1.

->

function double2(x){

    return (x\*x);

  }

  function square2(x){

    return (x\*2);

  }

  function compositeValue(double,square,x){

    return double2(square2(x)) ;

  }

  double2();

  square2();

  console.log(compositeValue(double2(),square2(),5));

2.

->

function doubleD(x){

  return (x\*x);

}

function squareD(x){

  return (x\*2);

}

function composeA(x){

  return doubleD(squareD(x)) ;

}

function composeB(x){

  return squareD(doubleD(x)) ;

}

console.log(composeA(5));

console.log(composeB(5));

3.

->

function isEven(x){

  return (x%2==0);

}

function findD(arr){

  for ( i = 0; i < arr.length; i++) {

    var res=isEven(arr[i]);

    if(res === true) {

      return arr[i];

      break;

    }

  }

}

console.log(findD([3,4,5,7]));

4.

->

function squareD(x){

  return (x\*x);

}

function map(arr,test){

  for(i=0;i<arr.length;i++){

    var res = squareD(arr[i]);

    newArr.push(res);

  }

  return newArr;

}

var newArr = [];

console.log(map([1,2,3,4],squareD));

5.

->

function printP(arr)

{

      var len = arr.length;

      var res = arr.slice(i, i+1);

      var res1 = isEven(res);

        if(res1 === true) {

          console.log(res)[0];

        }

        count = count +1;

        if(count <= len){

          i= i+1;

          printP(arr);

        }

}

function isEven(x){

  return (x%2==0);

}

var i = 0;

var count =0;

console.log(printP([1,3,4,5,6]));

6.

->

function printP(arr)

{

      var len = arr.length;

      var res = arr.slice(i, i+1);

      var res1 = squareD(res);

      var result = newarr.concat(res1);

      count = count +1;

      if(count <= len){

        i= i+1;

        newarr = result ;

        printP(arr);

      }

  return newarr;

}

function squareD(x){

  return (x\*x);

}

var newarr = [];

var i = 0;

var count =0;

console.log(printP([1,2,3,4]));

7.

->

function square(x){

  return (x\*x);

}

console.log(square.toString());