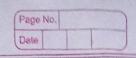
	Page No.
	Name: Liza lad Enrollnment Number: 200303108087
	Name: Liza lad Enrollnment Number: 200303102087 Subject: DSA Code: 203105205
	Suggett Dort tale . 2001
	Asignment - 2
	Assignment - 2 Ch-2 Btacks and Queue.
91	what is Stack? List out different operation of it and write algorithm for any two operations.
	operation of it and write algorithm
	for any two operations.
	J. J.
Soln:	A stack is an Abstract Data Type.
	en stack is an Abstract Data Type
1	It is maned stack as it behaves in
	for example: - a deck of caras
-Few	for example: - a deck of cards or a pill of plates, etc.
->	operations of stacles.
	operations are limited.  push (add item to stack)  pop (sumouse top item from stack)  top (get top item without removing it)
1)	tual (add item to stack)
2)	hop ( gemoise top item from stack)
3)	tool act too item without removing it)
4)	clear
5)	istmpty
6)	isfuel
4)	372?
	Carried Barried Carried And Land Carried Carri
-3	Algorithim for Push Operation
	Step 1: Checks if the stack is full produces
335.1	set Step 2: If the stack is full, produces
	del Step 2: If the stack is full, produces and exit.
	Step 3: 91 the stack is not the
	Step 3: If the stack is not full to increments top to point next empty:
	space.



Step-4: Adds data element to the stack location, where top is pointing step-5: Returns success

> Algorithm for pop operation

Step 1 - Checks if the stack is empty.
Step 2 - If the stack is empty, produces
an error and emit.
Step 3 - If the stack is not empty,
accesses the data element at which
step 4 - Decreases the value of top by 1.
Step 5 - Returns success.

Drite a'C' program or an algorithm to convert infix empression without parenthesis to postfix expression

(1) Push "(" onto Stack, and add")" to the

Scan x from left to right and repeat step 3 to 6 for each element of x until the stack is empty.

(3) If an operand is encountered, add it

(u) 9 a left parenthesis is encountered, prish it conto stack

Page No. Date (5) If it an operator is encountered (1) Reflectedly pop from Stack and add to I lach operator (on the top of Stack) which has the same precedence as on higher precedence than operator Add operator to stack. End of 4) END. · los · Le · ·

Page No. 0-3 (1) White an algorithm to reverse a a string using stack Sel Step 1: String to Char[] step-2: Create a stack Step-3: Rush all characters, one by one They pop all characters one by Step-4: One and put into the charl'I Finally, convert to the string expression without parenthesis to post fix er (2) Write an algorithm to check if an expression has balanced for anthesis using stack. sol step-1: When any open symbol i.e., (: { [ is rencountered, it will be pushed in the stack. Step 2: If any close symbol i.e, ), 3, J is encountered any of the three can happen. step: 3: The Tos(Jop of stack) is checked if the encountered close symbol matches with its open symbol, then open symbol which is at TOS is popped out. The TOS (Jopof Stack) is checked, if the encountered close symbol does not matches with its open symbol, then matching symbol.

The ToS (Jop of Stack) is checked . If the stack is empty, then -1 is returned as there is no open symbol in the be taken in use writing orecursive efunction? give example of any one recursive function. I what is Tower of Hanoi ? Explain it with n= 3. Sol" Recursion is a programming Jechnique in which a function contains a function call to itself A recursive algorithm must & change its state and move toward the the function will terminate. Enample of any one recursive function Direct recursion Eg: = sfactorial(int n) factoral (int x-1); Jower of Hansi 3ts a simple game concept which can be applied in stack. The concept is that the bigget value should not be Kept upon the smaller value.

95 translate the following string into potals notation and trace the content of Stack. A-(B/C+(D % E\*F)/G)\* H Infix Character Scanned Stack Postfix/ Polish ABC C-C+ ABC/ C- (+( ABC/ C-C+( ABC/D C-C+C% ABC/D: C-(+(1/, ABC/DE (- (+ ( 1, \* ABC/DE C-C+("/.\* ABC/DEF ABC DEF\*1/. - (+ ABC DEF\* 1. C-C+1 C-C+1 ABC DEF\* 1. G ABC/DEF\* 1. G1/+ ABC/DEF\*1/. G/+ ABCIDER\* 1. GI/+H ABCIDEF \* 1. COAH+-