

Lab7 Linked Stack and Queue

4/29 (수)

- 1) 문제: Stack 과 Queue 를 함께 사용하여 "abcde" 를 입력하고, "ebcda"를 출력하는 program 을 Linked List로 구현하시오
- 2) ADT:
 1. PUSH, 2. POP, 3. PRINTStack, 4. ENQUEUE, 5. DEQUEUE, 6. PRINTQueue
 7. Merge, 8. INVERT 9. PRINTList
- 3) 수행순서
 - List1: push 'a' -> push 'e' -> push 'c' -> pop 하여 Stack 내용 출력
 - List2: enqueue 'e' -> enqueue 'b' -> enqueue 'c' -> enqueue 'd' -> dequeue 하여 Queue내용 출력
 - List1 과 List2를 합쳐 오름차순의 리스트 List3 생성 후 출력
 - List3를 역순으로 만들어 List4 생성 후 출력 (Invert함수 사용)

*검사: 다음 순서대로 실행한 후 결과를 검사.

- 1) Display List1 (Linked Stack의 내용을 출력)
- 2) Display List2 (Linked Queue의 내용을 출력)
- 3) Display List3 (List1과 List2가 합쳐진 새로운 리스트 출력)
- 4) Display List4 (List3를 역으로 만든 Invert List인 List4를 출력)

● 출력)

- List1(Stack) : e a
- List2(Queue): b c d
- List3(Merge) : a b c d e
- List4(Invert) : e d c b a

● Invert 함수

- 1) Chain $x = (a_1, \dots, a_n)$ 이 $x = (a_n, \dots, a_1)$ 로 역순으로 변환된다.

```
template <class Type>
void List<Type>::Invert() {
    ListNode<Type> *p = head, *q = 0, *r; //
    while(p) {
        r = q; q = p; // r은 q를 따라간다.
        p = p->link; // p가 다음 노드로 옮겨 간다.
        q->link = r; // q에 이전 노드를 연결한다.
    }
    head = q;
}
```

```

MENU: (1.push, 2.pop, 3.printST, 4.enqueue, 5.dequeue, 6.PrintQ 7.Merge 8. invert 9.PrintList, 10.quit) :1
Enter an integer to push = > a
MENU: (1.push, 2.pop, 3.printST, 4.enqueue, 5.dequeue, 6.PrintQ 7.Merge 8. invert 9.PrintList, 10.quit) :1
Enter an integer to push = > e
MENU: (1.push, 2.pop, 3.printST, 4.enqueue, 5.dequeue, 6.PrintQ 7.Merge 8. invert 9.PrintList, 10.quit) :1
Enter an integer to push = > c
MENU: (1.push, 2.pop, 3.printST, 4.enqueue, 5.dequeue, 6.PrintQ 7.Merge 8. invert 9.PrintList, 10.quit) :2
c ==> is popped!
MENU: (1.push, 2.pop, 3.printST, 4.enqueue, 5.dequeue, 6.PrintQ 7.Merge 8. invert 9.PrintList, 10.quit) :4
Enter an integer to enqueue = > e
MENU: (1.push, 2.pop, 3.printST, 4.enqueue, 5.dequeue, 6.PrintQ 7.Merge 8. invert 9.PrintList, 10.quit) :4
Enter an integer to enqueue = > b
MENU: (1.push, 2.pop, 3.printST, 4.enqueue, 5.dequeue, 6.PrintQ 7.Merge 8. invert 9.PrintList, 10.quit) :4
Enter an integer to enqueue = > c
MENU: (1.push, 2.pop, 3.printST, 4.enqueue, 5.dequeue, 6.PrintQ 7.Merge 8. invert 9.PrintList, 10.quit) :4
Enter an integer to enqueue = > d
MENU: (1.push, 2.pop, 3.printST, 4.enqueue, 5.dequeue, 6.PrintQ 7.Merge 8. invert 9.PrintList, 10.quit) :5
e ==> is dequeued!
MENU: (1.push, 2.pop, 3.printST, 4.enqueue, 5.dequeue, 6.PrintQ 7.Merge 8. invert 9.PrintList, 10.quit) :3
Stack: e a
MENU: (1.push, 2.pop, 3.printST, 4.enqueue, 5.dequeue, 6.PrintQ 7.Merge 8. invert 9.PrintList, 10.quit) :6

Queue: b c d
MENU: (1.push, 2.pop, 3.printST, 4.enqueue, 5.dequeue, 6.PrintQ 7.Merge 8. invert 9.PrintList, 10.quit) :7
List: a b c d e
MENU: (1.push, 2.pop, 3.printST, 4.enqueue, 5.dequeue, 6.PrintQ 7.Merge 8. invert 9.PrintList, 10.quit) :8

List: e d c b a
MENU: (1.push, 2.pop, 3.printST, 4.enqueue, 5.dequeue, 6.PrintQ 7.Merge 8. invert 9.PrintList, 10.quit) :9
List: e d c b a
MENU: (1.push, 2.pop, 3.printST, 4.enqueue, 5.dequeue, 6.PrintQ 7.Merge 8. invert 9.PrintList, 10.quit) :

```