

```
public class AdvancedFiveSix {
```

```
    public static void main(String[] args) {
```

```
        int smallestMult = lcmUpTo(20);
```

```
        System.out.println("5. " + smallestMult);
```

```
        int diff = squareOfSum(100) - sumOfSquares(100);
```

```
        System.out.println("6. " + diff);
```

```
    }
```

```
    public static int lcmUpTo(int num) {
```

```
        // gives the smallest multiple of all numbers up to num
```

```
        int currLcm = 1;
```

```
        int currGcf = 1;
```

```
        for (int j = 1; j <= num; j++) {
```

```
            currGcf = gcf(j, currLcm); // finds the gcf of the current number being checked and the product so far
```

```
            currLcm *= (j / currGcf); // multiplies by the number divided by the gcf from the last step so no duplicated factors occur
```

```
        }
```

```
        return currLcm;
```

```
    }
```

```
    public static int gcf(int first, int second) {
```

```
        int a = Math.max(first, second);
```

```
        int b = Math.min(first, second);
```

```
        int r = a % b;
```

```
        if (r == 0) {
```

```
    return b; // base case if a is divisible by b
} else {
    return gcf(b, r); // implements Euclid's algorithm recursively
}
}
```

```
public static int sumOfSquares(int num) {
    int total = 0;
    for (int i = 1; i <= num; i++) {
        total += Math.pow(i, 2); // adds the squares of all numbers up to and including num
    }
    return total;
}
```

```
public static int squareOfSum(int num) {
    double sum = (num * (1 + num)) / 2; // formula for the sum of numbers up to num
    double returnSum = Math.pow(sum, 2); // square the above
    return (int) returnSum;
}
}
```