

```

void multiply(const Matrix& left, const Matrix& right, Matrix& product)
{
    // Check for empty left or empty right.
    if (left.size() == 0 || right.size() == 0) {
        product.resize(0);
        return;
    }

    // Get dimensions. We hope left is m-by-n, right is n-by-p.
    int m = left.size();
    int n = left.at(0).size();
    int p = right.at(0).size();

    // Check for compatible dimensions.
    if (n != right.size() || p == 0) {
        product.resize(0);
        return;
    }

    // Super-careful checks: Do all rows have the right sizes?
    for (int i = 1; i < m; i++)
        if (left.at(i).size() != n) {
            product.resize(0);
            return;
        }
    for (int j = 1; j < n; j++)
        if (right.at(j).size() != p) {
            product.resize(0);
            return;
        }

    // Make product the right size.
    product.resize(m);
    for (int i = 0; i < m; i++)
        product.at(i).resize(p);

    // Finally, we know it's safe to multiply.
    for (int i = 0; i < m; i++)
        for (int j = 0; j < p; j++) {
            product.at(i).at(j) = 0.0;
            for (int k = 0; k < n; k++)
                product.at(i).at(j) += left.at(i).at(k) * right.at(k).at(j);
        }
}

```