OMKAR BHARITKAR

112016020 ECE

Omkar R. Bharitkar
112016020 ECE

DSA LAB 7: Theory

Q1. What is stack overflow and underflow?

→ ① Stack overflow happens when we try to push one more item onto our stack that it can actually hold.
② Stack underflow happens when we try to POP (remove) an item from the stack, when nothing is actually there to remove.

Q2. Differentiate between : Array and stack.

→

	Array	stack
①	In array the element belong to indexes, ie., if you want to get into the forth element you have to write the variable name with its index or location within the square bracket eg arr[4]	stacks are based on the LIFO principle, i.e, the element inserted at the last, is the first element to come out of the list.

Array	Stack.
Insertion and deletion in array can be done at any index in the array	Insertion and deletion in stacks take place only from one end of the list called the top.
Array has a fixed size	Stack has a dynamic and fixed size.
Array caintains elements of same data type.	the stack can caintain elements of the diffrent data types
Diffrent types of Arrays are 1D, 2D, etc.	Stack has only one type.

3. How a stack implemented using a linked list differs from a stack implemented wing an array?

→ Instead of using array, we can also we linked list to implement stack. Linked list allocates the memory dynamically. However, time complexity in both the scenario is same for all the operations i.e. push, pop and peek.
In linked list implementation of stack, the nodes are maintained non-cantiguously in the memory.

Q5. Explain : Infix, Prefix and postfix expression.

→ Infix, prefix and postfix are three different but equivalent notation of writing algebric expression.

a. Infix :

The traditional method of our writing of mathematical expression is called as the infix expression. it is of the form <operand> <operator> <operand>.

As the name suggests, here the operator is fixed inside between the operands. eg A+B here the plus operator is placed inside between the two operators, (A+B)/Q.

b. Postfix :

the Postfix expression has the operator placed right after the two operands. It is of the form <operand> <operand> <operator> eg. PQ-C/, here - operation is done on P and Q and then / is applied on C and the previous result.

Infix expression	Postfix expression
(P+Q) * (M - N)	PQ + MN - *
(P+Q) / (M - N) - (A*B)	PQ + MN - / AB * -

c. Prefix expression.

The prefix expression as the name suggests has the operator placed before the operand is specified. It is of the form <operator> <operand> <operand>. It works entirely in same manner as the postfix expression.

Infix expression	Postfix expression
(P+Q) * (M-N)	* + PQ - MN
(P+Q) / (M-N) - (A*B)	- / + PQ - MN * AB.

Q6. Convert the following infix expression to their equivalent postfix expression:

a) A + B * C / (E - F)

→ ~~AB + CEF - / *~~ ABC * EF - / +

b) (A ^ B * (C + (D * E) - F)) / G.

→ AB ^ CDE * + F - * G /

= AB ^ CDE * + F - * G /

c) (A + (B * (C - (D / E ^ F) * G) * H)

→ A BC * DEF ^ / G * - H * +

Q4. How stacks are used in a non-recursive program?

→ In non recursive function if there is return value then it is pushed into the stack then jump is performed back to calling address.