

LinkedList

เขียนโปรแกรมรับค่า **input** เป็นจำนวนเต็ม โดยมีเงื่อนไขว่า **input** ที่รับมาจะต้องมีจำนวนของเลขคู่และเลขคี่อย่างละเท่า ๆ กัน ถ้าจำนวนของเลขคู่และเลขคี่ไม่เท่ากัน ให้เพิ่มส่วนที่ขาดด้วย 0 แทนเลขคู่ และ 1 แทนเลขคี่

****หาก input เป็น -1 จะถือว่าหยุดรับค่า **** ในการเพิ่มจำนวนเต็มลงใน **LinkedList** โดยใช้วิธีการเพิ่มจาก **Node** แรกสุด (**head**) เสมอ

สร้าง **Method void adjust()** สำหรับการจัดเรียงค่าใน **LinkedList** ตามเงื่อนไขที่กำหนดโดย มีเงื่อนไขว่าจำนวนเต็มคี่ และ จำนวนเต็มคู่ ต้องสลับตำแหน่งกันไปเรื่อย ๆ กล่าวคือ หากพบว่าจำนวนเต็มชนิดเดียวกันที่อยู่ติดกัน (คู่-คู่, คี่-คี่) จำนวนเต็มนั้นจะถูกย้ายไปอยู่ที่ตำแหน่งสุดท้ายของ **LinkedList (tail)** และเมื่อสิ้นสุดการทำงานจะต้องแสดงค่าใน **LinkedList** ที่ถูกจัดเรียงใหม่ออกมา

ตัวอย่าง **Input** และ **Output** :

Input	Output
1 2 3 4 2 5 6 4 7 1 -1	LinkedList Before Adjust : 1->7->4->6->5->2->4->3->2->1 LinkedList After Adjust : 1->4->5->2->3->2->1->6->7->4
2 3 6 5 5 9 7 4 -1	LinkedList Before Adjust : 0->0->4->7->9->5->5->6->3->2 LinkedList After Adjust : 0->7->6->3->2->9->0->5->4->5
1 1 1 1 1 1 -1	LinkedList Before Adjust : 0->0->0->0->0->0->1->1->1->1->1->1 LinkedList After Adjust : 0->1->0->1->0->1->0->1->0->1->0->1

ArrayList

เขียนโปรแกรมรับค่า **input** เป็นจำนวนเต็ม และบันทึกค่าจำนวนเต็มที่ได้รับเข้ามาลงใน **ArrayList A** **หาก input เป็น -1 จะถือว่าหยุดรับค่า ** จากนั้นรับ **input** ตัวอักษร(String) จำนวนครั้งเท่ากัน และบันทึกลงใน **ArrayList B**

เขียนคำสั่งให้โปรแกรมแสดงผลชุดข้อความที่สร้างขึ้นมาใหม่และแสดงผลออกมา โดยการอ้างอิงจากค่าของ **ArrayList** ทั้งสองอัน ณ ตำแหน่ง **index** เดียวกัน กำหนดให้ **B** แทนค่าของตัวอักษร และ **A** แทนจำนวนที่ซ้ำของตัวอักษรในข้อความ

ตัวอย่าง Input และ Output :

Input	Output
1 2 3 4 5 -1 A B C D E	ABBCCCDDDDDEEEEEE
1 0 2 1 -1 X O X O	X XXO
3 3 3 -1 . ----...