

```
/**
 * Nombres entiers et boucles.
 *
 * @author gamca174 (Gamboa Carlos) / olial319 (Olinger Alex)
 * @version 10/03/2016 07:13:16
 * Classe: 11TG
 */
public class SimpleCalculationWithTwoInts
{
    private int a;
    private int b;

    public SimpleCalculationWithTwoInts(int pA, int pB)
    {
        a = Math.abs(pA);
        b = Math.abs(pB);
    }

    public void printAll()
    {
        if (a > b)
        {
            swap();
        }

        int i = a;
        while (i <= b)
        {
            System.out.println(i);
            i++;
        }
    }

    public int calculateSumEven()
    {
        if (a > b)
        {
            swap();
        }

        int sum, i;
        sum = 0;

        i = a;
        while (i <= b)
        {
            if (i % 2 == 0)
            {
                sum = sum + i;
            }
            i++;
        }

        return sum;
    }

    public double calculatePower()
    {
        double res = 1;
        int i;

        // pas de swap car a exp b est différent de b exp a

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

        return res;
    }

    public void swap()
    {
        int tmp;

        tmp = a;
        a = b;
        b = tmp;
    }
}
```

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

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

public int GCD_Euclid()
{
    // il ne faut pas modifier les attributs
    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()
{
    // il ne faut pas modifier les attributs
    int tempA = a;
    int tempB = b;
    int h;

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

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

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