Instructions

1. Implement the `JMT(start, end, T)` function in `main.cpp`
2. Hit `Test Run` and see if you're correct!

Tips

Remember, you are solving a system of equations: matrices will be helpful! The Eigen library used from Sensor Fusion is included.

The equations for position, velocity, and acceleration are given by:

$$s(t) = s_i + \dot{s}_i t + \frac{\ddot{s}_i}{2} t^2 + \alpha_3 t^3 + \alpha_4 t^4 + \alpha_5 t^5$$

$$\dot{s}(t) = \dot{s}_i + \ddot{s}_i t + 3\alpha_3 t^2 + 4\alpha_4 t^3 + 5\alpha_5 t^4$$

$$\ddot{s}(t) = \ddot{s}_i + 6\alpha_3 t + 12\alpha_4 t^2 + 20\alpha_5 t^3$$

and if you evaluate these at $t = 0$ you find the first three coeffecients of your JMT are:

$[\alpha_0, \alpha_1, \alpha_2] = [s_i, \dot{s}_i, \frac{1}{2}\ddot{s}_i]$

and you can get the last three coefficients by evaluating these equations at $t = T$. When you carry out the math and write the problem in matrix form you get the following:

$$\begin{bmatrix} T^3 & T^4 & T^5 \\ 3T^2 & 4T^3 & 5T^4 \\ 6T & 12T^2 & 20T^3 \end{bmatrix} \times \begin{bmatrix} \alpha_3 \\ \alpha_4 \\ \alpha_5 \end{bmatrix} = \begin{bmatrix} s_f - (s_i + \dot{s}_i T + \frac{1}{2}\ddot{s}_i T^2) \\ \dot{s}_f - (\dot{s}_i + \ddot{s}_i T) \\ \ddot{s}_f - \ddot{s}_i \end{bmatrix}$$

All these quantities are known except for $\alpha_3, \alpha_4, \alpha_5$

main.cpp | grader.h

```cpp
#include <cmath>
#include <iostream>
#include <vector>

#include "Dense"
#include "grader.h"

using std::vector;
using Eigen::MatrixXd;
using Eigen::VectorXd;

/**
 * TODO: complete this function
 */
vector<double> JMT(vector<double> &start, vector<double> &end, double T) {
  /**
   * Calculate the Jerk Minimizing Trajectory that connects the initial state
   * to the final state in time T.
   *
   * @param start - the vehicles start location given as a length three array
   *   corresponding to initial values of [s, s_dot, s_double_dot]
   * @param end - the desired end state for vehicle. Like "start" this is a
   *   length three array.
   * @param T - The duration, in seconds, over which this maneuver should occur.
   *
   * @output an array of length 6, each value corresponding to a coefficent in
   *   the polynomial:
   *   s(t) = a_0 + a_1 * t + a_2 * t**2 + a_3 * t**3 + a_4 * t**4 + a_5 * t**5
   *
   * EXAMPLE
```

RESET QUIZ | TEST RUN | SUBMIT ANSWER