสแต็ก	+ *	A B

ตัวอย่างการแปลงนิพจน์อินฟิกซ์เป็นนิพจน์โพสต์ฟิกซ์

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
C	“C” เป็นตัวถูกดำเนินการจึงนำออกไปเป็นผลลัพธ์	+ *	A B C
	ข้อมูลเข้าหมดแล้ว ทำการป้อนข้อมูลในสแต็กออกให้หมด	+ *	A B C * +

จากนิพจน์ $A + B * C$ แปลงไปเป็นนิพจน์โพสต์ฟิกซ์ได้เป็น A B C * +

ตัวอย่างการแปลงนิพจน์อินฟิกซ์เป็นนิพจน์โพสต์ฟิกซ์

ตัวอย่างที่ 2 จากนิพจน์ $(A + B - C) * D / E$ จงทำการแปลงไปเป็นนิพจน์โพสต์ฟิกซ์

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
(ข้อมูลเข้าเป็นวงเล็บเปิด ให้pushเข้าสแต็ก	(
A	“A” เป็นตัวถูกดำเนินการจึงนำออกไปเป็นผลลัพธ์	(A

ตัวอย่างการแปลงนิพจน์อินฟิกซ์เป็นนิพจน์โพสต์ฟิกซ์

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
+	“+” เป็นโอเปอเรเตอร์ จะพิจารณาข้อมูลในสแต็ก ซึ่งยังไม่มีโอเปอเรเตอร์ในการคำนวณ จึงทำการ pushลงสแต็ก	(+	A
B	“B” เป็นตัวถูกดำเนินการจึงนำออกไปเป็นผลลัพธ์	(+	A B

ตัวอย่างการแปลงนิพจน์อินฟิกซ์เป็นนิพจน์โพสต์ฟิกซ์

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
-	“-” เป็นโอเปอเรเตอร์ จะพิจารณาข้อมูลในสแต็ก ซึ่ง “+” <u>มีความสำคัญเท่ากัน</u> จึงต้องป๊อป ออกจากสแต็กก่อน ต่อมาจะเหลือ “(” ซึ่งไม่ใช่โอเปอเรเตอร์ในการคำนวณ จึงหยุดการเปรียบเทียบแล้วพুষ “-” ลงไป	(-	A B +

ตัวอย่างการแปลงนิพจน์อินฟิกซ์เป็นนิพจน์โพสต์ฟิกซ์

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
C	“C” เป็นตัวถูกดำเนินการจึงนำออกไปเป็นผลลัพธ์	(-	A B + C
)	ข้อมูลเข้าเป็นวงเล็บปิด ให้ป้อนข้อมูลออกจาก สแต็กไปเป็นผลลัพธ์จนกว่าจะถึงวงเล็บ เปิด จากนั้นทิ้งวงเล็บเปิดและปิดไป		A B + C -

ตัวอย่างการแปลงนิพจน์อินฟิกซ์เป็นนิพจน์โพสต์ฟิกซ์

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
*	“*” เป็นโอเปอเรเตอร์ จะพิจารณาข้อมูลในสแต็ก ซึ่งว่าง จึงทำการpushลงสแต็ก	*	
D	“D” เป็นตัวถูกดำเนินการจึงนำออกไปเป็นผลลัพธ์	*	A B + C - D

ตัวอย่างการแปลงนิพจน์อินฟิกซ์เป็นนิพจน์โพสต์ฟิกซ์

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
/	“/” เป็นโอเปอเรเตอร์ จะพิจารณาข้อมูลในสแต็ก ซึ่ง “*” มี <u>ความสำคัญเท่ากัน</u> จึงต้องป๊อป ออกจากสแต็กก่อน แล้วพুষ “/” ลงไป	/	A B + C – D *
E	“E” เป็นตัวถูกดำเนินการจึงนำออกไปเป็นผลลัพธ์	/	A B + C – D * E

ตัวอย่างการแปลงนิพจน์อินฟิกซ์เป็นนิพจน์โพสต์ฟิกซ์

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
	ข้อมูลเข้าหมดแล้ว ทำการป้อนข้อมูลในสแต็กออกให้หมด		$A B + C - D * E /$

จากนิพจน์ $(A + B - C) * D / E$ แปลงไปเป็นนิพจน์โพสต์ฟิกซ์ได้เป็น $A B + C - D * E /$

การนำนิพจน์โพสต์ฟิกซ์ไปใช้งาน

1. ถ้าข้อมูลเข้า (input) เป็นตัวถูกดำเนินการ (operand) ให้ให้ทำการพูลงสแต็ก
2. ถ้าข้อมูลเข้าเป็นโอเปอเรเตอร์ (operator) ให้ดำเนินการดังนี้
 - 2.1 ให้ทำการป้อนค่าในสแต็กออกมา 2 ค่า โดยค่าแรกจะเป็นตัวถูกดำเนินการทางขวาของโอเปอเรเตอร์ ส่วนค่าที่ 2 ที่ป้อนออกมาจะเป็นตัวถูกดำเนินการทางซ้ายของโอเปอเรเตอร์
 - 2.2 ทำการหาคำตอบแล้วนำคำตอบที่ได้พูลงสแต็ก
3. ถ้าข้อมูลเข้าหมด ให้ป้อนข้อมูลออกจากสแต็กจะกลายเป็นผลลัพธ์

ตัวอย่างการนำนิพจน์โพสต์ฟิกซ์ไปใช้งาน

จากตัวอย่างที่ 1 จงหาคำตอบของนิพจน์โพสต์ฟิกซ์ $A \ B \ C \ * \ +$ กำหนดค่า $A = 5 \ B = 3 \ C = 7$

จากค่าของ $A \ B$ และ C แทนลงในนิพจน์โพสต์ฟิกซ์จะได้ $(5) \ (3) \ (7) \ * \ +$

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
(5)	“5” เป็นตัวถูกดำเนินการจึงpushลงสแต็ก	(5)	
(3)	“3” เป็นตัวถูกดำเนินการจึงpushลงสแต็ก	(5) (3)	

ตัวอย่างการนำนิพจน์โพสต์ฟิกซ์ไปใช้งาน

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
(7)	“7” เป็นตัวถูกดำเนินการจึงpushลงสแต็ก	(5) (3) (7)	
*	“*” เป็นโอเปอเรเตอร์ จะป้อนค่าออกจากสแต็ก 2 ค่า เพื่อมาคำนวณหาคำตอบจะได้ $(3) * (7) = 21$ ทำการpushคำตอบลงสแต็ก	(5) (21)	

ตัวอย่างการนำนิพจน์โพสต์ฟิกซ์ไปใช้งาน

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
+	“+” เป็นโอเปอเรเตอร์ จะป้อนค่าออกจากสแต็ก 2 ค่า เพื่อมาคำนวณหาคำตอบจะได้ $(5) + (21) = 26$ ทำการป้อนคำตอบลงสแต็ก	(26)	
	ข้อมูลเข้าหมดแล้ว ทำการป้อนข้อมูลในสแต็กจะได้คำตอบ		26

จากนิพจน์โพสต์ฟิกซ์ A B C * + เมื่อทำการแทนค่าตัวแปรลงไปจะได้คำตอบเป็น 26

ตัวอย่างการนำนิพจน์โพสต์ฟิกซ์ไปใช้งาน

จากตัวอย่างที่ 2 จงหาคำตอบของนิพจน์โพสต์ฟิกซ์ $A B + C - D * E /$ กำหนดค่า $A = 7$ $B = 3$ $C = 5$ $D = 4$ $E = 10$ จากค่าของ A B และ C แทนลงในนิพจน์โพสต์ฟิกซ์จะได้ $(7) (3) + (5) - (4) * (10) /$

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
(7)	“7” เป็นตัวถูกดำเนินการจึงpushลงสแต็ก	(7)	
(3)	“3” เป็นตัวถูกดำเนินการจึงpushลงสแต็ก	(7) (3)	

ตัวอย่างการนำนิพจน์โพสต์ฟิกซ์ไปใช้งาน

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
+	“+” เป็นโอเปอเรเตอร์ จะป้อนค่าออกจากสแต็ก 2 ค่า เพื่อมาคำนวณหาคำตอบจะได้ $(7) + (3) = 10$ ทำการpushคำตอบลงสแต็ก	(10)	
(5)	“5” เป็นตัวถูกดำเนินการจึงpushลงสแต็ก	(10) (5)	

ตัวอย่างการนำนิพจน์โพสต์ฟิกซ์ไปใช้งาน

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
-	“-” เป็นโอเปอเรเตอร์ จะป้อนค่าออกจากสแต็ก 2 ค่า เพื่อมาคำนวณหาคำตอบจะได้ $(10) - (5) = 5$ ทำการpushคำตอบลงสแต็ก	(5)	
(4)	“4” เป็นตัวถูกดำเนินการจึงpushลงสแต็ก	(5) (4)	

ตัวอย่างการนำนิพจน์โพสต์ฟิกซ์ไปใช้งาน

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
*	“*” เป็นโอเปอเรเตอร์ จะป้อนค่าออกจากสแต็ก 2 ค่า เพื่อมาคำนวณหาคำตอบจะได้ $(5) * (4) = 20$ ทำการpushคำตอบลงสแต็ก	(20)	
(10)	“10” เป็นตัวถูกดำเนินการจึงpushลงสแต็ก	(20) (10)	

ตัวอย่างการนำนิพจน์โพสต์ฟิกซ์ไปใช้งาน

นิพจน์อินฟิกซ์ ข้อมูลเข้า	ขั้นตอนการทำงาน	ข้อมูลใน สแต็ก	นิพจน์โพสต์ฟิกซ์ ข้อมูลออก
/	“/” เป็นโอเปอเรเตอร์ จะป้อนค่าออกจากสแต็ก 2 ค่า เพื่อมาคำนวณหาคำตอบจะได้ $(20) * (10) = 2$ ทำการpushคำตอบลงสแต็ก	(2)	
	ข้อมูลเข้าหมดแล้ว ทำการป้อนข้อมูลในสแต็กจะได้คำตอบ		2

จากนิพจน์โพสต์ฟิกซ์ $A B + C - D * E /$ เมื่อทำการแทนค่าตัวแปรลงไปจะได้คำตอบเป็น 2