

Stack and Queue

Chapter 03

รหัสนักศึกษา

66070014

ชื่อจริงและนามสกุล

MAR

San



จงประมวลผลคำสั่งต่อไปนี้ พร้อมวาดภาพผลลัพธ์ของ Stack เมื่อทำทุกคำสั่งเสร็จหมดแล้ว

- กำหนด S_1 และ S_2 เป็น Empty Stack
- pushStack(s₁, 6)
- \triangleright pushStack(s₁, 5)
- \triangleright pushStack(s_1 , 4)
- \triangleright pushStack(s₁, 3)
- pushStack(s₁, 2)
- pushStack(s₁, 1)
- Loop (not emptyStack(s1))

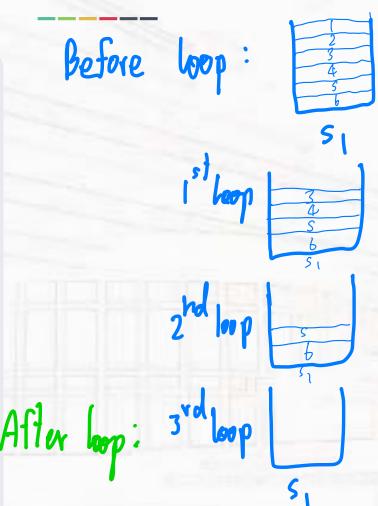
popStack(s₁, x)

 $popStack(s_1, x)$

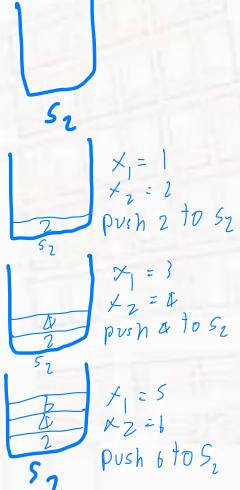
pushStack(s2,x)

end Loop











ให้วาดรูปผลลัพธ์ของ queue Q เมื่อได้รับข้อมูลและมีการ ทำงานต่อไปนี้

กำหนดข้อมูลคือ: 9, 72, 1, 43, 29, 0, 34, 62, 3, 56, 0, 34

```
createQueue(Q)
loop (not end of file)
read number provided if (number is greater than 5)
enqueue(Q, number)
else
queueRear(Q, x) > ham no moder
enqueue(Q, x) > ham no moder
end if
end loop
```

Exercise 2



```
enqueue (Q, 9), rom 75
read 72, enqueul (Q, 72), num 75
         engulue (Q, 43), num > 5
     29, enguere (Q,29), num 75
     0, engulue (Q, 29), nom 25
Year
     34, enqueue la, 341, mm 75
     62, enqueve (Q, 62), mm 75
   3, enqueve (Q,62), num Ls
     Sb, enqueul (Q, sb), num >5
YEAR
      0, enqueue (Q,S), nrm cs
      341 enqueve (Q.34), num 75
```



Exercise 3



จงวาดภาพแสดงการแปลง			ession
Infix	stack	postix	
*(B+c)-D*f-E		A	
C13+C)-0*F-E	*	A	
β+()-D*F-E	×	A	
+ c)-D*F-E	<u> </u>	AB	H
c)-D*F-E	*	AB	
-D*F-E		ABCT	
DXF-E		ABC+*	

ดยใช้ Stack	กำหนดข้อมูลคือ: <i>A * (B + C) - D * F - E</i>		
Infix	stack postix		
7 * F-E	ABC+*)		
F-E	ABC+*D		
-6	ABC+*DF		
E	ABC+*DF*-		
	ABC+*DF*-E		
	ABC+*DF*-E-		

ous: ABC+*DF*-E-



Exercise 4-5



ให้เขียนอัลกอริทึม copyStack(stack₁, stack₂)

ที่ใช้คัดลอกข้อมูลของ stack_1 ให้กับ stack_2 ให้มีลำดับข้อมูลที่เหมือนกัน

copyStack Cstack, Stack)
create Empty Stack (copy) While (stack 2 not empty) end loop
while (stack (not empty)

popstack (stack (, x))

popstack (stack (, x))

popstack (stack (copy, x))

end loop
while (copy no) empty) push stack cstack 2 (x)

ให้เขียนอัลกอริทึม concatQueue(Q_1, Q_2)

- ullet นำข้อมูลของ Q_1 ต่อท้ายด้วยข้อมูลของ Q_2 แล้วนำผลลัพธ์ที่ได้เก็บไว้ใน Q_1
- ใช้คำสั่งของ queue ADT ได้และ Q₂ ยังคงเหมือนเดิม

