6단원(Stack)

1. 배열 기반 스택 구현
   1. 요약

배열을 바탕으로 구현하는 스택은 topIdx만 잘 조절하면 된다. 초기 topIdx는 -1

1. 연결 리스트 기반 스택
   1. 특징
      1. 처음 head 포인터는 NULL
      2. Last In First Out
   2. 멤버 변수
      1. head : top,가장 마지막으로 입력된 data의 노드를 가리킴(LIFO )
   3. ADT 기능
      1. StackInit(Stack \*pstack) : 스택 초기화
      2. SIsEmpty(Stack \*pstack) : 스택이 비었는지(비었으면 return TRUE)
      3. SPush(Stack \*pstack, Data data) : top에 데이터 추가
      4. SPop(Stack \*pstack) : top에 있는 데이터 삭제
      5. SPeek(Stack \*pstack): top에 있는 데이터 참조
2. 계산기 프로그램
   1. 알고리즘
      1. Infix 형태의 수식을 Postfix 형태의 수식으로 변형한다.
      2. Postfix 형태의 수식을 계산한다.
      3. 위 두 과정 모두 stack을 필요로 한다.
   2. Infix -> Postfix
      1. 왼쪽부터 시작
      2. tok가 숫자이면 새로운 문자열에 추가
      3. tok가 operator면 stack 확인
         1. 우선순위가 tok이 높으면 stack 위에 tok을 쌓는다
         2. 우선순위가 tok이 낮으면 stack에 있는 연산자를 Pop에서 문자열에 추가
         3. 우선순위(‘\*’ , ’/’ > ‘+’ , ’-‘ > ‘(‘ )
   3. Postfix 계산
      1. 숫자이면 Stack에 push
      2. operator이면 stack에 있는 숫자 두개를 Pop한 후 결과를 Stack에 Push

**나중에 코드 다시 보기 꼭!!**