// print 1-n with normal recursion

let n = 4;

var f = function(i,n){

    if(i>n) return;

    console.log(i);

    f(i+1,n);

}

f(1,n);

// print the number 1-N using backtracking

let n = 4;

var f = function (i, n) {

  if (i < 1) {

    return;

  }

  f(i-1,n); // first call the function then print the number

  console.log(i);

};

f(n, n);

// print N to 1 using backtracking

var f = function(i,n){

    if(i>n) return;

    f(i+1,n);

    console.log(i);

}

let n = 10;

f(1,n);

// summation of numbers from 1-N using parameterized method

// in parameterrized method, we pass the needed parameter as a param in the next function call

// in below example we are doing summation from N to 1

var summation = function(i,sum){

    if(i==0){

        console.log(sum);

        return;

    }

    summation(i-1,sum+i);

}

let num = 5;

summation(num,0);

// summation using function returning a value

var summation = function(n){

    if(n==0){

        return 0;

    }

    return n + summation(n-1);

}

console.log(summation(5));

// factorial of a number using recursion

var fact = function(n){

    if(n==1){

        return 1;

    }

    return n\*fact(n-1);

}

console.log(fact(5));