# Code Challenge 06


## Disclaimer

*By reading this problem statement, I agree not to copy code from any source (including websites, books, or friends and colleagues) to complete this assessment. I also agree not to share this code challenge/assessment through any means. I may, however, reference programming language documentation or use an IDE that has code completion features.*

## Code Challenge

### Risk Assessment

Some statistics and data points are required to evaluate the risk of defaulting by potential new customers; these are the standard deviation of past late payments, unusual late payment, and probability of late payments.

The unusual late payment is detected when there is a very late payment *(number of days after the payment due date)*, and the number of days of late payments *before* and *after* this day is **smaller**; graphically, you would see a peak. For this data, we require the largest peak found *(max peak)*.

To calculate the probability of late payments, we must use the late payments of other products the client already has, given a number of time periods.

#### Input

For both the standard deviation and unusual late payment *(max peak)*, you will receive an integer array containing the number of days of delays in payments.

Each index of the array represents a payment period, usually a month, `0` meaning no delay, `1` meaning one day of delay in payment, and so on.

To calculate the late payment probability, you will receive a two dimensional array, where every row represents the late payments of a product for different time periods *(columns)*.

## Expected output

### Expected output for Standard deviation

The expected output for the standard deviation is a `double`, representing the result of calculating the standard deviation (*The population standard deviation*).

### Expected output for unusual late payment *(max peak)*

The expected output is an `int` representing the index of the array where the *max peak* is found.

### Expected output for late payment probability

The expected output is a `double` array where each position contains the late payment probability for the period *(column index)* of the different products *(rows of the array)*.

## Example

### Sample input

```
payment delays array:
{0, 15, 2, 0, 1, 3}
[0  1   2  3  4  5] //index
```

### Sample input for late payment probability

```
A two-dimensional array, each row is a product, each column is a time period.
{
    {0, 3, 6, 1, 0, 5, 0, 0}, //product 1
    {0, 3, 0, 2, 0, 0, 0, 2}, //product 2
    {0, 0, 1, 0, 3, 0, 2, 0}, //product 3
    {0, 4, 0, 2, 0, 1, 1, 0}  //product 4
    [0  1  2  3  4  5  6  7]  //index (time period)
}
```

### Sample output

### Standard deviation

```
5.251983752
```

**Unusual late payment *(max peak)***

```
1 //index of the array where the max peak is located.
```

**Late payment probability**

```
{0.0, 0.75, 0.5, 0.75, 0.25, 0.5, 0.5, 0.25}
```

**Explanation**

The standard deviation for these numbers `{0, 15, 2, 0, 1, 3}` is `5.251983752`.

The *max peak* for the array is located at index `1` since `15 > 0 and 15 > 2` index 5 also has a peak but `3` is smaller than `15`.

Let see the first two periods *(indices 0 and 1 of the two-dimensional array)* for the late payment probability. For the first period *(index 0)*, we see no late payments in any of the `4` products, so the result is `0.0` `0/4 = 0.0`. For the second period *(index 1)*, we see three late payments *(3, 3, 4)*; this gives us a result of `0.75` `3/4 = 0.75`.

## Completing the code challenge

To complete the code challenge, you must complete the methods `standardDeviation`, `paymentDelayMaxPeakIndex` and `latePaymentProbabilityByPeriod` inside the `CreditRiskAssessment` class so that every unit test in `CreditRiskAssessmentTest` passes.

*Bonus: Document the source code to get bonus points.*

Good luck!