Matrix APSP_starting_from(Matrix& matrix, VertexID id):

```
    int n = matrix.size(); // size of matrix (# of vertices)

    Matrix d = matrix; // this is equal to D0 * W
    Matrix c(n,std::vector<int>(n - 1, 0)); // return matrix

    for (int j = 0; j < n; j++){
        // this loop is necessary to get the d[x, k] format
        c[j][0] = d[id][j];
    }

    for (int i = 1; i < n - 1; i++){
        // basic APSP algorithm
        d = matrix_extend(d, matrix);

        // loop for d[x, k] format again
        for (int j = 0; j < n; j++){
            c[j][i] = d[id][j];
        }
    }
    return c;
```

Matrix matrix_extend(Matrix& d, Matrix& w):

```
int n = d.size(); // size of matrix (# of vertices)

for (int i = 0; i < n; i++){
    for (int j = 0; j < n; j++){
        for (int k = 0; k < n; k++){
            // calculation magic
            d[i][j] = std::min(d[i][j], d[i][k] + w[k][j]);
        }
        if (d[j][j] < 0){
            // having a negative value on the diagonal means there's
            // a negative cycle
            throw std::out_of_range("Negative cycle detected!");
        }
    }
}
return d;
```