

〈자료구조 및 실습〉 기말고사 문제지 (100점 만점)

2020.06.24. (수) 16:30-18:30

[주의사항]

- 고사 중 **전체화면 녹화&본인을 녹화**합니다. 본인 녹화 시 **본인 확인이 가능하도록 촬영**합니다. 본인 확인이 불가능하도록 촬영하면 부정행위로 간주합니다.
(예. 마스크 쓰기, 얼굴 반만 나오게 하기 등 허용하지 않음)
- 줌을 사용하는 경우 2시간 녹화 시 **45분가량의 영상변환 시간이 필요합니다**. 변환을 마치면 이메일을 통해 대용량 메일로 제출합니다. (고사 종료 이후 한 시간 안에 제출 필수) **제출 이메일 주소: jwon@rcv.sejong.ac.kr / jmpark@rcv.sejong.ac.kr** (두 곳 모두)
- 녹화를 위해 Zoom이 아닌 프로그램을 사용해도 무방하나, 발생하는 모든 책임은 본인이 집니다.
- **핸드폰은 반드시 종료**합니다. 고사 중 랜덤으로 전화 예정이며, 신호음이 갈 경우 부정행위로 간주합니다. 핸드폰 종료의 이유는 핫스팟을 통한 모여서 시험을 보는 행위 및 메신저를 통한 코드 공유를 막기 위함입니다.
- 고사장소가 교내인 경우 혹은 무선공유기를 사용하는 경우 **아이피가 동일하여 OJ에 로그인 못할 수** 있습니다. 이런 경우 핸드폰을 꺼고 교수님께 전화를 걸거나 슬랙을 통해 조교에게 **IP 예외 신청**을 하시기 바랍니다. (전화 010-4742-6757)
- 무선공유기 사용으로 인해 발생하는 IP 예외신청과 이로 인한 시간 소비는 본인이 책임 집니다. 부득이한 경우를 제외하고 집에서 시험을 보시기 바랍니다. 동일 고사장에 존재하는 것으로 간주되는 친구들은 높은 강도의 코드 카피 검사를 실시할 예정입니다.
- 부정행위 발견 시 퇴실 조치하며, 학점은 F로 학칙에 의거 **학사경고** 처리됩니다. 학사 경로로 인해 불이익을 받지 않도록 해주시기 바랍니다.
- 강의안과 본인 실습코드만 활용할 수 있습니다. 인터넷 검색을 허용하지 않으며, 친구 실습 코드 활용도 허용하지 않습니다. 화면이 녹화중인 PC이 외의 전자장비(아이패드, 노트북, 기타 PC) 사용을 금합니다.
- **샘플 제출**을 통해 디버깅이 가능합니다. 참고하셔서 고사에 응시해주시기 바랍니다.
-
- **추가점수**로 획득한 점수로 기말고사의 감점을 보상받을 수 있으며, **추가점수의 합은 기말고사 최대 배점(100점)을 넘을 수 없습니다**.

[문제1][7점] 주어진 두 개의 연결리스트 A와 B의 동일성 여부를 판단하고자 한다. 즉, A와 B를 비교하여 노드 수는 물론 원소들과 그 순서까지 모두 같으면 True, 아니면 False를 반환한다. 연결리스트 A와 B가 각각 elem 및 next 필드로 구성된 노드들의 **헤더 단일연결리스트**로 구현된 경우, 이에 대한 동일성 검사를 수행하는 equal(A, B)를 작성하라.

- ▶ 아래의 의사코드를 참고하여 C 프로그램을 작성하여라.
- ▶헤더 단일 연결리스트가 아닌 경우 100% 감점 (헤더 없음/테일 없음)
- ▶메모리 누수 없으면 1점 추가 점수 획득

<p>Alg add(r, e) input singly linked list with header H, rank r, element e, output none</p> <ol style="list-style-type: none"> if ((r < 1) (r > n)) invalidRankException() p ← H for i ← 1 to r p ← p.next addNodeAfter(p, e) n ← n + 1 {optional} return 	<p>Alg addNodeAfter(p, e) input singly linked list with header H, node p, element e output none</p> <ol style="list-style-type: none"> q ← getNode() q.elem ← e if (p->next ≠ ∅) q.next ← p.next.next p.next ← q return
---	---

<p>Alg equal(A, B) {non-recursive} input singly linked list A, B output True if A ≡ B, False o.w.</p> <ol style="list-style-type: none"> while((A≠∅) & (B≠∅)) if(A.elem ≠ B.elem) return False else A←A.next B←B.next if((A=∅)&(B=∅)) return True else return False 	<p>Alg equal(A, B) {recursive} input singly linked list A, B output True if A ≡ B, False o.w.</p> <ol style="list-style-type: none"> if((A=∅)& (B=∅)) return True if((A=∅) (B=∅)) return False if(A.elem ≠ B.elem) return False else return equal(A.next, B.next)
--	---

입출력 형식:

1) **입력:**

- 첫 줄은 입력받을 연결리스트의 노드 수, 두 번째 줄은 입력할 노드 값
- 입력은 항상 두 개의 연결리스트의 정보를 입력받기 위해 4줄임

2) **출력:** 두 연결리스트가 완전 동일하면 1, 동일하지 않으면 0을 출력한다.

입력 예시 1

5 ⇨ 첫 번째 연결리스트 노드 수
 1 2 3 4 5 ⇨ 첫 번째 연결리스트 각 노드 정보
 5 ⇨ 두 번째 연결리스트 노드 수
 1 2 3 4 5 ⇨ 두 번째 연결리스트 각 노드 정보

출력 예시 1

1

입력 예시 2

5
 1 2 3 4 5
 2
 1 2

출력 예시 2

0

입력 예시 3

3
 1 2 3
 5
 1 2 3 4 5

출력 예시 3

0

입력 예시 4

3
 0 0 0
 3
 0 0 0

출력 예시 4

1

[문제2][7점] 즉각적인 삭제의 대안으로 '게으른 삭제'를 사용할 수 있다. '게으른 삭제'란 삭제해야 할 원소를 일단 표시만 해놓은 후, '삭제' 표시된 원소가 '삭제' 표시되지 않은 원소만큼 많아졌을 때 비로소 '삭제' 표시된 원소들 모두에 대해 실제의 삭제 알고리즘을 수행하는 것이다. 헤더 단일 연결 리스트를 이용하여 다음의 알고리즘을 작성하라.

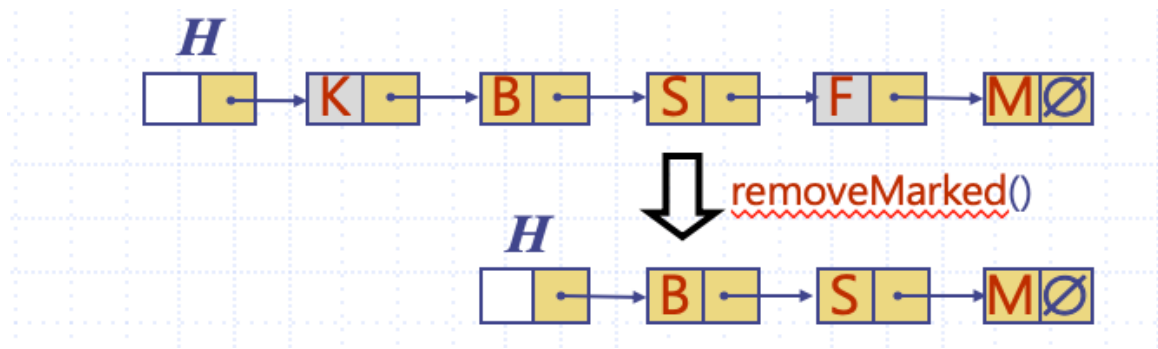
- ▶ 아래의 의사코드를 참고하여 C 프로그램을 작성하여라.
- ▶헤더 단일 연결리스트가 아닌 경우 100% 감점 (헤더 없음 / 테일 없음)
- ▶메모리 누수 없으면 1점 추가 점수 획득

<p>Alg addLast(e) input singly linked list p, element e output none</p> <ol style="list-style-type: none"> 1. q ← getnode() 2. q.elem ← e 3. if(p = ∅) p ← q 4. else pp ← p while(pp) if(pp.next=∅) pp.next ← q break pp ← pp.next 5. return p 	<p>Alg removeMarked() input singly linked list p, element e output none</p> <ol style="list-style-type: none"> 1. p ← H.next 2. pprev ← H 3. while(p ≠ ∅) if(p.deleted) pnext ← p.next pprev.next ← p.next putnode(p) else pprev ← p p ← p.next 4. return
<p>Alg remove(r) input a singly linked list p, rank r output element</p> <ol style="list-style-type: none"> 1. if ((r < 1) (r > n)) invalidRankException() 2. p ← H 3. for i ← 1 to r p ← p.next 4. e ← markNode(p) 5. n ← n - 1 {optional} 6. return e 	<p>Alg markNode(p) input a singly linked list p output element</p> <ol style="list-style-type: none"> 1. e ← p.elem 2. p.deleted ← 1 3. return e

입출력 형식:

- 1) 입력: 'A' 는 노드 입력, 'M'은 삭제노드 마킹, 'D'는 게으른 삭제 수행, 'P'는 노드 출력
- 2) 출력: 'P' 명령어가 입력되면 연결리스트를 순차적으로 출력합니다.

예외사항이 발생하면 InvalidException을 출력합니다.



입력 예시 1

출력 예시 1

10	↪ 연산의 개수	K B S F M	↪ 노드 출력 (1)
A K		B S M	↪ 노드 출력 (2)
A B			
A S			
A F			
A M	↪ 입력 수와 노드 정보		
M 1	↪ 1번 노드 삭제 표시		
P	↪ 노드 출력 (1)		
M 4	↪ 4번 노드 삭제 표시		
D	↪ 게으른 삭제 실행		
P	↪ 노드 출력 (2)		

입력 예시 2

출력 예시 2

9	↪ 연산의 개수	K B S F	↪ 노드 출력 (1)
A K		InvalidException	↪ 4번 노드 삭제 표시
A B		B S F	↪ 노드 출력 (2)
A S			
A F			
M 1	↪ 1번 노드 삭제 표시		
P	↪ 노드 출력 (1)		
M 5	↪ 4번 노드 삭제 표시		
D	↪ 게으른 삭제 실행		
P	↪ 노드 출력 (2)		

[문제3][5점] 두 개의 단일 연결리스트를 잇는 알고리즘 Concat을 작성하라. 주의사항으로는 A와 B는 비어 있을 수도 있다.

- ▶ 아래의 의사코드를 참고하여 C 프로그램을 작성하여라.
- ▶단일 연결리스트가 아닌 경우 100% 감점
- ▶메모리 누수 없으면 1점 추가 점수 획득

<p>Alg addFirst(e) input singly linked list p, element output new node</p> <ol style="list-style-type: none"> 1. q ← getnode() 2. q.elem ← e 3. if(p=∅) p ← q else q.next ← p 4. return q 	<p>Alg addLast(e) input singly linked list p, element e output none</p> <ol style="list-style-type: none"> 1. q ← getnode() 2. q.elem ← e 3. if(p->next ≠ ∅) q.next ← p.next.next 4. p.next ← q 5. return
---	--

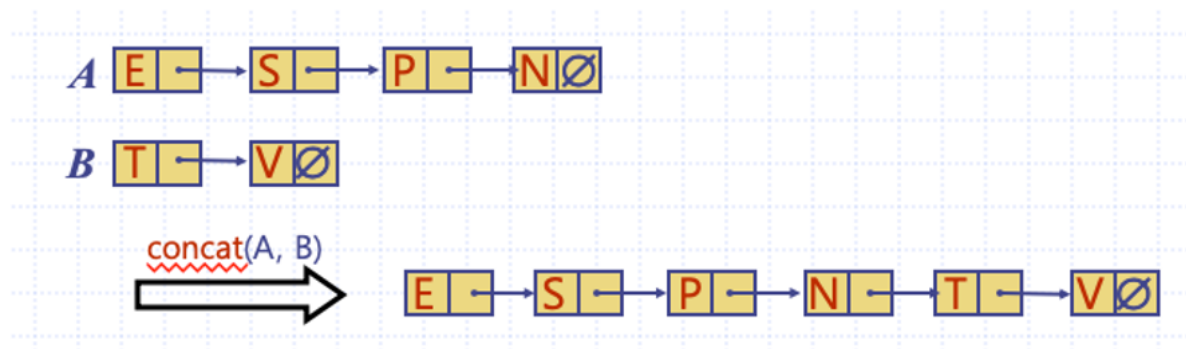
<p>Alg concat1(A, B) {non-recursive} input singly linked list A, B output concatenation of A and B</p> <ol style="list-style-type: none"> 1. if(A=∅) return B 2. if(B=∅) return A 3. a ← A 4. while(a.next≠∅) a ← a.next 5. a.next ← B 6. return A 	<p>Alg concat2(A, B) {recursive} input singly linked list A, B output concatenation of A and B</p> <ol style="list-style-type: none"> 1. if(B=∅) return A 2. return rConcat2(A.next, B) <p>Alg rConcat2(A, B)</p> <ol style="list-style-type: none"> 1. if(A=∅) return B 2. A.next ← rConcat2(A, B) 3. return A
--	--

입출력 형식:

1) 입력:

- 첫 줄은 첫 번째 연결 리스트 노드 수, 두 번째 줄은 첫 번째 연결 리스트 노드의 정보
- 세 번째 줄은 두 번째 연결 리스트 노드 수, 네 번째 줄은 두 번째 연결 리스트 노드의 정보

2) 출력: 두 개의 단일 연결리스트 잇고 순차적으로 출력



입력 예시 1

4	↳ A 노드 수	출력 예시 1
E S P N	↳ A 노드 정보	E S P N T V
2	↳ B 노드 수	
T V	↳ B 노드 정보	

입력 예시 2

6	↳ A 노드 수	출력 예시 2
S E J O N G	↳ A 노드 정보	S E J O N G I M C
3	↳ B 노드 수	
I M C	↳ B 노드 정보	

[문제4][5점] 두 개의 비어 있지 않은 단일 연결리스트 L과 S가 있다. S의 원소들은 오름차순으로 된 양의 정수들이다. printSelect(L,S)는 S에 지정된 순위의 L의 원소들을 인쇄한다.

- ▶ 아래의 의사코드를 참고하여 C 프로그램을 작성하여라. (버전 선택은 자율)
- ▶ 단일 연결리스트가 아닌 경우 100% 감점
- ▶ 메모리 누수 없으면 1점 추가 점수 획득

```
Alg printSelectButSlow(L,S) {version 1.0}
  input singly linked lists L and S
  output none
1. s ← S
2. while(s≠∅)
    snum ← s.elem
    l ← L
    lcount ← 1
    while(lcount ≠snum)
        l ← l.next
        lcount ← lcount + 1
    write(l.elem)
    s ← s.next
3. return
```

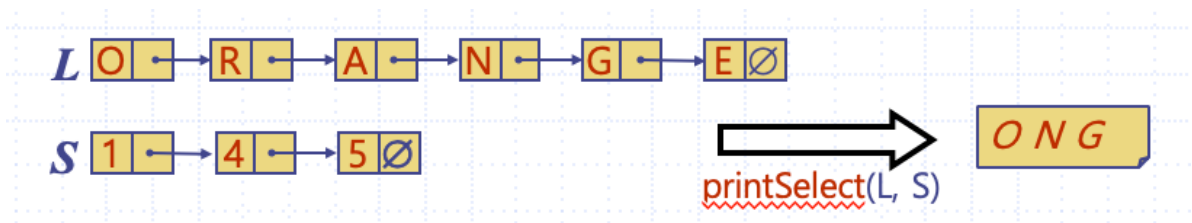
```
Alg printSelect(L,S) {version 2.0}
  input singly linked lists L and S
  output none
1. s ← S
   l ← L
   lcount ← 1
2. while(s≠∅)
    snum ← s.elem
    while(lcount ≠snum)
        l ← l.next
        lcount ← lcount + 1
    write(l.elem)
    s ← s.next
3. return
```

입출력 형식:

1) 입력:

- 첫 줄은 L 리스트 노드 수, 두 번째 줄은 L리스트 노드의 정보
- 세 번째 줄은 S 리스트 노드 수, 네 번째 줄은 출력할 S 리스트 순위

2) 출력: 두 연결리스트가 완전 동일하면 1, 동일하지 않으면 0을 출력한다.



입력 예시 1

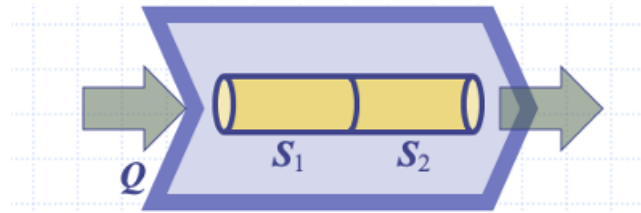
6	↪ L 노드 수
O R A N G E	↪ L 노드 정보
3	↪ S 노드 수
1 4 5	↪ S 노드 정보

출력 예시 1

O N G

[문제5][8점] 두 개의 일반 스택을 사용하여 큐ADT를 만들기 위한 isEmpty, Front, Enqueue, Dequeue 메소드를 작성하라. 주의: 큐 ADT에는 중복 원소들의 저장이 가능하다.

- ▶ 아래의 의사코드를 참고하여 C 프로그램을 작성하여라.
- ▶메모리 누수 없으면 1점 추가 점수 획득



<p>Alg enqueue(e) input stack S_1, S_2, element e output none</p> <ol style="list-style-type: none"> 1. $S_1.push(e)$ 2. return 	<p>Alg dequeue() input stack S_1, S_2 output element</p> <ol style="list-style-type: none"> 1. if($S_2.isEmpty()$) If($S_1.isEmpty()$) emptyQueueException() while(!$S_1.isEmpty()$) $S_2.push(S_1.pop())$ 2. return $S_2.pop()$
<p>Alg isEmpty() input stack S_1, S_2 output none</p> <ol style="list-style-type: none"> 1. return $S_1.isEmpty() \ \& \ S_2.isEmpty()$ 	<p>Alg front() input stack S_1, S_2 output element</p> <ol style="list-style-type: none"> 1. if($S_2.isEmpty()$) If($S_1.isEmpty()$) emptyQueueException() while(!$S_1.isEmpty()$) $S_2.push(S_1.pop())$ 2. return $S_2.top()$

입출력 형식:

1) 입력:

다음 네 가지 연산을 지원해야 함

- A: enqueue(list,e)
- D: dequeue(list)
- P: SSQue 순회

원소는 소문자, 대문자 모두 가능해야 함

입력 예시 1

5 A S A t A a A r P	출력 예시 1 Star
------------------------------------	-----------------

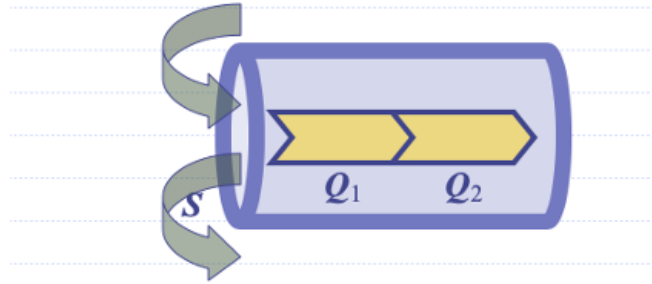
입력 예시 2

7 A D A a A y D P A S P	출력 예시 2 ay ayS
--	----------------------

[문제6][8점] 두 개의 일반 큐를 이용하여 스택 ADT를 구현하라.

▶ 아래의 의사코드를 참고하여 C 프로그램을 작성하여라.

▶메모리 누수 없으면 1점 추가 점수 획득



<p>Alg push(e) input queue Q_1, element e output none</p> <ol style="list-style-type: none"> 1. $S1.push(e)$ 2. return <p>Alg isEmpty()</p> <ol style="list-style-type: none"> 1. return $Q1.isEmpty()$ & $Q2.isEmpty()$ 	<p>Alg pop() input queue Q_1, Q_2 output element</p> <ol style="list-style-type: none"> 1. if($Q1.isEmpty()$) while($!Q1.isEmpty()$) $e \leftarrow Q1.dequeue()$ if($!Q1.isEmpty()$) $Q2.enqueue(e)$ return e 2. if($Q2.isEmpty()$) while($!Q2.isEmpty()$) $e \leftarrow Q2.dequeue()$ if($!Q2.isEmpty()$) $Q1.enqueue(e)$ return e 3. emptyStackException() 	<p>Alg top() input queue Q_1, Q_2 output element</p> <ol style="list-style-type: none"> 1. if($Q1.isEmpty()$) while($!Q1.isEmpty()$) $e \leftarrow Q1.dequeue()$ $Q2.enqueue(e)$ return e 2. if($Q2.isEmpty()$) while($!Q2.isEmpty()$) $e \leftarrow Q2.dequeue()$ $Q1.enqueue(e)$ return e 3. emptyStackException()
--	---	---

입출력 형식:

1) 입력:

다음 네 가지 연산을 지원해야 함

- A: $push(list, e)$

- D: $pop(list)$

- P: 스택이 빌 때까지 $pop()$ 하여 출력

원소는 소문자, 대문자 모두 가능해야 함

입력 예시 1

5 A S A t A a A r P	출력 예시 1 ratS
------------------------------------	-----------------

입력 예시 2

7 A D A a A y D P A S P	출력 예시 2 aD SaD
--	----------------------

[문제7][5점] 괄호 쳐진 수식을 읽어드려 결과 값을 출력하는 프로그램을 작성하려고 한다. 아래의 단계를 따라 순차적으로 알고리즘을 완성하라.

- ▶ 아래의 의사코드를 참고하여 C 프로그램을 작성하여라. 필요에 따라 의사코드 변경 가능하다.
- ▶스택ADT와 트리ADT를 활용하여 문제를 해결하라.
- ▶문제7번은 소문제로 계속 이어짐을 주의하라.
- ▶메모리 누수 없으면 1점 추가 점수 획득

[문제 7-1] 중위 수식을 후위수식으로 변환하라.

- 스택은 연결리스트로 구현함
- 수식은 다음과 같은 우선순위를 갖는 연산자들을 포함함 => 곱셈, 나눗셈, 덧셈, 뺄셈

Alg convert(v) 1. S←empty stack {stores operators} 2. i←0 3. while(!endOfFile()) s←getSymbol() if(isOperand(s)) A[i]←s elseif(s='(') S.push(s) elseif(s=')') while(S.top() ≠ '(') A[i] ← S.pop() S.pop() else {s is an operator} while(!S.isEmpty() & (P[s] ≤ P[S.top()])) A[i] ← S.pop() S.push(s) i ← i+1 4. while(!S.isEmpty()) A[i]←S.pop() i←i+1 5. n←i 6. return	Alg main() 1. convert(A)
---	------------------------------------

입력 예시 1

출력 예시 1

1x2+3+(4+5)x6

12x3+45+6x+

입력 예시 2

출력 예시 2

3+5x7

357x+

입력 예시 3

출력 예시 3

6/3-1+2x4-5x7

63/1-24x+57x-

입력 예시 4

출력 예시 4

1+(2x3+4)x5

123x4+5x+

[문제 7-2][20점] 후위 수식을 이진트리를 이용하여 수식 트리를 만들고 수식 인쇄하라.

- ▶ 아래의 의사코드를 참고하여 C 프로그램을 작성하여라. 필요에 따라 의사코드 변경 가능하다.
- ▶스택ADT와 트리ADT를 활용하여 문제를 해결하라
- ▶문제7번은 소문제로 계속 이어짐을 주의하라.
- ▶메모리 누수 없으면 1점 추가 점수 획득

- 수식 트리는 이진 트리로 구현하라. 수식트리에 사용되는 스택은 트리 노드를 관리하는 스택이다. 스택 재설계가 필요하다.
- 수식 인쇄는 중위 순회로 해결하라.

Alg buildTree(A) input array A of n symbols representing a postfix expression output root of an expression tree 1. S←empty stack {stores operand nodes} 2. n←strlen(A) 3. for i←0 to n-1 s ← A[i] node ← makeNode(s) if(!isOperand(s)) {s is an operator} a ← S.pop() b ← S.pop() makeTriNode(node,b,a) S.push(node) 3. return S.pop()	Alg makeTriNode(op, x, y) input operator node op, operand nodes x,y output op node with left child x and right child y 1. op.left ← x 2. op.right ← y 3. return Alg makeNode(e) input element e output a node with element e 1. node ← getnode() 2. node.elem ← e 3. node.left, node.right ← ∅ 4. return node
--	---

Alg printExpr(v) 1. if(isInternal(v)) write('(') printExpr(leftChild(v)) 2. write(element(v)) 3. if(isInternal(v)) printExpr(rightChild(v)) write(')')	Alg main() 1. convert(A) 2. v←build(A) 3. printExpr(v)
--	--

입력 예시 1

1x2+3+(4+5)x6

출력 예시 1

((((1x2)+3)+((4+5)x6)))

입력 예시 2

3+5x7

출력 예시 2

(3+(5x7))

입력 예시 3

6/3-1+2x4-5x7

출력 예시 3

(((((6/3)-1)+(2x4))-(5x7)))

입력 예시 4

1+(2x3+4)x5

출력 예시 4

(1+(((2x3)+4)x5))

[문제 7-3][5점] 수식 트리의 값을 재귀적 방법으로 계산(평가)하라.

- ▶ 아래의 의사코드를 참고하여 C 프로그램을 작성하여라. 필요에 따라 의사코드 변경 가능하다.
- ▶스택ADT와 트리ADT를 활용하여 문제를 해결하라.
- ▶문제7번은 소문제로 계속 이어짐을 주의하라.
- ▶메모리 누수 없으면 1점 추가 점수 획득

- 수식 트리는 이진 트리로 구현하라.
- 수식 인쇄는 중위 순회로 해결하라.

=> 노드를 방문시 인쇄, 왼쪽 부트리를 순회하기 전에 '('를 인쇄, 오른쪽 부트리를 순회하고 나서 ')'를 인쇄

<p>Alg evalExpr(v) input 수식트리의 root 노드</p> <pre> if(isExternal(v) return element(v) else x ← evalExpr(leftChild(v)) y ← evalExpr(rightChild(v)) ◇ ← element(v) return x◇y </pre>	<p>Alg main()</p> <pre> 1. convert(A) 2. v←build(A) 3. evalExpr(v) </pre>
--	---

입력 예시 1

출력 예시 1

1x2+3+(4+5)x6

59

입력 예시 2

출력 예시 2

3+5x7

38

입력 예시 3

출력 예시 3

6/3-1+2x4-5x7

-26

입력 예시 4

출력 예시 4

1+(2x3+4)x5

51

[문제8][15점] 빈칸에 들어갈 용어를 채우시오.

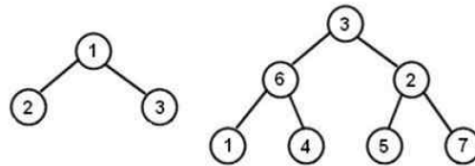
- ▶ 소문제 하나당 1점. 오답을 작성 시 -1점 감점
- ▶ 정확한 정답을 알지 못할 때는 답안을 작성하지 말 것
- ▶ 0J정답을 기준으로 재채점하여 찍어쓰기, 혹은 유사 정답은 정답 처리 함

- ▶ 스택 ADT는 임의의 개체를 저장한다. 삽입과 삭제는 (8-①) 순서를 따르며 스택의 top이라 불리는 위치에서 이루어진다.
- ▶ 큐 ADT는 대기열을 추상화한 데이터구조로서 임의의 객체들을 저장한다. 큐에 대한 삽입과 삭제는 (8-②) 순서를 따른다. 삽입은 큐의 뒤, 삭제는 큐의 앞이라고 불리는 위치에서 이루어진다.
- ▶ 크기 N의 배열을 (8-③)으로 사용하여 큐를 효율적으로 구현할 수 있다. (8-③)배열을 사용한 큐에서는 두 개의 변수를 사용하여 (8-④)와 (8-⑤)위치를 기억한다.
- ▶ (8-⑥)는 스택이나 큐와 마찬가지로 임의의 개체들을 저장한다. (8-⑥)는 앞과 뒤 양 위치에서 삽입과 삭제가 가능하다. (8-⑥)는 스택과 큐가 합친 방식으로 작동하며 배열 또는 연결리스트로 구현할 수 있다.
- ▶ 트리에서의 (8-⑦)란 트리의 노드들을 체계적인 방식으로 방문하는 것을 말한다. 가장 많이 사용되는 순회 방식으로는 (8-⑧)와 (8-⑨)의 두 가지 방식이 있다. 이외 레벨순회는 트리의 각 레벨을 위에서 아래 방향으로 순회한다.
- ▶ (8-⑩)는 순서트리를 모델링한다. 트리의 각 내부노드는 두 개의 자식을 가지며 각각 왼쪽 및 오른쪽 자식이라 부른다. 모든 내부노드가 왼쪽, 오른쪽 자식을 모두 가지는 이진트리를 (8-⑪)라고 부른다.
- ▶ 이진트리에만 가능한 순회 방식으로 (8-⑫)가 있다. (8-⑫)에서는 노드가 그의 왼쪽 부트리보다 나중에는, 오른쪽 부트리 보다는 앞서 방문된다.
- ▶ (8-⑬)는 이진트리에 대한 일반순회다. 즉 선위, 중위, 후위를 모두 포함하는 순회방식이다.
- ▶ (8-⑭)는 유일한 개체들을 담는 용기를 모델링한다. (8-⑭) 관련 작업들의 효율적인 구현을 위해 집합을 집합원소들의 정렬된 리스트로 표현할 수 있다.
- ▶ (8-⑮)란 트리 맨위의 부모가 없는 노드를 말한다.

[문제9][10점 추가점수]

▶ 메모리 누수 없으면 1점 추가 점수 획득

상근이는 슬로베니아의 도시 Donji Andrijevci를 여행하고 있다. 이 도시의 도로는 깊이가 K인 완전 이진 트리를 이루고 있다. 깊이가 K인 완전 이진 트리는 총 $2K-1$ 개의 노드로 이루어져 있다. (아래 그림) 각 노드에는 그 곳에 위치한 빌딩의 번호가 붙여져 있다. 또, 가장 마지막 레벨을 제외 한 모든 집은 왼쪽 자식과 오른쪽 자식을 갖는다.



깊이가 2와 3인 완전 이진 트리

상근이는 도시에 있는 모든 빌딩에 들어갔고, 들어간 순서대로 번호를 종이에 적어 놓았다. 한국으로 돌아온 상근이는 도시가 어떻게 생겼는지 그림을 그려보려고 하였으나, 정확하게 기억이 나지 않아 실패 했다. 하지만, 어떤 순서로 도시를 방문했는지 기억해냈다. 가장 처음에 상근이는 트리의 루트에 있는 빌딩 앞에 서있다. 현재 빌딩의 왼쪽 자식에 있는 빌딩에 아직 들어가지 않았다면, 왼쪽 자식으로 이동한다. 현재 있는 노드가 왼쪽 자식을 가지고 있지 않거나 왼쪽 자식에 있는 빌딩을 이미 들어갔다면, 현재 노드에 있는 빌딩을 들어가고 종이에 번호를 적는다. 현재 빌딩을 이미 들어갔다면 온 상태이고, 오른쪽 자식을 가지고 있는 경우에는 오른쪽 자식으로 이동한다. 현재 빌딩과 왼쪽, 오른쪽 자식에 있는 빌딩을 모두 방문했다면, 부모 노드로 이동한다. 왼쪽 그림에 나와 있는 마을이라면, 상근이는 2-1-3 순서대로 빌딩을 들어갔을 것이고, 오른쪽 그림의 경우에는 1-6-4-3-5-2-7 순서로 들어갔을 것이다.

상근이가 종이에 적은 순서가 모두 주어졌을 때, 각 레벨에 있는 빌딩의 번호를 구하는 프로그램을 작성하시오.

입력 상세:

- 첫째 줄에 K ($1 \leq K \leq 10$)가 주어진다.
- 둘째 줄에는 상근이가 방문한 빌딩의 번호가 들어간 순서대로 주어진다. 모든 빌딩의 번호는 중복되지 않으며, 구간 $[1, 2K)$ 에 포함된다.

출력 상세:

- 총 K개의 줄에 걸쳐서 정답을 출력한다. i번째 줄에는 레벨이 i인 빌딩의 번호를 출력한다. 출력은 왼쪽에서부터 오른쪽 순서대로 출력한다.

입력 예시 1

3
1 6 4 3 5 2 7

출력 예시 1

3
6 2
1 4 5 7