```
/**
     * calculate num! 1*2*3*…*num
*/

public static int factorial(int num) //(method from lectures)
    {
        if(num==0)
            return 1;
        else
            return num*factorial(num-1);
    }
-----------------------------------------------------------------
/**
     * calculate sum 2 power num
*/

public static int twopn(int num)  {
      if(num==0)
            return 1;
      return 2*twopn(num-1);
    }
-----------------------------------------------------------------
/**
     * calculate sum 1+2+3+….+n
*/
    public static int sum(int num){
        if(num==1)
                  return 1;
         return(  num+sum(num-1));
        }
-----------------------------------------------------------------
/**
     * calculate greatest common divisor
*/
        public static int gcd(int m,int n){
            if(n==m)
                return n;
             if(n<m)
                   return gcd(n,m-n);
              return (gcd(m,n-m));
            }
```

```
/**
     * print input nums  in reverse
*/
public static void reverse() {

                Scanner scan=new Scanner(System.in);
                int a;

                      a=scan.nextInt();
                if(a!=0)
                   reverse();
                 System.out.println(a);
                }
```

```
/**
     * calculate minimum steps from x to y using *2 or +1
     */
public static int minOps (int x,int y){//from y to x
                        if (y / x < 2)
                                   return y - x;

                        if (y % 2 != 0)
                                    return (2 + minOps (x,(y-1)/2));
                        else
                                    return (1 + minOps (x,y/2));
            }
```

```
 public static int minOps(int x, int y)  //from x to y

    {
       if (y / x < 2)
           return y - x;
       return Math.min(minOps(x + 1, y) + 1, minOps(2 * x, y) + 1);

    }
```

```
------------------------------------------------------------------
/**
    * calculate number in nth position in Fibonacci sequence
*/
   public static int fibonacci(int n)
   {
      if( (n==1) || (n==2) )
         return 1;
      else
         return ( fibonacci(n-1) + fibonacci(n-2) );
   }
------------------------------------------------------------------

/**
    * calculate a*b with + and -
*/
public static int multiply(int a, int b)
   {
      if(b==0)
         return 0;
      else
         return (a + multiply(a,b-1) );
   }

------------------------------------------------------------------



/**
 * Computes the sum of 2 non-negative integers using only +1/-1
 */
   public static int add(int a, int b) {
      if (b == 0) {
         return a;
      }
      else {
         return add(a+1, b-1);
      }
   }
------------------------------------------------------------------
/**
    *  calculate log n
    */
   public static int log(int n)
   {
        if(n<2) return 0;
        return 1+log(n/2);
      }
```

```
/**
     * calculate sum of all digits
     */
        public static int sumDigits(int n)
    {
           if(n==0) return 0;
           return sumDigits(n/10)+n%10;
        }
```

```
// has sqrt?
public static boolean f (int n)
    {
        return g(n,n);
    }
    private static boolean g (int n, int x)
    {
        if (n>x*x)
            return false;
        if (n<x*x)
            return g(n, x-1);
        return true;
    }
```

```
/**
     * get left most digit
     */

 public static int leftDigit(int n)
    {
           if(n<0) return leftDigit(-n);
           if(n<10) return n;
           return leftDigit(n/10);
        }
```

```
    /**
     * Prints a positive integer in reverse
     */
    public static void reverseDigits(int number)
    {
        System.out.print( number%10 );
        if( number/10 != 0)
            reverseDigits(number/10);
    }
```

```
------------------------------------------------------------------
/**
     * print binary of n
     */

public static void binary(int n){
if(n>0)
{
   binary(n/2);
   System.out.print(n%2);
}}
------------------------------------------------------------------


    /**
     * count ways from x , y to 0,0 only down or left
     */
    public static int count (int x, int y)
    {
        if( (x==0) || (y==0) )
            return 1;
        else
            return ( count(x-1,y) + count(x,y-1) );
    }
```