Assignment No.02

(1)

Assignment No.02 (1)
Title: Expression conversion
THE EXPLOSION
Aim I To fine al analysis on consequence 1827 and
Aim: To implement expression conversion using
Stack Data structures
Topological established a la established
problem statement: Implement stack an abstract
data type using singly LL and use this ADT for conversion of infix expression to postfix,
for conversion of infix expression to posttix,
prefix and evalution of postfix and prefix oxp
the state of a state property of the palgranks of the
objective:
To understand the concept and implementation
of stack data structure using SSL.
To understand the concept of evalution of exp
To understand the concept of evaluation of exp
Outcomes:
Implement stack as an ADT.
Implement applications of stack i.e expression
conversion and evaluation.
Theory:
Concept of Linear data Structure:
The data structure where data items are
another is called linear data structure



PICT, PUNE	
	TT 10 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	These kind of data structures are very easy to
	implement because memory of computer also has
	been organized in linear Foshion
	Linear Data structure
	1
	Array stack queue linked list
2)	Example of linear data structure:
	Example: Array
	8 teete
	gueue Linked 18t
	LINFEO IIZI
8)	Stack!
	andered list in which insertion and deletion
	one made at one end couled the top.
	Stack principle: LIFO (last-in first-out)
	To refrieve oth chement needs to remove n-1 etement
	Defination of stack:
	A stack is an ordered list in which all insertion
	and deletions are made out the end couled the
	Hop

PICT, PUNE								<u> </u>)
	If we	have to	make	stack	0F	elem	ents i	10,20,	30
	40,50	,60 th	en 10	ed Ilia	POH	ow u	noste	elemen	F
	and E	11100 08	be to	p most .	elem	ent	in ste	ide	
							n – si Lana e		
(ji)	Termin	ology au	nd dias	preum :					
			T. M. C.		()				
	Top ->	60	push A	4 pu	sh B	pu	sh e	POP	
		50			-				
		40				Es bel	Δ 4		
		30	Y Brindley	700		A GNA	C	6	
		20		TOP	A B	11	<u>B</u>	B	
		10					<u> </u>	<u> </u>	
	S.	Y	Top		1 3				
(D)	ADT C	of Stack	,				1		
<u>U</u>	1 6	2 3 3			- 1 F 501				
	precor	ndition:							
1)	Stack	full()): ch+	eck sta	CK is	s fu	11 000	not. I	F
	Full	then o	de car	1+ Inse	erta	ny (eleme	nt in	
	Stack			7 50	3 S	J			
		in the second							
2)	Stack.	empty (j: che	eck ste	ick is	s fu	llor	- not .=	77
	not the one can insert element.								
	Opener	tions!			.,41	ne -			
<u> </u>	push!	by th	is open	ration	one	car	n pus	neter	1en
	into 1	the ste	ick be	tore pe	extor	mir	10 pu	sh we	
	must	cheek	the st	ack_to	<u>ul ()</u>	<u>)</u>	noutr	<u>ov</u>	
									TORNICATE (IV.)



PICT, PUNE				
	this as a permition	con "	remove the	
2)	2) pop: by this exporation can remove to elements from stack before popping el we should check stack empty() candition			
	elements room states	1 po	condition.	
e de la companya della companya della companya de la companya della companya dell	we should check stack con	p190)	Contony	
	0 100 1			
(A)	Realisation of ADT using:			
	Amay Stack:			
	Declare Stack (Max-Size)			
	// can be float, chan, int			
	Declare top		E 1	
	Array of structure:	8	1 top=4	
	Amon of structure: type def stack struct	D		
Harming of the second of the s	\$	C		
	Laci2-xpM] x	В	ikt v dyz	
	top	A	0	
	3	Parks to		
	Initialize top = -1		Lander Committee	
		termon Karlanti		



			-	4
		20	100	П
		-	-	23
			-39	n
	- 1	1 3	2	10
	63	35	200	2
	27.	200	100	2
-	6.4 :		1	7
32.5	3.1	1	1.72	а
		1	MAC.	×
1	3.5	Silve	200	
- 1	8.30	10 a.	100	- 1
- 2	4.4	5	- 22	
	1.0	X	5464	
	3.3	5.5		9
	70		-250	÷
	78	1	100	•
		14		•
			- 160	•

PICT, PUNE	
2)	pop: by this operations can remove the
	elements from stack before popping element.
	we should check stack empty() condition.
	Application of stack:
	Expression conversion
	Infix to postfix
	Infix to prefix
	prefix to infix
	prefix to infix
	Expression evalution
	passing
	Simulation of recursion
	Game playing find paths
-	exhaustive searching
	coell format paranthesis
	Reversal of string
_	Tree or graph traversal Backtraina
	Backtraing

(3)

PICT, PUNE	
	Expression conversion and stack:
(1	Need for expression Conversion:
	Usually ifix notation is used but the problem
	coithinfix is that is lot of one head in
	computing the result for infix expression
	which result in loss of efficiency.
	In the alternative form there is no need for
	paranthesis which in no need for repeated
	sa scanning of thee expression alternative
	Forms procedure and associativity among
	operator is alredy accounted for
	- I would be a second of the s
2)	Advantages of polish notations:
West L	Expression can be shown without paranthesis
, . 	It is convient to evalute formula an stacks
7	polish notation timinates problems faced by
inge Men	precedence.
	The complete expression can be passed in one
	Travorsel
100016	
and the state of the state of	
	and the control of the first and the same that the same the same of the same that the same of the same
	A TANTA TO THE PROPERTY OF THE





PICT, PUNE	
	Working:
1)	Infix to post-fix conversion:
	During the scan the expression an append
	te immediation added to the output string
	cabile the operation stack stones the operators
	& left paranthesis as soon as they appear
	handles sub expressions and mangoes the order
	of precedence and associtivity of operations.
	riche publication de la facto de la companya de la
2)	Infix to infix conversion:
	For this conversion the infly expression is co
4.23643	sconned from night instead of lest to right
	like a postfix conversion. The apoperand are
	added directly to output string while the
	stack contains the operators and parenthesis
3)	Postfix evalution:
	This is done by adding the operand onto a
	stack and removing two whenever an operator
	Is read in postfix expression string. The results
	of the operation is then added to the stack
	and process is repeated until there is only
	one element remaining in the stack

PICT	2870
Plant .	(4)
PICT, PUNE	
4)	Prefix evaluation
	This is done very similar to postfix evalution except instead of traversing the string left to right. The presence is repeated until there is only one element left in stack.
	Test cases /validations;
	validations:
1	Mo of operator & operand relation-ship
2)	well formed peranthesis
	Test cases:
1)	Based on precedence of operators:
	그들은 보다는 경기에 살아가 되었다. 이번 사람들은 아니는 사람들은 아니는 사람들은 사람들은 사람들은 사람들은 사람들은 사람들은 사람들은 사람들은



	237
PICT, PUNE	(5)
	Declaration of Data structures:
	In this program the doubt structures used are Stack and linked list.
	conclusion:
	Successfully implemented the assignment using stack as stack as ADT to do the infix to postfix, prefix expression learned stack data structures, its implemented using stack tits application in expression conversion as a temporary data structures.