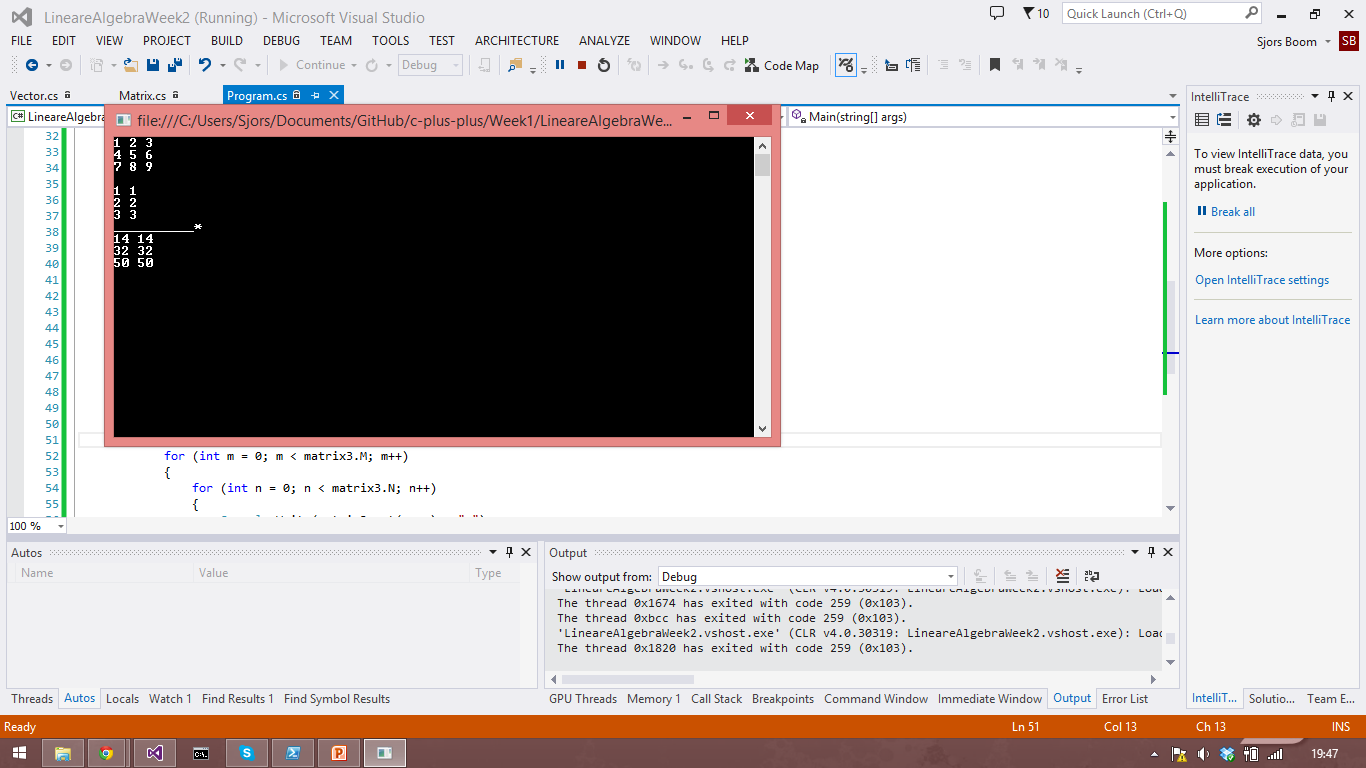
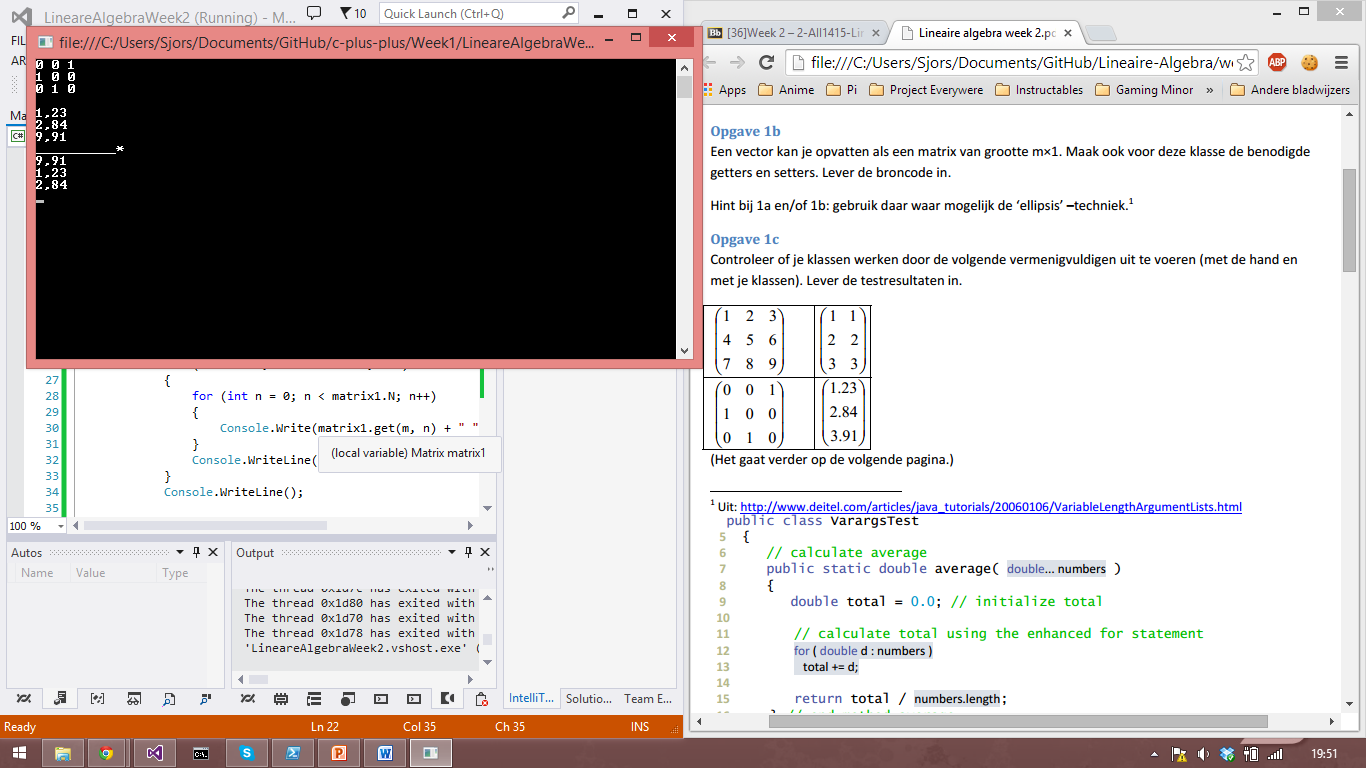
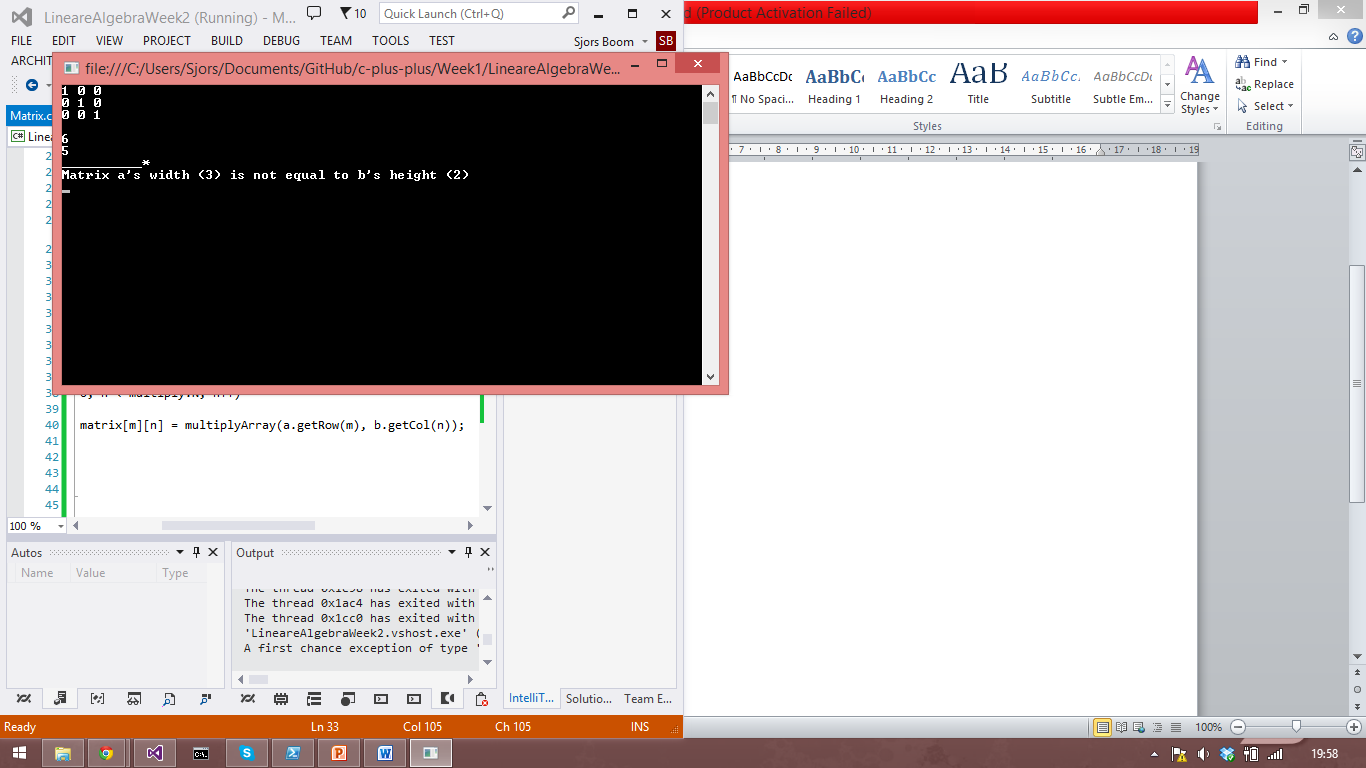
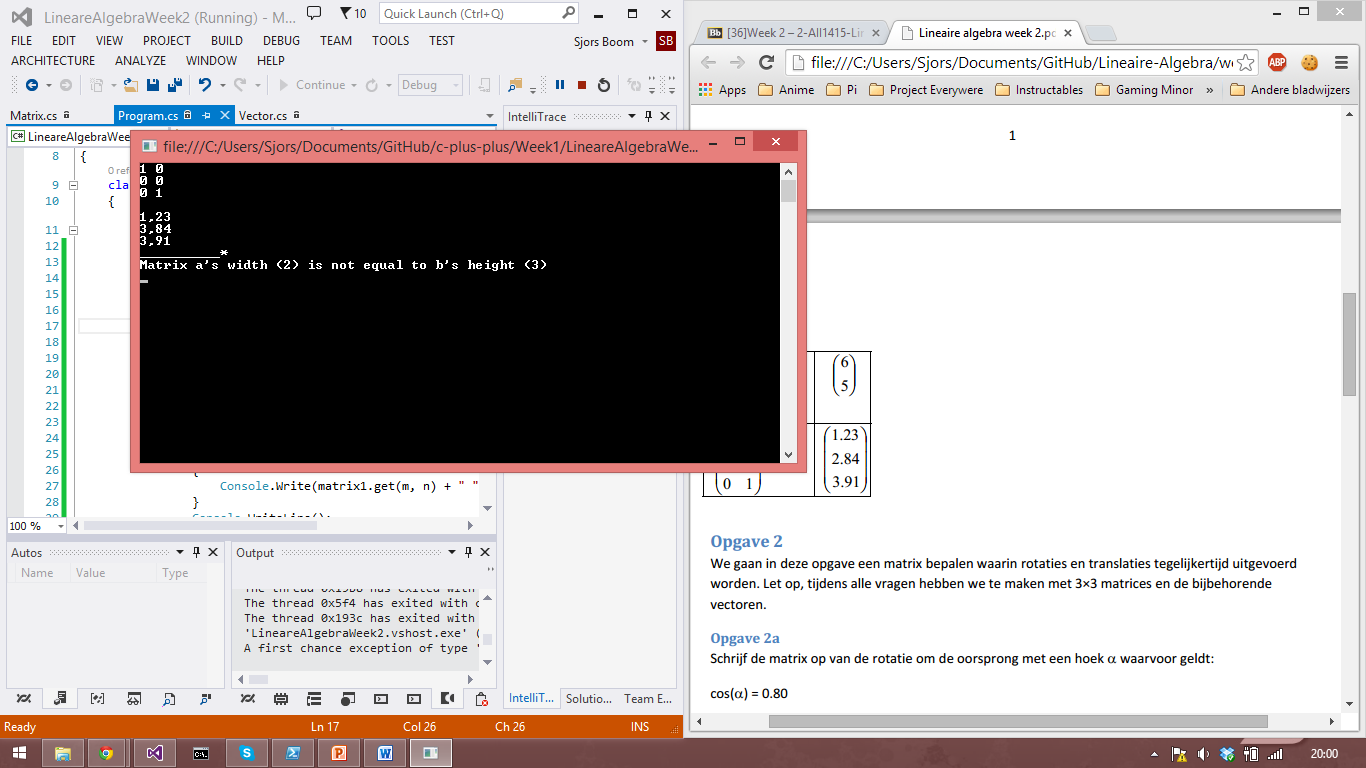
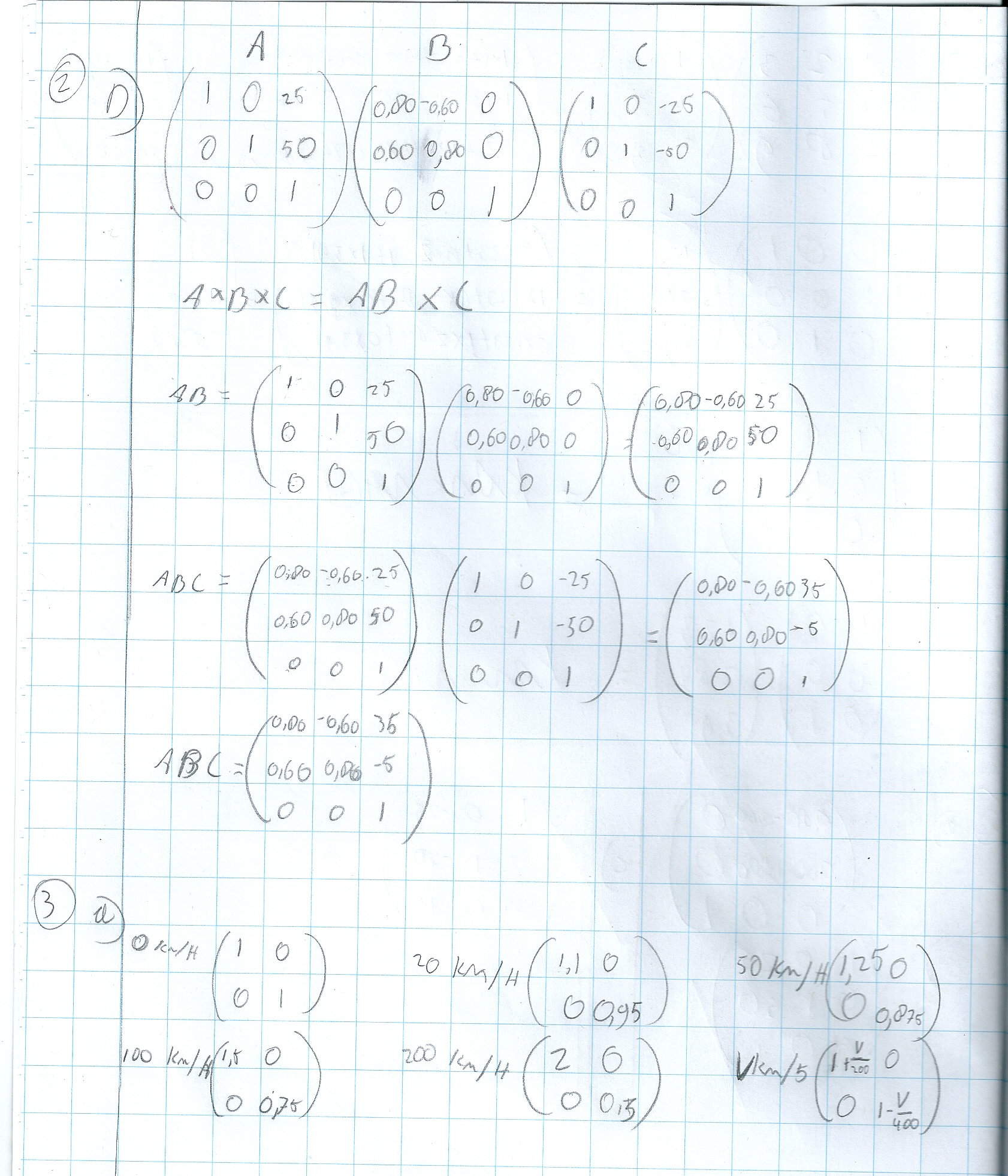
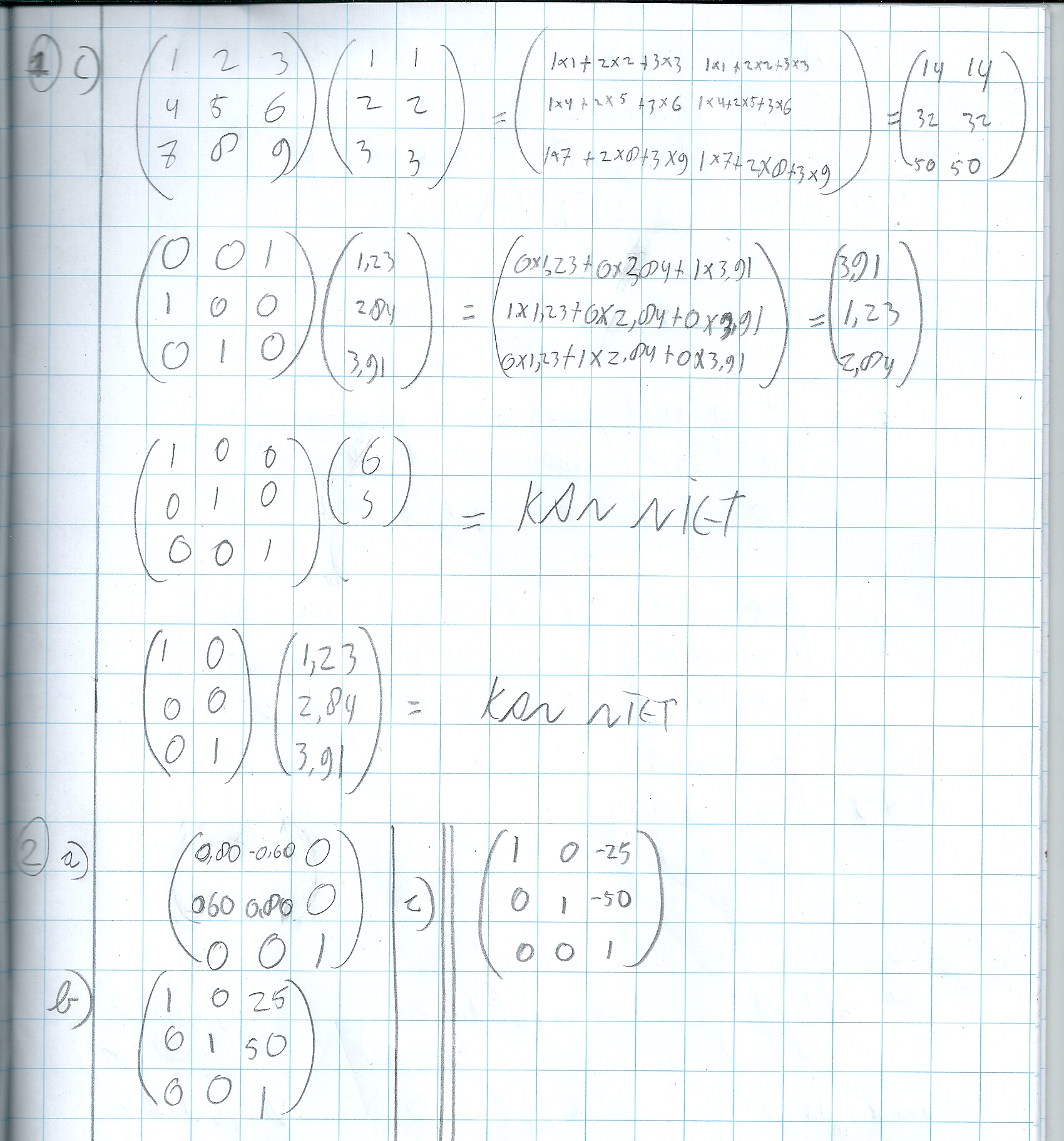
# Opgave 1

## C







class Matrix

{

double[][] matrix;

public int M { get; private set; }

public int N { get; private set; }

public Matrix(int m, int n)

{

M = m;

N = n;

matrix = new double[M][];

for (int i = 0; i < m; i++)

{

matrix[i] = new double[N];

}

}

public static Matrix operator \*(Matrix a, Matrix b)

{

if (a.N != b.M)

{

throw new Exception("Matrix a's width (" + a.N + ") is not equal to b's

height (" + b.M + ")");

}

Matrix multiply = new Matrix(a.M, b.N);

for (int m = 0; m < multiply.M; m++)

{

for (int n = 0; n < multiply.N; n++)

{

multiply.matrix[m][n] = multiplyArray(a.getRow(m), b.getCol(n));

}

}

return multiply;

}

private static double multiplyArray(double[] row, double[] col)

{

double result = 0;

for (int i = 0; i < row.Length; i++)

{

result += row[i] \* col[i];

}

return result;

}

public double[] getCol(int n)

{

double[] col = new double[M];

for (int i = 0; i < M; i++)

{

col[i] = matrix[i][n];

}

return col;

}

public double[] getRow(int m)

{

return matrix[m];

}

public double get(int m, int n)

{

return matrix[m][n];

}

public void set(int m, int n, double value)

{

matrix[m][n] = value;

}

public static Matrix scale(double velocity)

{

Matrix scale = new Matrix(2, 2);

scale.set(0, 0, 1 + velocity / 200);

scale.set(1, 1, 1 - velocity / 400);

return scale;

}

}