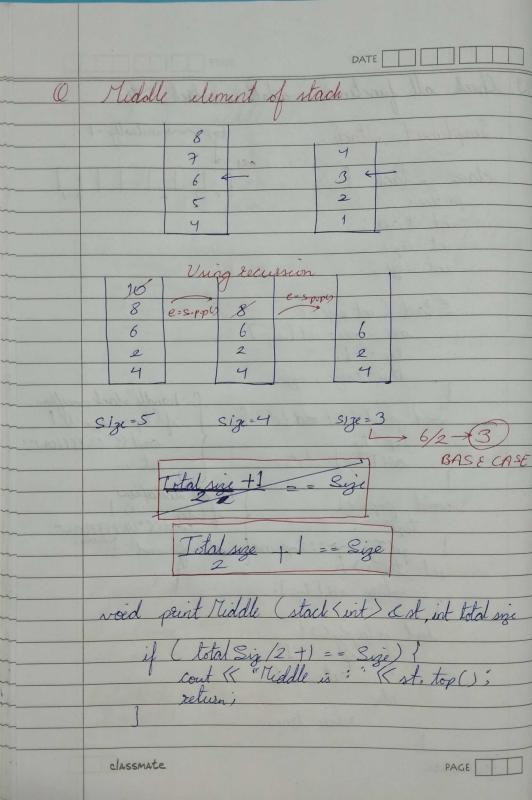
~	DATE
	33 Stack I
	Stock - in a data trustice
	Stack - is a data structure that follows LIFO I dates
	stack of plates
~	in stack out Last in first out
V	1 - 34
~	2 3 3 2 4 3 1
~	3 , 2
~	
~	A NOTICE: Revelsed
~	4, 2
~	3,5,7,9 9 9 9 7 5 3
~	7 ) )
~	5
~	
~	Stack using STL // get top element
~	1/ leation // get top element
~~	stack (int) st;
~~	siack In / M
-	1/ insertion / delete top element st. pop();
~~	// insertion st.pop();  st. push (5);  // get size
~	11 at rice
	st. size )
	Usheck it empty
	11 check if empty st. empty
~~	
	classmate
	PAGE

	DATE
4112	Sheck all functions available in stack STL
	Implement Stack top winitially -1  class Stack 1  public:
	ase
	class Stack & Delice of the stack of the sta
	public:
	int *ale;
	int top;
	int size;
	Stack (int size) 1
	all - new int (sige ).
	top = -1; this → size = size;
	shus size = size,
	handle stack overflow
	vaid push (int statutata) [ if (top == syi-)) { top+t; card << "OFRFIOW",
320	top+t; cad K"OLFRFIOW",
	roid pop() { handle undaflow if (top = ) !  top; cout(("UNDERFLOW";  get Top ~> : top name already used futurn;  int top()!
	top; cout << "UNDERFLOW";
	get Top ~ : top name already used fullism;
	int top Ol
	return are [top];
1 0	
	bod is Emply () {
	f(lof = -1)
	else return false:
	setuen true;
	June,
	3; classmate PAGE PAGE



DATE DATE
1/ step 1
int top Element: st. top ();
st. pop(); print Middle (st. total Sige);
print Middle (st. total size),
11 step 2
Reverse a stack
TEN COSE A MANNE
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$
1 Save @ Remsion 3 Inset at
top & pop illrevase bottom
remaining
Insert at bottom
MUNICAL ACTION
4 3
$\begin{array}{c c} 3 \\ 2 \\ \end{array}$
1 4
element = 7
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$
1 1 1
3
classmake 1 put protection of the production of

		DATE
	vaid insut At Bottom Cstack	(int) &st, int item)
	11 base case	
	if (st. empty) {     st. push (item);     return;	The Kithing
	int top Etement = st. to	o C);
	st. pop ();	
-	insert At Bottom Cst, vter	n);
	st. push (top Eternent);	
	Reverse	
	void reverse (stack (int)	lst) [
	// base can	Nave -
	// base case if (st. empty()) return:	about at letter
	// step ] int top Element = st. t.	():
	st. pop();	
	insut At Bottom	to the time
1	reverse (st);	13+112
9	insert At Bottom Cst, top	Element);
	classmate	PAGE