❖ INTRODUCTION TO STACK

Applications of STACKS

Stacks can be used to reverse a sequence. For example, if a string ”Computers” is entered by the user the stack can be used to create and display the reverse string “sretupmoC” as follows.

•The program simply pushes all of the characters of the string into the stack. Then it pops and display until the stack is empty

Postfix Expressions Calculator

- Postfix: a b +
- Infix: a + b (what we use in grammar school)
- Prefix: + a b
-

•In high level languages, infix notation cannot be used to evaluate expressions A common technique is to convert a infix notation into postfix notation, then evaluate it.

Priority :

4	^ (power),Unary(-),Unary(+),Not
3	* , / , AND, DIV ,MOD
2	+ , - , OR
1	= , < , > , != , <= , >=

Converting infix expression to postfix expression:

•Initialize an empty stack and a Postfix String S.

•Read the tokens from the infix string one at a time from left to right

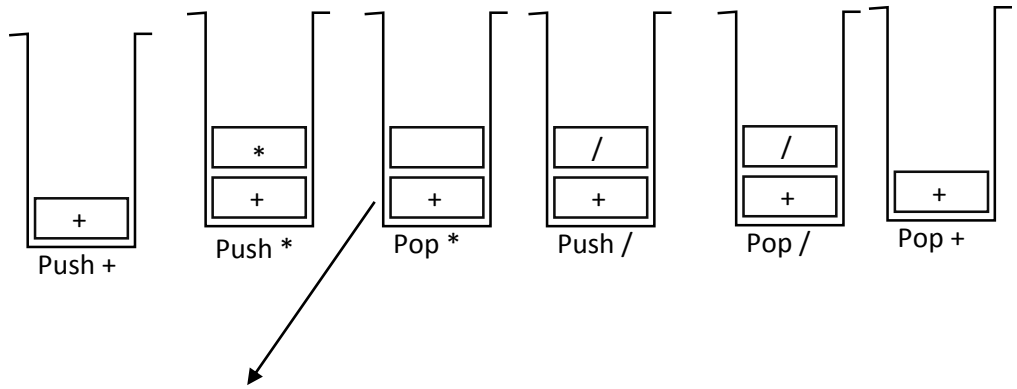
If the token is an **operand**, add it to the Postfix string S.

•If the token is a left **parentheses**, Push it on to the stack.

Algorithm for converting infix expression to postfix expression:

- If the token is **an operator**
- If the stack is empty push the operator in to the stack
- If the stack is not empty compare the precedence of the operator with the element on top of the stack
  - If the operator in the stack has higher precedence than the token Pop the stack and add the poped out element in to the string S.
  - else Push the operator in to the stack
- Repeat this step until the stack is not empty or the operator in the stack has higher precedence than the token
- Repeat this step till all the characters are read.
- After all the characters are read and the stack is not empty then Pop the stack and add the tokens to the string S
- Return the Postfix string S

Example : convert infix into post fix expression 3+4*5/6



Out put string : 3 4 5 *6/+

Example: Convert the following infix expression to postfix using stack:

A + (B /C) #

ch	opstack	Postfix	commentary
	#		Push # to opstack read ch
A	#	A	Add ch to postfix read ch
+	# **#**+	A A	Push + to opstack read ch
(	# + # + (	A A	Push (to opstack read ch
B	# + (# + (	A A B	Add ch to postfix read ch
/	# + (# + (/	A B A B	Push / to opstack read ch
C	# + (/	A B	Add ch to postfix read ch

	# + (/	A B C	
)	# + (/ # +	A B C A B C /	Pop and add to postfix until (is reached. Read ch
#	# +	A B C / A B C / + #	Pop and add to postfix until the opstack is empty.

H.W.

Convert the following infix expression topostfixusing stack

((A–(B + C)) * D) / (E +F) #