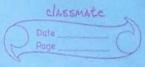
Date 2020 Assignment -10 # Broblem Statement: Implement the program for expression conversion as infort to post fix & it's evaluation using stack based on given conditions: p operands & operators, both must be single character 3) Enput position expression must be in desired format or only '+', -', '\* 4 '/1 operators are experted. Objectives:

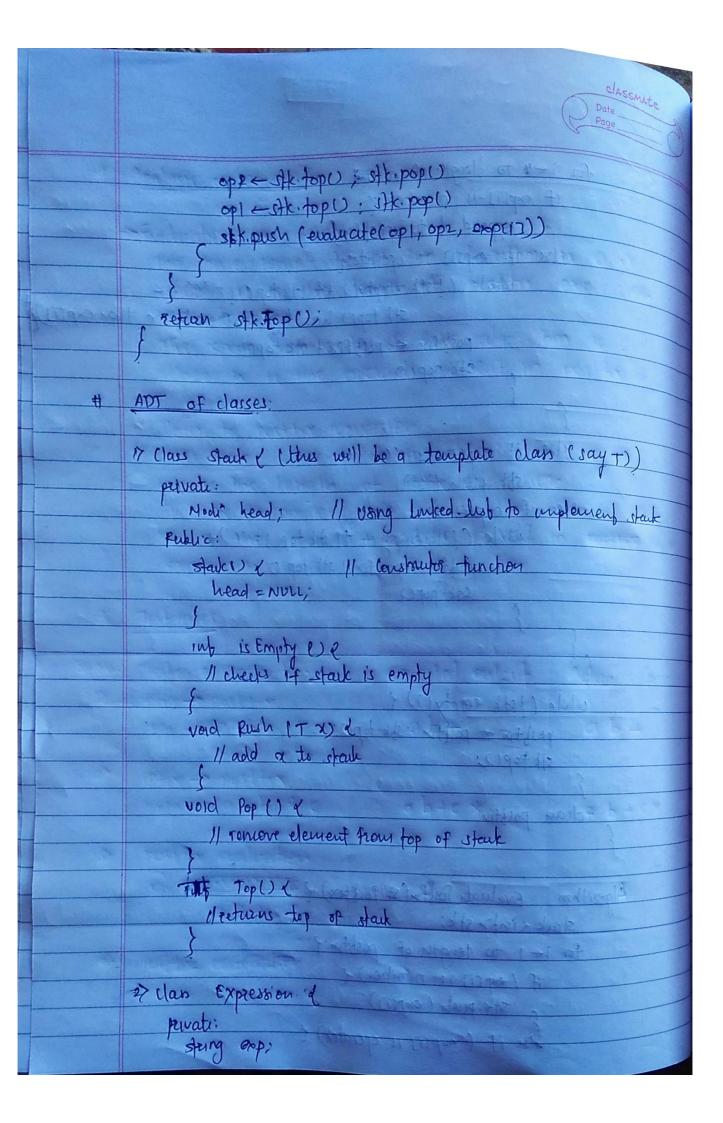
1) To understand types of empression olong with adu &

dus advantages of each types # Objectives, disadvartages of each type 2) Understand usage of stack in evaluating & converting one form of expression to enother. # outcomes: 1> To implement stack using linked list/ arrays.
2> Use stack for calculation of post fix expression which contains simple operators. # Hardware Requirement: manufactures & model: Acer Soiff-3 Processos: Intel core is 8 th gen (82654 @ 1.8 GH2) Infalled Menrozy: 8GB RAM, 512GB SSO. Architecture: 64-bif # Soffware Regurrement: operating System. Oberty 20.04 LTS on Oroule Virtual Marline B processors & 4096MB base memory is allocated) CH version used: CH14 Compiler for (+1: 9+1 (version: 10.1.0)
Lode Editor: Sublime Text (Build 2011)



for i = to longth of infrod of emplis is number of appoind it to postfx. else IF empli] is operatory comile (1stk. empty() &f stk.top() = '(' +4 stk top() has higher frecedence than exptily post-fix + post fix + stk. top() stk. pop() else if expens is 'c'. ste push ('C') else if expen is ')'. while (1strempty () that kitop 1= '(') ? post fix + postax + stk-top 0 Stk-popU uslule (Istk. empty ()) { poston a poston +skx-topos: str. pop(); return postfix Algorithm Evaluate Postfor (\$ Post Fix) 2 stack kinds stk; fa i = 1 to length of postfx: ? of (opped is nymber) <.

stk. push (expers) else it (explit is operator) t



Los is Number (char c) { Hereits if there is number of not hold isoperator (char () & Il cheder if its operator @ not Inf evaluate 1 m op1, int op2, char opets) &
y evaluate op1 & op2 an ording to gets. Int gub Priorty ( char operts)?

11 returns priority of 1/p operator 10) 11 Higherfreiedance (char opr 121, char opr 122) 1 I check the precedence one of opets, over opets Expression (string str = ") { public's string get Expression () 1 gavoid set Expression () 2 9 11 sy value of exp posttix evaluation my evaluade Post fin () { & l'evaluates the postition expression string Infix To Past for (string) & Monvert sep to postile form

## # Analysis of Algorithms:

- The time complexity of the algorithm is O(n)
  where is the length of postfix expression.

  coparate outs?

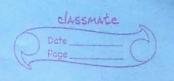
  The space complexity of the algorithm is O(n)
  where is the operands present in postfix expression
  where is the operands present in postfix expression
- The time complexity of the algorithm is O(a) where is the length of infix expression.

  The space complexity of the algorithm is O(a) where the space complexity of the algorithm is O(a) where me is the operands + brankets present in postfix infix expression.

# Test cases:

		A STATE OF THE STA		
SE.NO.	Testicase Description	expected output	Actual output	
D	infix -> postfix	Lead September 19 of	Sign.	
	1/p. 1(a+b)	9 b +	ab+	
	11			
2>	infix - postfix	- Charles - Co	Navet	
	infix -> postfix i/p: (9+b) * c	ab+c*	q b + c*	
37	postfix evaluation	enduration of the contraction of	-we	
	YP: (ANAMANA)	a xilfred oil whouls		
	23+5/			

\* partie exactions



Applications
starts are ideal data structure to calculate fevaluate

prefix @ postfix expressions.

prefix / postfix expressions are preferred in computer

prefix / postfix expressions are preferred in computer

science due to their ease of calculation by

computers.

from infix to possifix efficiently. One can see that we of data standard can significantly reduce the complexity of problem.