Quiz 1

- จงประมวลผลคำสั่งต่อไปนี้ พร้อมวาดภาพ Stack ผลลัพธ์เมื่อทำทุกคำสั่งเสร็จแล้ว (กำหนด s1 และ s2 เป็น Empty Stack)
  - pushStack(s1,6)
  - pushStack(s1,5)
  - pushStack(s1,4)
  - pushStack(s1,3)
  - pushStack(s1,2)
  - pushStack(s1,1)
  - Loop not emptyStack(s1)

    popStack(s1,x)

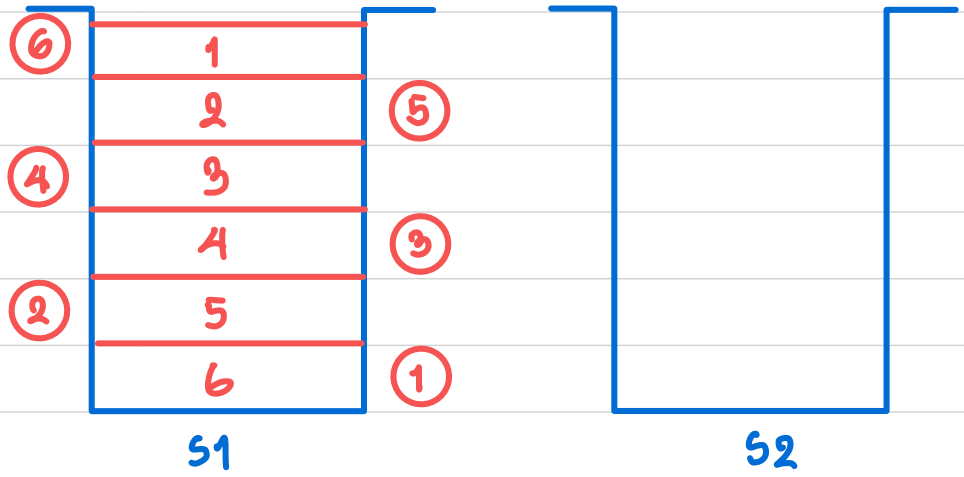
    popStack(s1,x)

    pushStack(s2,x)

    end Loop

(กำหนด s1 และ s2 เป็น Empty Stack)

1. pushStack(s1,6)
2. pushStack(s1,5)
3. pushStack(s1,4)
4. pushStack(s1,3)
5. pushStack(s1,2)
6. pushStack(s1,1)



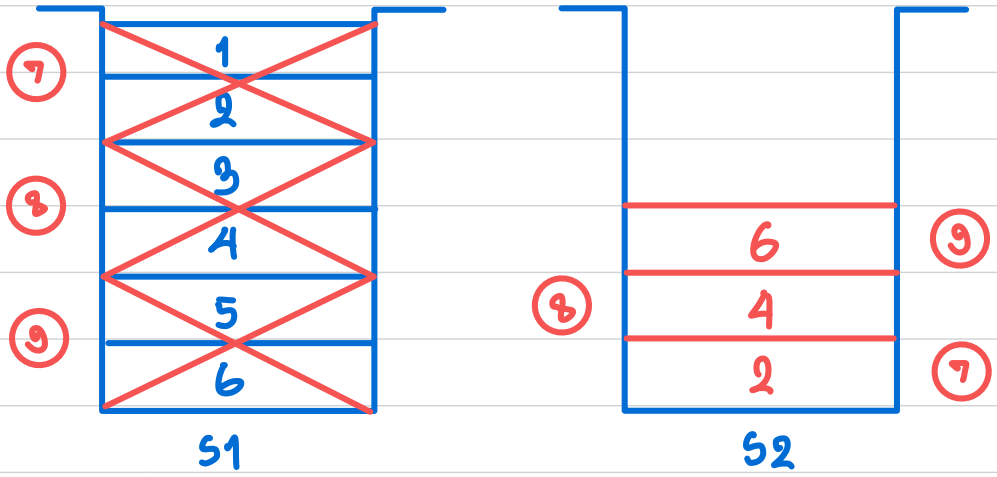
Loop not emptyStack(s1)

popStack(s1,x)

popStack(s1,x)

pushStack(s2,x)

end Loop



7. 1st loop : x = 1
   x = 2
8. 2nd loop : x = 3
   x = 4
9. 3rd loop : x = 5
   x = 6

x = 6

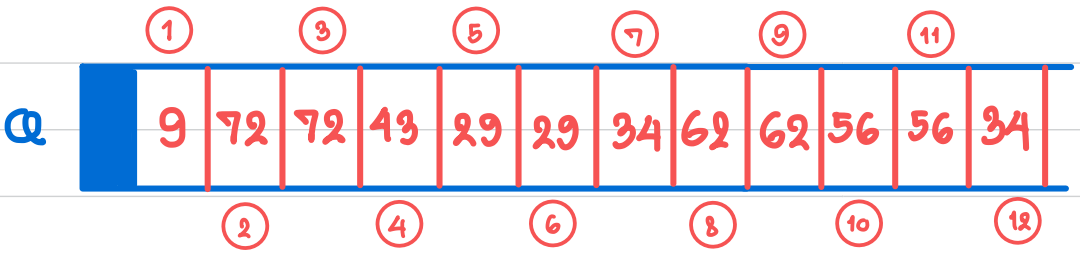
Quiz 2

- ให้วาดรูป queue Q ผลลัพธ์ เมื่อได้รับข้อมูล และมีการทำงานต่อไปนี้
- กำหนด data : 9, 72, 1, 43, 29, 0, 34, 62, 3, 56, 0, 34

```
createQueue(Q)
loop (not end of file)
        read number
        if (number is greater than 5)
            enqueue(Q,number)
        else
            queueRear(Q,x)
            enqueue(Q,x)
        end if
end loop
```

createQueue(Q)

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



③ X = 72
⑥ X = 29
⑨ X = 62
⑪ X = 56

X = 56

```
loop (not end of file)
    read number
    if (number is greater than 5)
        enqueue(Q,number)
    else
        queueRear(Q,x)
        enqueue(Q,x)
    end if
end loop
```

① 1st loop : read 9 → greater than 5 (enqueue)
② 2nd loop : read 72 → greater than 5 (enqueue)
③ 3rd loop : read 1 → queueRear → enqueue
④ 4th loop : read 43 → greater than 5 (enqueue)
⑤ 5th loop : read 29 → greater than 5 (enqueue)
⑥ 6th loop : read 0 → queueRear → enqueue
⑦ 7th loop : read 34 → greater than 5 (enqueue)
⑧ 8th loop : read 62 → greater than 5 (enqueue)
⑨ 9th loop : read 3 → queueRear → enqueue
⑩ 10th loop : read 56 → greater than 5 (enqueue)
⑪ 11th loop : read 0 → queueRear → enqueue
⑫ 12th loop : read 34 → greater than 5 (enqueue)

Quiz 3-4

- จงวาดภาพแสดงการแปลง Infix Expression เป็น Postfix Expression โดยใช้ Stack
  - A * (B + C) – D * F - E

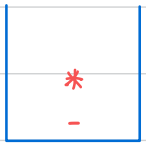
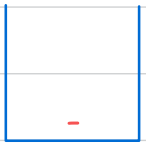
- จงเขียนอัลกอริทึม copyStack(stack1,stack2) ที่ใช้คัดลอกข้อมูลของ stack1 ให้กับ stack2 (ให้มีลำดับข้อมูลเหมือนกัน)

Quiz 3

- จงวาดภาพแสดงการแปลง Infix Expression เป็น Postfix Expression โดยใช้ Stack
  - A * (B + C) – D * F - E

Infix	Stack	Postfix
A*(B+C)-D*F-E		
*(B+C)-D*F-E		A
(B+C)-D*F-E		A
B+C)-D*F-E		A

Infix	Stack	Postfix
+ C) - D * F - E	(*	AB
C) - D * F - E	+ (*	AB
) - D * F - E	+ (*	ABC
- D * F - E	 *	ABC +
D * F - E	 -	ABC + *
* F - E	 -	ABC + * D
F - E	* -	ABC + * D

Infix	Stack	Postfix
- E		ABC+*DF
E		ABC+*DF*-
		ABC+*DF*-E
		ABC+*DF*-E-

Quiz 4

- จงเขียนอัลกอริทึม copyStack(stack1,stack2) ที่ใช้คัดลอกข้อมูลของ stack1 ให้กับ stack2 (ให้มีลำดับข้อมูลเหมือนกัน)

```
Algorithm copyStack(stack1, stack2)
    loop (stack2 is not empty)
        popStack(stack2, x)
    end loop
    createStack(stack3)
    loop (stack1 is not empty)
        popStack(stack1, x)
        pushStack(stack3, x)
    end loop
    loop (stack3 is not empty)
        popStack(stack3, x)
        pushStack(stack1, x)
        pushStack(stack2, x)
    end loop
end copyStack
```

Quiz 5

- ให้เขียนอัลกอริทึม concatQueue (ใช้คำสั่งของ queue ADT ได้)
  - รูปแบบ : concatQueue(Q1,Q2)
  - นำข้อมูลของ Q1 ต่อท้ายด้วย ข้อมูลของ Q2 แล้วนำผลลัพธ์ที่ได้เก็บไว้ใน Q1
  - Q2 ยังคงเหมือนเดิม

```
Algorithm concatQueue(Q1, Q2)

    createQueue (Q3)

    loop (Q2 is not empty)

        dequeue (Q2, x)

        enqueue (Q3, x)

    end loop

    loop (Q3 is not empty)

        dequeue (Q3, x)

        enqueue (Q1, x)

        enqueue (Q2, x)

    end loop

end concatQueue
```