3. Stacks

msdb@korea.ac.kr

Agenda

- Instruction
- Stack ADT
- Brace check program
- Calculator

LAB 1 > 1_StackADT

Stack ADT

- Implement stack ADT.
  - **CreateStack**: allocate memory and initialize.
  - **Push**
  - **Pop**
  - **Stack Top**
  - **DestroyStack** : remove all items and deallocate memory.
  - **CatStack** : concatenate two stacks.
- You may implement additional functions for your convenience.

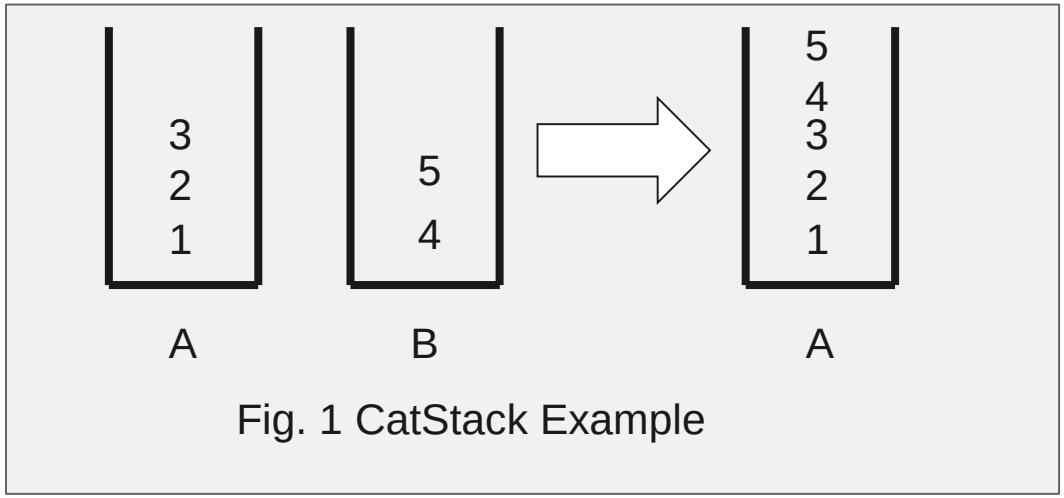

Fig. 1 CatStack Example

LAB 2 > 2_Brace

Brace check program

- Implement a function that checks if a string has correct brace pairs(i.e. (, {, [) .
  - Use your Stack ADT.
  - If the source code has correct pairs, print ‘yes’.
  - If the function find a wrong brace, print ‘no’.
  - You have to check all kinds of brace ‘( )’, ‘{ }’, ‘[ ]’.
- Input : a single line of characters.

Examples:

Input	Output
So when I die (the [first] I will see in (heaven) is a score list). [first in] (first out). Half Moon tonight (At least it is better than no Moon at all]. A rope may form)(a trail in a maze. Help(I[m being held prisoner in a fortune cookie factory)]. ([(([([ ]) ( ) (( ))]))]). .	yes yes no no no yes yes

LAB 3 > 3_Calculator

Calculator

- Implement a function that calculates an infix formula.
  - Use your Stack ADT.
  - Support arithmetic operations (‘+, -, *, /’) and brace (‘( )’).
  - Only non-negative single-digit integers are allowed for input.
  - If a formula can not be calculated, print error message.
    - e.g., divide by zero, invalid formula
- Input: a single line of formula.

Examples:

Input	Output
1+2*3-4/5	6.2
(1+2)*(3-4)/5	-0.6
1++2	INVALID_FORMULA
1/(2-2)	DIVIDED_BY_ZERO