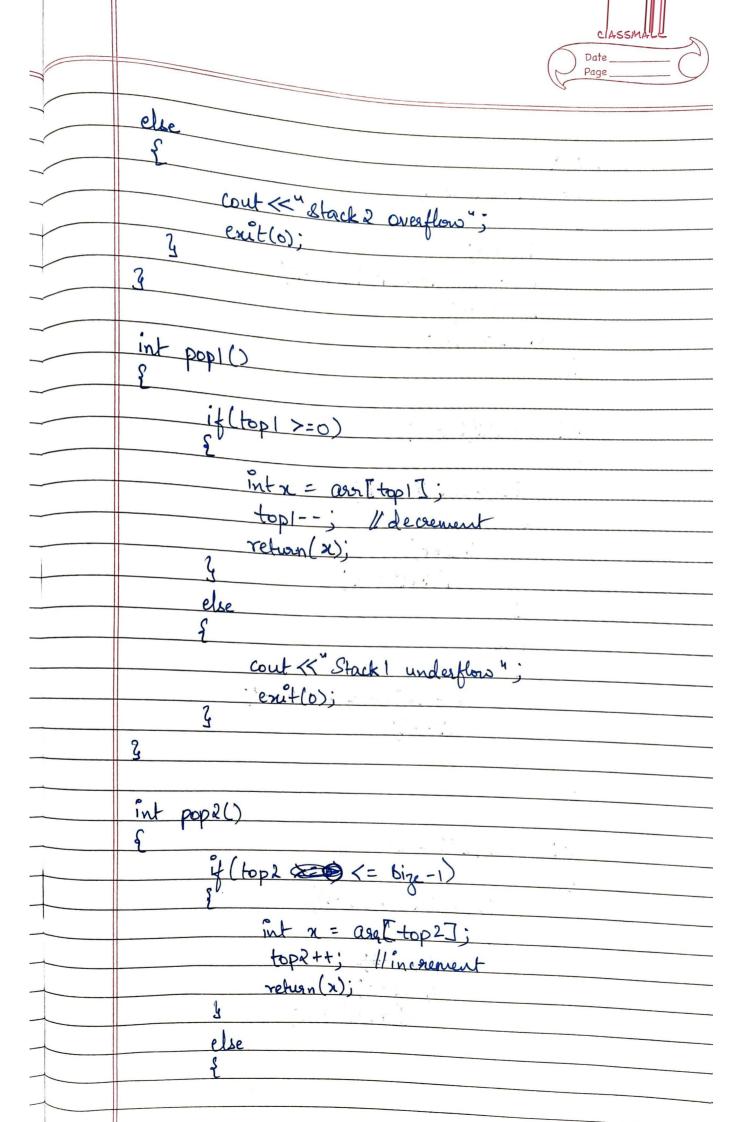
## R. Darshan Nagpal RVCER4MCADII



	2 STACK IN AN ARRAY
	Implement 2 stacks using one array, it can be done by the following steps.
	Step 1: Maintain 2 pointers for both the stack top elements.  top 1: for stack ( starting from 0 index ).  top 2: for stack d (Starting for n-1 index).  1st stack will start from beginning.  Rud stack will start from last.
	Step 2: Since stacks cannot overlap in one single array, voe will have to see whether top 1 < top 2.  Stacks have the concept of overflow and underflow
<b>→</b>	program: (C++)  #include (iostream.h>  #include (stdlib.h)
	class twostacks & int* ass; int size;
	int lop1, top2;

classmate



	1030	
	cout « Stack 2 index flow:	
	exit(o);	
	4	
	3	<b>↓</b>
	£;	
	int main()	
	4	
	twoStacks (ts(5);	
	ts.push(5);	
	ts. push 2 (10);	
	ts. push (15);	
	ts.push 2 (11);	
	(s.push 2 (7);	
	cont < Popped element from Stack 1: " < te. pop ();	
	cout < "Popped element from Stack 1: " < ts. pop 1();	
	return (O);	
	<u>L</u>	
<i>→</i>	Output:	
	Popped element from Stack 1: 15 Popped element from Stack 2 is: 7	
	Popped element from Stack 2 is: 7	
$\rightarrow$		
	Complexities:	_
	= T= · M() / . A / A / A / A	_
	· Time: O(1) - for both bush and pop	_
	· Space: O(n) - where n is array size.	_
		_