21bcm061

Practical 2:

Q2

// • Write a program to reverse a given string

// using stack.

**#include** <stdio.h>

**#include** <string.h>

**#define** max 100

**int** top, stack[max];

**void** push(**char** *x*)

{

**if** (top **==** max **-** 1)

{

printf("stack overflow");

}

**else**

{

stack[**++**top] **=** x;

}

}

**void** pop()

{

printf("%c", stack[top**--**]);

}

**int** main()

{

**char** str[] **=** "Sumeet";

**int** len **=** strlen(str);

**int** i;

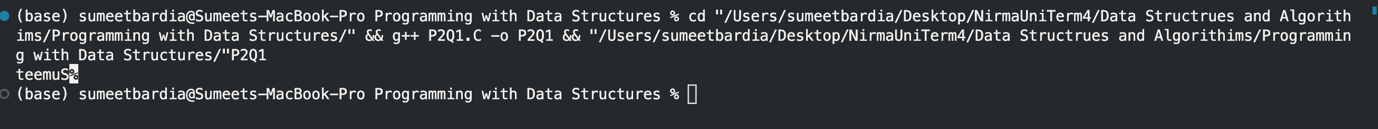
**for** (i **=** 0; i **<** len; i**++**)

push(str[i]);

**for** (i **=** 0; i **<** len; i**++**)

pop();

}



// Write a program to convert fully

// parenthesized infix expression to postfix expression

**#include**<stdio.h>

**#include**<ctype.h>

**char** stack[100];

**int** top **=** **-**1;

**void** push(**char** *x*)

{

stack[**++**top] **=** *x*;

}

**char** pop()

{

**if**(top **==** **-**1)

**return** **-**1;

**else**

**return** stack[top**--**];

}

**int** priority(**char** *x*)

{

**if**(*x* **==** '(')

**return** 0;

**if**(*x* **==** '+' **||** *x* **==** '-')

**return** 1;

**if**(*x* **==** '\*' **||** *x* **==** '/')

**return** 2;

**return** 0;

}

**int** main()

{

**char** exp[100];

**char** **\***e, x;

printf("Enter the expression : ");

scanf("%s",exp);

printf("\n");

e **=** exp;

**while**(**\***e **!=** '\0')

{

**if**(isalnum(**\***e))

printf("%c ",**\***e);

**else** **if**(**\***e **==** '(')

push(**\***e);

**else** **if**(**\***e **==** ')')

{

**while**((x **=** pop()) **!=** '(')

printf("%c ", x);

}

**else**

{

**while**(priority(stack[top]) **>=** priority(**\***e))

printf("%c ",pop());

push(**\***e);

}

e**++**;

}

**while**(top **!=** **-**1)

{

printf("%c ",pop());

}**return** 0;

}

