

```
/**
 * Nombres entiers et boucles.
 *
 * @author biech153 (Biersbach Chris) / gamca174 (Gamboa Carlos) / olial319 (Olinger Alex)
 * @version 25/04/2019 7:00:50
 * Classe: 3GIG
 */
public class SimpleCalculationWithTwoInts
{
    public void printAll(int a, int b)
    {
        for (int i=a; i<=b; i++)
        {
            System.out.println(i);
        }
    }

    public int calculateSumMultiple(int a, int b, int multiple)
    {
        int sum = 0;

        for (int i=a; i<=b; i++)
        {
            if (i % multiple == 0)
            {
                sum = sum + i;
            }
        }

        return sum;
    }

    public int calculateSumOdd(int a, int b)
    {
        int sum = 0;

        for (int i=a; i<=b; i++)
        {
            if (i % 2 == 1)
            {
                sum = sum + i;
            }
        }

        return sum;
    }

    public int calculateSumEven(int a, int b)
    {
        return calculateSumMultiple(a, b, 2);
    }

    public double calculatePower(int a, int b)
    {
        double res = 1;

        for (int i=1; i<=b; i++)
        {
            res = res * a;
        }

        return res;
    }

    public int GCD_search(int a, int b)
    {
        int gcd = Math.min(a, b);
        if (gcd == 0)
            return Math.max(a, b);

        while ((a%gcd != 0) || (b%gcd != 0))
        {
            gcd--;
        }
        return gcd;
    }

    // ... suite page suivante ...
}
```

```
public int GCD_Euclid(int a, int b)
{
    // il ne faut pas modifier les paramètres
    int tempA = a;
    int tempB = b;

    if (tempA == 0)
    {
        return tempB;
    }
    else
    {
        while (tempB != 0)
        {
            if (tempA > tempB)
                tempA = tempA - tempB;
            else
                tempB = tempB - tempA;
        }
        return tempA;
    }
}

public int GCD_OptimizedEuclid(int a, int b)
{
    // il ne faut pas modifier les paramètres
    int tempA = a;
    int tempB = b;
    int h;

    while (tempB != 0)
    {
        h = tempA % tempB;
        tempA = tempB;
        tempB = h;
    }
    return tempA;
}

public int LCM_search(int a, int b)
{
    if ((a == 0) || (b == 0))
        return 0;

    int lcm = Math.max(a, b);
    while ((lcm % a != 0) || (lcm % b != 0))
    {
        lcm = lcm + 1;
    }
    return lcm;
}

public int LCM_shortcut(int a, int b)
{
    if ((a == 0) || (b == 0))
        return 0;

    return (a*b) / GCD_search(a, b);
}
}
```