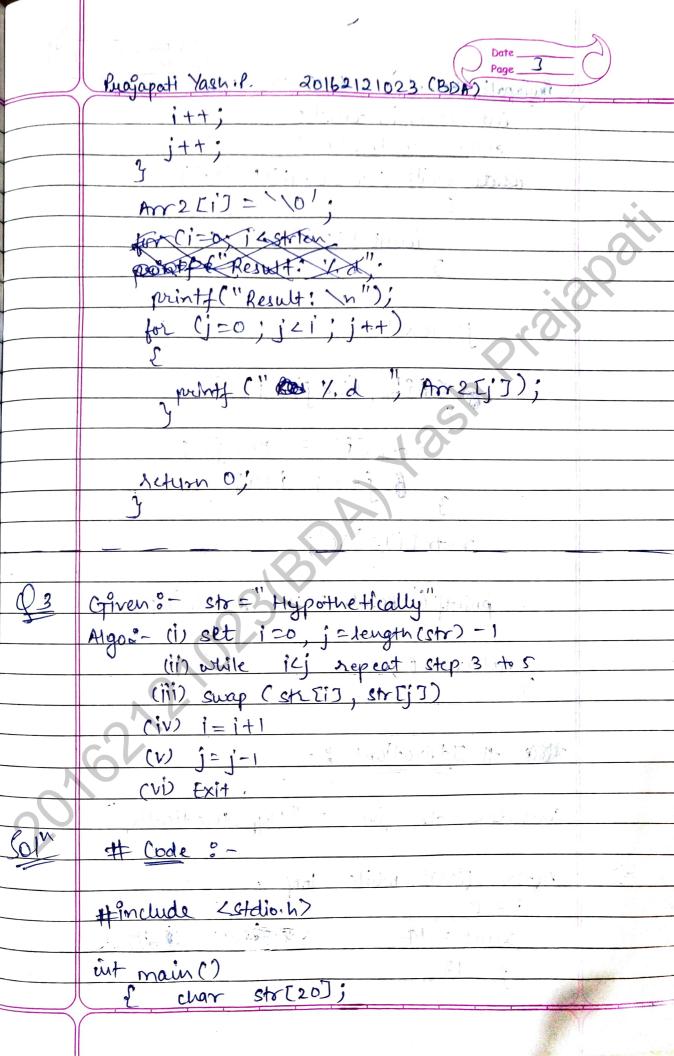
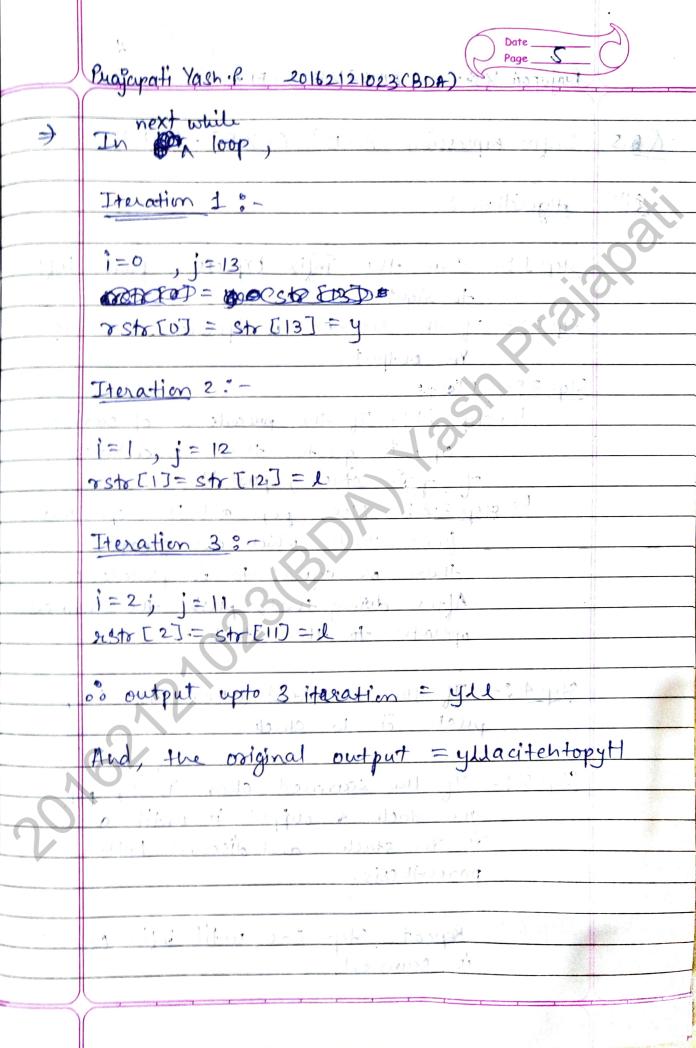


Biogarati Yash. P. 20162121023 (BDA) that Arr1 = []= {11,12,13,143; Pmt Am 2 [] = {21, 22, 23, 24, 25 }; int i,j=0; int rev_Arr1[10]; int count; while (Arra [count] 1= 101) count ++; j= count - 1; COOCONTRA SOCIO Quale (ici) { rev-Arr1[i] = Arr1[j]; rev_Arr1 [i] = \o! 1=0; while (Arr2[i]1=1101) 9=0; while (rev - Arr 1 [j] 1 = 10') Am2[i] = rev_Am1[j];



Brojapati Yash .f. 20162121023 (BDA) int 120, j, count; char rstr[20]; osside while (sto [court] != 1/0') count ++; j= went - 1; , while (12j) rstr[i] = str[i]; 7 ; j--; i++; ~str[i]= 101; printf (Result = 1/15 1); Return 0; # Iterations ? -Consider str given = "Hypothetically" After first while loop, count = 14 9= 13



Date 6

A		20162121023	(BDA)
Projapati	Yach 1	20102121025	CP

Jes Infix expression A+B* (CY.D)-(E/F) Algorithm :-314 Step 1:- Scan the infix Expression from left to right: Sty 2: - If the scanned is operand, show it In output. Stop 33- Else, Ly step 3.1; - If the precedence of samued op. is greater than precedance of op in stack, pust it. 4) step 3.2: - Flee, pop all the operators from stack which are greater than or equal to in precedence After doing that, push the scanned operator to the stack. Step 4: - If the scanned char is () push of to stack 1. the plant of the second of Step 5:- If the scanned char is), pop the stack of output it until a 9s In stack and discard both parenthesis.

Step 6: - Repeat Step 2-6 until infix expression is scanned.

()	Date Page	(
	Brajapati Yash. P. 20162121023 (BDA)	
	Step 7: - luint support	
	Step 8 6 - pop and output elements from	
	stack until it is not empty.	20~
		20
	for egi-	0.
	Given Infix expression :- A+B* ((1.0).	- (E/F)
	Output :- ABCD %. * + & EF/	
	Stack: ABCD 1. * + EF /- / ABCD 7. * + EF /	
	/ ABCD 1, TX + L+ /	
	F 0020% + + FF	
	E ABCD 7. + E + ABCD 7. + +	
÷	+ ABCD 1/, * T	
	* ABCD */, *	
	Y. ABCD Y.	
	D ABCD	
	C ABC	
	B AB	
	A A	
7		
- 17		