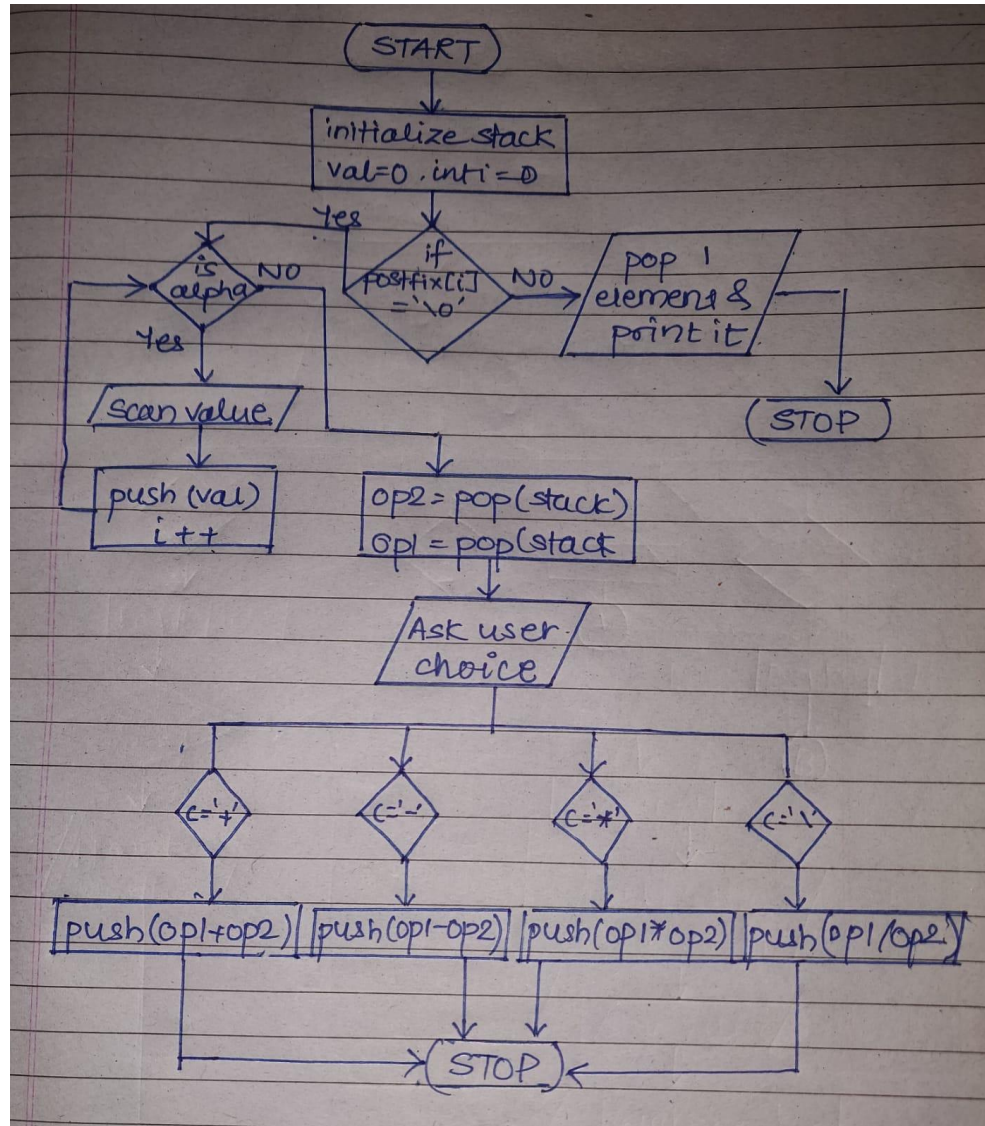
	PUNE INSTITUTE OF COMPUTER TECHNOLOGY	
	PUNE - 411043	
	Department of Electronics & Telecommunication	
	ASSESSMENT YEAR: 2020-2021	CLASS: SE 5
SUBJECT: DATA STRUCTURES		
EXPT No: 9	LAB Ref: SE/2020-21/	Starting date: 11/11/2020
	Roll No: 22119	Submission date: 12/11/2020
Title:	Postfix expression evaluation	
Prerequisites:	• DEVC++ IDE	
	• Knowledge about operations on stack	
Objectives:	1) Learn to use stack as a data structure to handle mathematical operations.	
	2) Implement stack and evaluate an expression to find final solution	
	3) Verify the evaluation process with the help of mathematical procedure	
Theory:		
	<p>A polynomial $p(x)$ is the expression in variable x which is in the form $(ax^n + bx^{n-1} + \dots + jx + k)$, where a, b, c, \dots, k fall in the category of real numbers and 'n' is non negative integer, which is called the degree of polynomial.</p> <p>An essential characteristic of the polynomial is that each term in the polynomial expression consists of two parts:</p> <ul style="list-style-type: none">• one is the coefficient• other is the exponent <p>Polynomial can be represented in the various ways. These are:</p> <ul style="list-style-type: none">• By the use of arrays• By the use of Linked List <p>Representation of polynomial using Array-</p> <p>There may arise some situation where you need to evaluate many polynomial expressions and perform basic arithmetic operations like addition and subtraction with those numbers. For this, you will have to get a way to represent those polynomials. The simple way is to represent a polynomial with degree 'n' and store the coefficient of $n+1$ terms of the polynomial in the array. So every array element will consist of two values:</p> <ul style="list-style-type: none">• Coefficient and• Exponent <p>Representation of polynomial using linked list-</p> <p>A polynomial can be thought of as an ordered list of non-zero terms. Each non zero term is a two-tuple which holds two pieces of information:</p> <ul style="list-style-type: none">• The exponent part• The coefficient parts	

Algorithm	<ol style="list-style-type: none"> 1. Start. 2. Initialize int op1,op2,i=0 and val =0. Declare stack s. 3. Initialize stack s. 4. Check If postfix[i] is not equal to '/0', if so go to step 5 else go to step 13. 5. Check if ith char in postfix array is an alphabet if so go to step 6 else jump to step 7. 6. Ask user the value of char and push it on stack s.i++ and go to step 4. 7. Pop 2 elements from stack and store it in variables op2 and op1. 8. Check ith character in postfix array. If it is '+' got to step 9, if '-', go to step 10, if '*', go to step 11, if '/', go to step 12. 9. Push op1 + op2 into stack. i++. Go to step 4. 10. Push op1 – op2 into stack. i++. Go to step 4. 11. Push op1 * op2 into stack. i++. Go to step 4. 12. Push op1 / op2 into stack. i++. Go to step 4. 13. Pop element from stack and print it. 14. Stop.

Flow-chart



ERROR	No error occurred during performance
REMEDY	No remedy needed
CONCLUSION:	
	1) We learnt to evaluate postfix functions 2) We learnt to implement stack operation and use it as data structure to implement mathematical operations 3) Stack was Implemented and evaluation of expression to find final solution Was done successfully
REFERENCES:	
	1) Seymour Lipschutz, Data Structure with C, Schaum's Outlines, Tata McGrawHill 2) Yedidyah Langsam – Data structures using C and C++ - PHI Publications (2nd Edition). 3) Yashavant Kanetkar, Data Structures Through C, BPB Publication, 2nd Edition

Continuous Assessment			Assessed By
RPP (5)	ARR (5)	Total (10)	Signature:
			Date: