Infix to Postfix Code Report – 2017310528 신재광

1. Stack관련 함수
   1. InitStack – top = -1로 스택을 시작시킴
   2. Push – 스택이 차 있지 않는 한(IsFull로 체크) 스택에 푸쉬함
   3. Pop – 스택이 비어 있지 않는 한(IsEmpty로 체크) 스택에서 팝함
   4. Peek – 스택이 비어 있지 않는 한 top위치의 요소를 리턴함
   5. IsFull – 스택이 꽉 차 있는지 확인
   6. IsEmpty – 스택이 비어 있는지 확인
2. 연산자 우선순위 비교 & 숫자 체크
   1. IsDigit – 인풋이 연산자이면 false, 그렇지 않을 때 숫자로 취급하고 true 리턴
   2. GetPriority – 연산자가 \*, /일때는 2, +.-일땐 1, 나머진 0 리턴
   3. ComPriority – 앞 연산자는 Hex\_num형식으로 받고, 연산자를 뽑아낸 후에 GetPriority 로 연산자의 우선순위를 받고, 앞 연산자가 우선순위가 더 크면 true를 리턴함
3. ConvInfixToPostfix 함수
   1. 스택 선언, InitStack실행해서 스택 시작
   2. Hex\_num peeking, pushing은 16진수를 스택에 푸쉬하거나 픽할 때 사용된다.
   3. pos : convExp의 자리를 나타내는 변수
   4. i=0 에서 len(식의 길이)까지 infix식의 반복문 진행
   5. 식의 요소가 숫자면(IsDigit함수 이용) 바로 Postfix배열에 추가
   6. 숫자가 아니고(operator이고), 스택이 비어있는 상황이면 스택에 푸쉬
   7. 숫자가 아니고, 스택이 비어있지 않으면, top에 있는 operator와 우선순위를 비교(ComPriority 함수이용), 스택의 operator가 더 높은 우선순위라면 그 연산자를 꺼내고, infix식의 연산자는 스택에 푸쉬.
4. 연산 관련 함수들
   1. Add, subtract, multiply, division
      1. Hex\_num의 Large는 \*16, small은 \*1을 해줘야 하는데, 캐릭터형으로 저장이 돼있으므로, 조건문과 아스키코드를 사용해 연산에 사용될 두 수를 십진수 int자료형으로 변환한다.
      2. 각각 주어진 연산(+,-,\*,/)를 진행한다.
      3. 다시 아스키코드를 이용해 16진수 character자료형으로 변환한다.
   2. EvalPostfix
      1. 스택 선언, initStack진행
      2. 식에는 16진수 숫자가 두 캐릭터 자료형으로 나눠서 저장되있는 상황이기 때문에, 숫자인 요소를 처음 발견하면, 그 요소와 그 다음요소를 바로 Hex\_num 변수에 넣어주고, 스택에 푸쉬한다. 그 다음 요소에선 당연히 이 과정을 진행하지 않도록 flag변수를 조절한다.
      3. 연산자 요소가 나온 경우에는 스택의 맨 위 두 숫자를 꺼내서 연산을 진행하고, 연산 결과를 다시 스택에 푸쉬한다.
      4. 연산이 모두 끝나고 스택 맨 위에 있는 수를 리턴한다.
5. 예시 진행 : 08\*0C-A5/A3
   1. Stack 선언
   2. 0,8 convExp 배열에 저장
   3. \*스택에 푸쉬
   4. 0,C convExp 배열에 저장
   5. -를 stack맨 위에 있는 \*와 우선순위 비교, \*이 더 크므로 \*을 convExp 배열에 저장, -는 스택에 푸쉬
   6. A,5 convExp 배열에 저장
   7. /를 스택 맨 위에 있는 -와 우선순위 비교, /이 더 크므로 /을 convExp 배열에 저장
   8. A,3 convExp 배열에 저장
   9. 스택에 -가 남았으므로 -를 convExp 배열에 저장
   10. 변형 결과 : 080C\*A5A3/-
   11. 연산에 사용될 스택 선언
   12. 08 Hex\_num으로 스택에 푸쉬
   13. 0C Hex\_num으로 스택에 푸쉬
   14. 맨 위 두 요소 (08, 0C)를 꺼내 \*진행, 결과값(60)을 푸쉬
   15. A5, A3 스택에 푸쉬
   16. 맨 위 두 요소(A5,A3)를 꺼내 /진행, 결과값(01)을 푸쉬
   17. 맨 위 두 요소(60, 01)을 꺼내 -진행, 결과값(5F)를 푸쉬
   18. 맨 위 남은 요소 5F를 리턴