

Assignment 4.1 String Check

- ✚ **Topics** การสร้างและตรวจสอบรูปแบบคำสั่ง
- ✚ **Learning outcomes** สามารถเขียนโปรแกรมวิเคราะห์ความถูกต้องของสตริงในรูปแบบที่กำหนด
- ✚ **โจทย์ปัญหา** สมมติให้มีคำสั่งที่ใช้งานได้ ให้ตรวจสอบว่าคำสั่งที่ป้อนถูกต้องหรือไม่
 - ลักษณะคำสั่ง (**case insensitive**) แบ่งออกเป็น 3 กลุ่ม
 - 1. คำสั่งที่มีเพียง **token** เดียว ห้ามมิให้มีคำอื่นปนในบรรทัด
list, end, sort, pop, help, sqrt, rec, neg, pow, +, -, *, /
 - 2. คำสั่งที่ต้องตามด้วยตัวเลข 1 ตัว ต้องมีตัวเลข 1 ตัวตามหลังคำสั่ง(ใช้เว้นวรรคคั่น)
delete <n>, search <n>, peek <n>, push <n>
 - 3. คำสั่งที่ต้องตามด้วยตัวเลข 1 ตัว ต้องมีตัวเลขตามหลังอย่างน้อย 1 ตัว
add <n1> [n2 n3 n4 ...], insert <n1> [n2 n3 n4 ...]
- ✚ **ขั้นตอน** เขียนโปรแกรมวนรอบอ่านค่าจนกว่าจะเจอคำสั่ง **end**
 - ถ้าป้อน **syntax** ผิดรูปแบบจาก 1, 2, 3 ให้แจ้ง **Error**
 - ถ้าป้อนรูปแบบ 2 หรือ 3 ให้แยกคำสั่งออกจาก ตัวเลข
 - ถ้าในส่วนของตัวเลขมีการพิมพ์ผิด/พิมพ์เกิน ให้แจ้ง **Error**
- ✚ **ตัวอย่างฟังก์ชัน**
 - อ่านข้อมูล สร้างฟังก์ชันแยกคำใส่ในอาร์เรย์
 - เช็คข้อมูลตัวแรกของอาร์เรย์ สร้างฟังก์ชัน แยกชนิดคำสั่ง
 - เช็คข้อมูลตัวที่สองของอาร์เรย์เป็นต้นไป สร้างฟังก์ชันเช็คว่าเป็นตัวเลขทั้งหมด
 - ถ้าเป็นคำสั่งชุดที่ 1 ให้ดูว่ามีอะไรเหลือค้างหรือไม่
 - ถ้าเป็นคำสั่งชุดที่ 2 ให้ดูว่าข้อมูลเหลือ 1 ตัว และแปลงเป็นตัวเลขได้
 - ถ้าเป็นคำสั่งชุดที่ 3 ให้ดูว่าข้อมูลเหลือมากกว่า 1 ตัวและแปลงเป็นตัวเลขได้

Assignment 4.1 (Test Case)

```
command> [list] [sort] [pop]           // หมายถึงทดสอบ 3 ครั้ง(ไม่ต้องพิมพ์ []) ครั้งแรกพิมพ์ list ครั้งที่สอง sort ....
answer> OK
command> [sqrt ], [neg], [rec], [*] [+], [-], [*], [/] // คำสั่งถูกมีพารามิเตอร์ 1 ตัว
answer> OK
command> [Delete 1], [search 2], [PUSH 3], [peek 4] // คำสั่งถูกมีพารามิเตอร์ 1 ตัว
answer> OK
command> [add 1], [insert 1 2 3] // คำสั่งถูกมีพารามิเตอร์ ได้มากกว่า 1 ตัว
answer> OK
command> [push pop] [end list sort] // ครั้งสั่งผิด พิมพ์มากกว่า 1 คำสั่งในบรรทัดเดียว
answer> parameter error
command> [pip] [delete0]               // พิมพ์คำสั่งผิด
answer> syntax error
command> delete x                      // พารามิเตอร์ตามหลังไม่ใช่ตัวเลข
answer> parameter error
command> delete 1 2 // พารามิเตอร์เกิน
answer> parameter error
command> add                           // พารามิเตอร์ไม่ครบ
answer> parameter error
command> add 123                       // คำสั่งถูก
answer> OK
command> add 1 12 345                  // คำสั่งถูก
answer> OK
command> add 1 12 34X                   // มีพารามิเตอร์ที่ผิดบางตัว
answer> parameter error
command> end                           // คำสั่งถูก จบโปรแกรม
answer> OK
End program
Program written by 62070501xx xxxxxxxx xxxxxxxxxxxxxxxxx
```

Assignment 4.2 (Linked List)

- ✦ สร้าง Linked List ของตัวเลขจำนวนจริง โดยใช้ภาษาซี วนรอบทำงานตามคำสั่งที่กำหนด
- ✦ โปรแกรมทำงานในลักษณะ line command ประกอบด้วย syntax คือ
[คำสั่งเดียว] [คำสั่ง เว้นวรรค ตัวเลข] [คำสั่ง เว้นวรรค ชุดตัวเลข]
- ✦ **add list** เพิ่มชุดตัวเลข เข้าไปใน linked list ตามลำดับ ใส่ต่อจากตัวสุดท้าย
- ✦ **insert list** เพิ่มชุดตัวเลขลงในลิงค์ลิสต์ คำสั่งนี้ใช้ได้เมื่อข้อมูลเรียงลำดับอยู่เท่านั้น
- ✦ **push n** เพิ่มตัวเลข เข้าไปเป็นตัวแรกของ linked list
- ✦ **peek n** เรียกดูข้อมูลในตำแหน่งที่ n ของ linked list (n เริ่มจาก 0,1,2...)
 - ถ้า n เป็น 0 ให้แสดงข้อมูลตัวแรก ถ้าเป็น -1 ให้แสดงข้อมูลตัวสุดท้าย
 - ถ้า n เกินจำนวนที่มีอยู่ ให้แสดงเดือนจำนวนข้อมูลสูงสุดที่มีอยู่
- ✦ **delete n** ค้นหาเพื่อลบตัวเลขที่มีค่า n ออกจาก linked list โดยมีการถามยืนยันก่อนการลบ
 - ถ้าไม่เจอ ให้ฟ้อง not found
 - ถ้ามีข้อมูลซ้ำกันเกินกว่า 1 ตัว ให้วนถามเพื่อลบไปเรื่อยๆ จนหมด
- ✦ **search n** ค้นหาตัวเลขที่มีค่า n ที่อยู่ใน linked list แสดงผลตำแหน่งที่เจอ
 - ถ้าไม่เจอ ให้ฟ้อง not found
- ✦ **[sqrt] [rec] [neg]** ดึงข้อมูลตัวแรกมากระทำ เช่น \sqrt{x} , $1/x$, $-x$ แล้วใส่กลับ **[+] [-] [*] [/]**
- ✦ **[pow]** ดึงข้อมูล 2 ตัวแรกมากระทำเช่น x^2/x^1 , $x^{2 \times 1}$ แล้วใส่กลับ
- ✦ **list** แสดงข้อมูลทั้งหมดใน linked list
- ✦ **sort** เรียงลำดับข้อมูลใน linked list
- ✦ **pop** ดึง(ลบ) ข้อมูลตัวแรก
- ✦ **help** แสดงชุดคำสั่งที่โปรแกรมสามารถทำงานได้
- ✦ **end** สั่งให้จบโปรแกรม

Assignment 4 (Linked List)

- ✚ ถ้าพิมพ์คำสั่งผิด หรือ พารามิเตอร์ผิด/ไม่ครบ ให้แสดง **Command error** แล้วรอรับคำสั่งใหม่
- ✚ ถ้าจำนวนข้อมูลไม่เพียงพอที่จะทำงานตามคำสั่งให้แสดงผล **Can't operation** แต่ถ้าข้อมูลหมดให้แสดงผล **No data**
- ✚ ถ้าคำสั่งทำงานได้สำเร็จ จะแสดงผลหรือไม่ก็ได้
- ✚ การแสดงผลหน้าจอโต้ตอบ จะมีการวนรอบแสดงผล 3 ข้อความ คือ
 - ✚ **list>** ใช้สำหรับแสดงข้อมูลที่มีอยู่ทั้งหมด ให้แสดงทุกครั้งก่อนรับคำสั่ง
 - ✚ **command>** ใช้สำหรับสถานะรอรับคำสั่ง
 - ✚ **answer>** ใช้สำหรับแสดงผลผลลัพธ์ของการทำงานนั้น หรือความผิดพลาดที่เกิดขึ้น

list> NULL command>	//แสดงข้อมูลที่เก็บอยู่ ถ้าไม่มีให้แสดงผล NULL //รอรับคำสั่ง
--	---

list> NULL command> Add xyz answer> Command error	//ถ้าพิมพ์ หรือ ใช้คำสั่งผิด //แสดงผลเมื่อป้อนคำสั่งผิด
---	--

list> NULL command> list answer> No data	//ถ้าป้อนคำสั่งถูก แต่ไม่มีข้อมูล //แสดงผลเมื่อไม่มีข้อมูล
--	---

list> NULL command> Add 1 answer> success	//ถ้าป้อนคำสั่งที่ทำงานได้ //ถ้าทำคำสั่งสำเร็จจะมีบรรทัดนี้หรือไม่ก็ได้
---	--

list> 1 command> + answer> Can't operation	//ถ้าป้อนคำสั่งถูก แต่ข้อมูลไม่พอ (การบวก ต้องมีข้อมูล 2 ตัว) //แสดงผลเมื่อข้อมูลไม่พอ
--	---

list> 1 command> delete 100 answer> 100 Not found	//แสดงข้อมูลเก็บอยู่ //ถ้าป้อนคำสั่งที่ทำงานไม่ได้ (หาข้อมูลที่ลบไม่เจอ) //แสดงผลเมื่อคำสั่งทำงานไม่ได้
---	---

Assignment 4 Test Case

```
1.list> NULL
command> [pip] [delete] [add] // ERROR
answer> Syntax error
list> NULL
command> [pop 10] [add 10.1.1] [add 3x]
answer> Parameter error
list> NULL
command> [delete 1] [neg] [+] // NULL1
answer> no data
list> NULL
2.command> add 30 20 10 20 // add
list> 30 20 10 20
3.command> peek 0 // peek
answer> 30
list> 30 20 10 20
command> peek 1
answer> 20
list> 30 20 10 20
command> peek 4
answer> Maximum peek = 3
list> 30 20 10 20
command> peek -1
answer> 20
list> 30 20 10 20
4.command> delete 30 //delete
answer> 30 found enter y to confirm [y]
list> 20 10 20
command> delete 100
answer> 100 Not found
list> 20 10 20
```

```
command> delete 20
answer> 20 found enter y to confirm [n]
answer> 20 found enter y to confirm [y]
list> 20 10
command> delete 20
20 found enter y to confirm [y]
list> 10
command> delete 10
10 found enter y to confirm [y]
list> NULL
command> delete 10
answer> no data
list> NULL
5.command> push 10 //push
list> 10
command> push 20
list> 20 10
command> push 30
list> 30 20 10
6.command> pop //pop
answer>30
list> 20 10
command> pop
answer>20
list> 10
command> pop
answer>10
list> NULL
command> pop
answer>no data
```

Assignment 4 Test Case

```
list> NULL
7.command> [sort][rec][sqrt] //NULL2
answer> no data
list> NULL
command> add 9.5 50.5 20 -5 -20 2
list> 9.5 50.5 20 -5 -20 2
8.command> + // 50.5 + 9.5
answer> 60
list> 60 20 -5 -20 2
command> - // 20-60
answer> -40
list> -40 -5 -20 2
command> * // -5 * -40
answer> 200
list> 200 -20 2
command> / // -20 / 200
answer> -0.1
list> -0.1 2
command> rec // 1/x
answer> - 10
list> -10 2
command> neg // -x
answer> 10
list> 10 2
command> pow // 2^10
answer> 1024
list> 1024
command> sqrt
answer> 32
```

```
9.command> [+] [-] [*] [/] [pow]
answer> can't operation
list> 32
command> pop
answer> 32
list> NULL
10.command> insert 300 // insert
list> 300
command> insert 100
list> 100 300
command> insert 800 400
list> 100 300 400 800
command> add 200
list> 100 300 400 800 200
command> insert 500
answer> can't insert please sorted before
list> 100 300 400 800 200
11.command> sort // sort
list> 100 200 300 400 800
command> insert 500
list> 100 200 300 400 500 800
12.command> help // แสดงชุดคำสั่ง
answer> list of command
.....
.....
list> 100 200 300 400 500 800
13.command> end // end จบโปรแกรม
```