**การจำลองการทำงานของโครงสร้างข้อมูล Doubly Linked List**

ให้เขียนโปรแกรมเพื่อทำการจำลองการทำงานของ Doubly Linked List โดยใช้ Pointer Based Implementation สำหรับข้อมูลที่แต่ละโหนดใน List เก็บกำหนดให้เป็นเลขจำนวนเต็ม 1 ตัว ส่วนการเลือกทำ Operation จะให้รับข้อมูลเข้าเป็นจำนวนเต็มระหว่าง 1-9 โดยที่

|  |  |
| --- | --- |
| Operation | หน้าที่ |
|  | เรียก operation isEmpty เพื่อสอบถามว่า List มีข้อมูลหรือไม่  การแสดงผล : ถ้าไม่มีให้แสดง yes ถ้ามีให้แสดง no |
|  | เรียก operation getSize เพื่อสอบถามว่า List มีสมาชิกกี่ตัว  การแสดงผล : ให้แสดงค่าเป็นจำนวนสมาชิกใน List |
|  | เรียก operation insertFirst เพื่อเพิ่มโหนดของข้อมูลใหม่เข้าไปใน List ในตำแหน่งแรก  ให้รับ parameter 1 ตัว เป็นข้อมูลที่ต้องการเพิ่ม  การแสดงผล : ไม่มี (ไม่ต้องมีการแสดงผลว่าเพิ่มได้หรือไม่) |
|  | เรียก operation insertLast เพื่อเพิ่มโหนดของข้อมูลใหม่เข้าไปใน List ในตำแหน่งสุดท้าย  ให้รับ parameter 1 ตัว เป็นข้อมูลที่ต้องการเพิ่ม  การแสดงผล : ไม่มี (ไม่ต้องมีการแสดงผลว่าเพิ่มได้หรือไม่) |
|  | เรียก operation removeFirst เป็นการลบโหนดแรกออกจาก List  ไม่รับ parameter  การแสดงผล : ไม่มี (ไม่ต้องมีการแสดงผลว่าลบได้หรือไม่) |
|  | เรียก operation removeLast เป็นการลบโหนดสุดท้ายออกจาก List  ไม่รับ parameter  การแสดงผล : ไม่มี (ไม่ต้องมีการแสดงผลว่าลบได้หรือไม่) |
|  | เรียก operation displayList เพื่อแสดงค่าข้อมูลของใน List ตั้งแต่โหนดแรกจนถึงโหนดสุดท้าย  การแสดงผล : แสดงค่าข้อมูลแต่ละตัวคั่นด้วยช่องว่าง (space) 1 ช่อง |
|  | เรียก operation traverseList เป็นการเดินไปยังโหนดที่ต้องการแล้วแสดงค่าข้อมูลของโหนดนั้น  ให้รับ parameter เป็นข้อมูล 2 ชุด โดยข้อมูลชุดแรกเป็นเลขจำนวนเต็มแทนจำนวนก้าว และ ข้อมูลชุดที่สองเป็นอักขระ L หรือ R จำนวนเท่ากับจำนวนก้าว โดย L เป็นการเดินไปยังโหนดก่อนหน้า ส่วน R เป็นการเดินไปยังโหนดถัดไป  การแสดงผล : ให้แสดงค่าข้อมูลที่โหนดนั้นเก็บ ถ้าไม่มีให้แสดง no |
|  | จบการทำงานของโปรแกรม (แต่ละข้อมูลเข้าจะมี 9 เป็นข้อมูลสุดท้าย) |

**ข้อมูลเข้า**

จะเป็นเลขจำนวนเต็ม 1 ตัวต่อหนึ่งบรรทัด ซึ่งเป็นการเรียกใช้งาน operation ต่างๆ และเรียกจบการทำงานด้วย 9 เป็นข้อมูลสุดท้ายเสมอ

**ข้อมูลออก**

แต่ละบรรทัดเป็นการแสดงผลตามการดำเนินการที่ได้รับ

**ตัวอย่าง**

|  |  |
| --- | --- |
| ข้อมูลเข้า | ข้อมูลเข้า |
| 3  5  3  6  3  2  4  1  7  9 | 3  5  3  6  3  2  4  1  8  3  LRR  9 |
| ข้อมูลออก | ข้อมูลออก |
| 2 6 5 1 | 5 |

**ตัวอย่างแรก**

1. เรียกใช้ insertFirst เพื่อเพิ่มข้อมูล 5 ไว้ที่ตำแหน่งแรก
2. เรียกใช้ insertFirst เพื่อเพิ่มข้อมูล 6 ไว้ที่ตำแหน่งแรก
3. เรียกใช้ insertFirst เพื่อเพิ่มข้อมูล 2 ไว้ที่ตำแหน่งแรก
4. เรียกใช้ insertLast เพื่อเพิ่มข้อมูล 1 ไว้ที่ตำแหน่งสุดท้าย
5. เรียกใช้ displayList เพื่อแสดงค่าแต่ละโหนด

จากนั้นจบการทำงานของโปรแกรมด้วย 9

**ตัวอย่างที่สอง**

1. เรียกใช้ insertFirst เพื่อเพิ่มข้อมูล 5 ไว้ที่ตำแหน่งแรก
2. เรียกใช้ insertFirst เพื่อเพิ่มข้อมูล 6 ไว้ที่ตำแหน่งแรก
3. เรียกใช้ insertFirst เพื่อเพิ่มข้อมูล 2 ไว้ที่ตำแหน่งแรก
4. เรียกใช้ insertLast เพื่อเพิ่มข้อมูล 1 ไว้ที่ตำแหน่งสุดท้าย
5. เรียกใช้ traverseList เพื่อเดินไปยังโหนดต่างๆ ใน List จำนวน 3 ก้าวด้วยรูปแบบการเดิน LRR แล้วแสดงค่าข้อมูลของโหนดที่เดินไปถึง (ในที่นี้ก้าวแรกเดินไปยังโหนดก่อนหน้า (L) ไม่ได้จึงไม่มีผล แล้วเดินไปยังโหนดถัดไปอีก 2 ก้าว ทำให้ไปหยุดอยู่ที่โหนดข้อมูล 5 )

จากนั้นจบการทำงานของโปรแกรมด้วย 9