We would like to see a sample of your coding ability. To understand where you would fit in at Odyssey Space Research, your submission will be evaluated by some of our senior developers with the same careful scrutiny that we apply to our developed products. Provide a solution that uses nothing more than standard C++ libraries; there are plenty of 3rd-party Matrix libraries out there, but simply wrapping one of those does not showcase your skills.

Write a C++ class that implements a matrix, with the following requirements:

-The matrix should be 2D n x m, with n and m set by the user when the object is constructed.

-Write a suitable overloaded C++ operator to get and set a specific element of the matrix. Helpful hint - the way you choose to satisfy this requirement will have a large impact on the design of your class. If you implement this last, you may find yourself re-writing a lot of your code.

-Access to the matrix should be unit-based, i.e. the first element of the first row would be referred to as 1 1 not 0 0. (This constraint may or may not seem arbitrary; sometimes customers come up with arbitrary requirements. Usually, we can't talk them out of it, because they have other reasons.)

-The matrix should support addition, scalar multiplication, at least one other operation of your choice, and matrix multiplication (i.e. given two matrices A and B, AB is the dot product of each row of A with each column of B; see https://en.wikipedia.org/wiki/Matrix_multiplaction if you need further details).

-Write unit tests and include them with your submission. You could use any test framework you're familiar with, or just write your own tests.

Beyond this, please try to impress us. This is our first chance to get to know you. We'll have at least 2 developers looking at your submission, so your efforts won't go unappreciated.

Our evaluation is inherently subjective, but we factor a lot of things into our analysis, including:

-Does the code compile? Does it do what it should? Does it pass all unit tests? Are the unit tests sufficient?

-How readable and maintainable is your code?

-Is your code robust enough to handle any edge cases?

-Do you exhibit the good habits that would be expected of a professional at your level of expertise? Do you follow (or fail to follow) C++ best practices?

-What sort of extra features to these bare requirements have you added?

-Is your code efficient?

-What design and implementation decisions did you make, and why did you make them? (You can help us here by adding relevant comments or other documentation.)

We have left these specifications deliberately vague. This is to give you a chance to make your own decisions about the implementation.

Submit all of your code as a zipped git repository. We will look at your git history.

We know you have better things to do than write code that will never be used, and we appreciate the time you're putting into this. We truly believe that this is a good way for us to get to know your skills. This entire task should be achievable in 2-6 hours. Please contact the person who sent this to you if you have any questions.