Use of stack

- We have already implemented stack using arrays and link list
- Your first task is to implement stack using array and link list.
- If you have any queries regarding stack implementation email me with screenshot of your code.

Use of Stack

- Example of use: prefix, infix, postfix expressions.
- Consider the expression A+B: we think of applying the *operator* “+” to the *operands* A and B.
- “+” is termed a *binary operator*: it takes two operands.
- Writing the sum as A+B is called the *infix* form of the expression.

Prefix, Infix, Postfix

- Two other ways of writing the expression are

+ A B	*prefix*
A B +	*postfix*

- The prefixes “pre” and “post” refer to the position of the operator with respect to the two operands.

Prefix, Infix, Postfix

- Consider the infix expression

  A + B * C

- We “know” that multiplication is done before addition.
- The expression is interpreted as

  A + (B * C)

- Multiplication has *precedence* over addition.

Prefix, Infix, Postfix

- Conversion to postfix

1. A + (B * C)   infix form

Prefix, Infix, Postfix

- Conversion to postfix

1. A + (B * C) infix form
2. $\underline{A}$ + ($\underline{B\ C\ *}$) convert multiplication

Prefix, Infix, Postfix

- Conversion to postfix

1.	A + (B * C)	infix form
2.	$\underline{A}$ + ($\underline{B\ C\ *}$)	convert multiplication
3.	A (B C *) +	convert addition

Prefix, Infix, Postfix

- Conversion to postfix

1. A + (B * C)	infix form
2. A + (B C *)	convert multiplication
3. A (B C *) +	convert addition
4. A B C * +	postfix form

Prefix, Infix, Postfix

- Conversion to postfix

(A + B) * C   infix form

Prefix, Infix, Postfix

- Conversion to postfix

$(A + B) * C$	infix form
$(\underline{A\ B\ +}) * \underline{C}$	convert addition

Prefix, Infix, Postfix

- Conversion to postfix

(A + B) * C	infix form
$(\underline{A\ B\ +})\ *\ \underline{C}$	convert addition
(A B +) C *	convert multiplication

Prefix, Infix, Postfix

- Conversion to postfix

(A + B) * C	infix form
(A B +) * C	convert addition
(A B +) C *	convert multiplication
A B + C *	postfix form

Precedence of Operators

- The five binary operators are: addition, subtraction, multiplication, division and exponentiation.
- The order of precedence is (highest to lowest)
- Exponentiation ↑
- Multiplication/division *, /
- Addition/subtraction +, -

Precedence of Operators

- For operators of same precedence, the left-to-right rule applies:

  A+B+C means (A+B)+C.

- For exponentiation, the right-to-left rule applies

  A ↑ B ↑ C means A ↑ (B ↑ C)

Infix to Postfix

1

Infix	Postfix
A + B	A B +
12 + 60 – 23	12 60 + 23 –
(A + B)*(C – D)	A B + C D – *
A ↑ B * C – D + E/F	A B ↑ C*D – E F/+

2