**Data Structure and Algorithm Practicals**

3. Reverse a string using stack

// Stack class

class Stack {

// Array is used to implement stack

constructor()

{

this.items = [];

}

// push function

push(element)

{

// push element into the items

this.items.push(element);

}

// pop function

pop()

{

// return top most element in the stack

// and removes it from the stack

// Underflow if stack is empty

if (this.items.length == 0)

return "Underflow";

return this.items.pop();

}

// peek function

peek()

{

// return the top most element from the stack

// but does'nt delete it.

return this.items[this.items.length - 1];

}

// isEmpty function

isEmpty()

{

// return true if stack is empty

return this.items.length == 0;

}

// printStack function

printStack()

{

var str = "";

for (var i = 0; i < this.items.length; i++)

str += this.items[i] + " ";

return str;

}

}

// Performs Postfix Evaluation on a given exp

function rev(exp)

{

var stack = new Stack();

for (var i = 0; i < exp.length; i++)

{

var c = exp[i];

stack.push(c);

}

var str="";

while(!stack.isEmpty())

{

str=str+ stack.pop();

}

console.log(str);

}

// calling the above method

// returns 9

rev("235\*+8-");