```
Stacks
                                4.6.1.11
1. Write a program to stimulate the working
of Stack waing an array with the following
a) push 6) pop c) Display.
 #include < statio. h>
 void push(int);
 void pap ();
void display ();
 int stack(s12E), top=-1;
 Void push(int value)
    1 top = top + 1;
      Stack[top] = value;
      Stack (top) : vin
print f ("Insertion Successful");
                             98 mistaf
                            2 El = 2 2 2 13. 3
```

ib(top == -1) (ones: babo): 1 printf ("Stack is Empty"); breaks Cases: desplay (); { printf ("the deleted element is "od"), stack [top]);

top=top-1;
} Chat put: void displaye) Stack = Stack(5) printf ("stack is empty"); Stack push(3) Start, display () for [i=top; i>=o; i--)

{

printf ("/.d", Stack(i));

printf ("/.d", Stack(i));

} 3 pished into the stack int value, choice; while (1) printfl " 1. PUSH, 2. POP, 3. DISPLAY, H. EXIT); Scanf (" 1.2", & choice); Switch (choice) Case t: printf ("enter a value");

Scanf "-1. d", & value); preh (value); (1- = da)) ! break; lanez: popl); private ("Stock is Empty"); break; Case 3: display (); Case 4: (exit(o); ("wrong input); Getput Stack = Stack (5) Stack. push (1) Stack. push(2) Stack. push (3) Stact, display () Stack.popl) I pushed into the stack 2 pushed into the Stack 3 pushed into the stack. Stack elements: int value, charte ? print| " 1. PUSH, 2-POP, 3. DISSPERY, H. EXELESS scanf [" 1.8", & charce); Switch (charce) Cose i: printif (entre a value ")

3

2. Drute a Program to convert a given Valid parenthesized infin aunthmetic Enpression to portfin enpression. The enpression consists of single character operands and the binary operators + (plos), - (minms), to (multiply) and Ildivide) and International to the standard beautiful to the stand It include < Stdio. h > Ctack stack: #include astd bb.in> # define MAX-SIZE 100 10-10-0-100 while (infin (i) ! - ' (o') { typedef stmit { cher items [mAx-SIZE]; int tops; (infinition (i) : : (1) { & Stack; void proh(Stack &s, charc) { if (S-> top== MAX-S12E-1) { return; 5-> items[++(5->top)]=c; Char pop (Stack \$5) { if (5-> top = 2-1) { = 1 got hotel fi print { ["Stack underflow"); I return s-> items [(s-> top)--]; int precedence (char operator) ? if Coperator == +, 11 operator == '-') { return;

Jehre if Coperator = 'a' 1) operator = 2/1){ return 2) shader overable and return 0; on & Common - (2019) +2+ void infin-to-postfin (char * infin, char & postfin Stack stack; Kindud 28tdno. h> whole establishins Stack top=-1; ne MAX-2[26 100 Int 1:00, 1:00; while (infin [i] != 'lo') { if [is alnews (infin [i])) { postfin[j++]=infin[i++]; I cheid linfin(i) == 'i') { push (&stact, infin[i++]); else if (infin[i] = 'J') { while (stack, top) = -1 && stack. items [stack.top postfin[j++]=pop(& stack); if (stack.top! = -1 && Stack.items (stack-top) mint flostack underflow? (1) = = pop(&stack); 3 i+ [+; (3+62)] runt (-2 ment), else { while (stack. top! = - 1 && precedence (stack. items (stack. top)) > = precedence (infin(i)) {

postfor (j++) = pop (& stack); push (& stack, infin(i++1): lencer Guene: while (stack. top! =-1) { of include estation ho postfin(j++) - pop(& stock) postfin[j]='10'; void engleene (int n) Char infin [MAX-52 28]; int main() { Char postfin [MAX-SIZE]) Prints ("Enter infin enpression"); Slamb ("d, s", infin); infin-to-postfin (infin, postfin); printf (" postfin enpression: 9.5", postfin); printy ("Vodey law"): Autput: Enter infin enpression: (A+B) = C-(D/E)

Postlin Postfin enpression: AB+C+DE/void displayes [(front == -1) Printle ("Underflow");