



Reverse Range

ให้เขียนโปรแกรมเพื่อทำการสร้างและแก้ไข Linked List ที่มี Node Structure ดังนี้

```
typedef struct node {
    int value;
    struct node *next;
} node_t;
```

โดยจะต้องทำ operation ตาม code ที่ได้จาก input ดังนี้

r = reverse at index range

- กลับลำดับ Node ใน Linked List ตาม *index range* ที่กำหนด โดยให้ *node* แรก (ตำแหน่ง *head*) มี *index* เป็น 0 และ *range* จะอยู่ในรูปจำนวนเต็ม $0 \leq start < stop < \text{จำนวน Node}$
- จะต้อง reverse Linked list เริ่มที่ตำแหน่ง *start* จนถึงตำแหน่ง *stop-1* โดยไม่รวม Node ที่ตรงกับตำแหน่ง *stop* เช่นหาก คำสั่งเป็น $r \ 0 \ 2$ ก็จะมีแค่สอง Node ที่สลับตำแหน่งเพียง 2 Node คือ index ที่ 0 และ 1 (ไม่รวม 2)

v = value at index

- แสดงค่า value จาก Node ที่ตำแหน่ง *index* ในรูป **index: value**
- แสดงผล [**<index>: invalid index**] ถ้า *index* ไม่อยู่ในตำแหน่งที่แสดงผลได้

p = print list

- แสดงค่าทั้งหมดใน List
- แสดงบรรทัดว่างถ้าเป็น List ว่าง

Input

- จำนวนแรก คือ m แทนจำนวน Node เริ่มต้นทั้งหมด operation ทั้งหมด
- หลังจากนั้น m บรรทัด จะเป็นค่าของจำนวนเต็มที่จะต้องเก็บลงใน list โดยให้คงลำดับตามทีอ่านข้อมูลนำเข้าไว้
- ถัดมาจะเป็นจำนวนเต็ม n แทน operation ทั้งหมด
- หลังจากนั้น n บรรทัด จะขึ้นต้นด้วยอักษรคำสั่ง i (Insert), v (Value) และ p (Print List) แล้วตามด้วย $index$ และ ค่าที่ต้องการ Insert (เฉพาะในกรณี i)

Input

```
5
1
2
3
4
5
8
r 1 3
p
v 2
r 3 4
p
r 3 5
v 25
p
```

Output

- แสดง output จากการ Reverse (r) Value (v) หรือ Print (p) ตามที่ระบุด้วยชุดคำสั่งใน Input

output

```
1 3 2 4 5
2: 2
1 3 2 4 5
[25: invalid index]
1 3 2 5 4
```