**Projekt 1**

Zadanie : Przeprowadzenie eksperymentów

Kod wykorzystany w eksperymencie:

using System;

using System.Diagnostics;

namespace Zadanie

{

class Program

{

static ulong OpComparisonEQ;

static int[] TestVector;

const int NIter = 10;

static void Main(string[] args)

{

{

static bool IsPresent\_LinearTim(int[] Vector, int Number)

{

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

if (Vector[i] == Number)

return true;

return false;

}

static bool IsPresent\_LinearInstr(int[] Vector, int Number)

{

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

{

OpComparisonEQ++; if (Vector[i] == Number)

return true;

}

return false;

}

static bool IsPresent\_BinaryTim(int[] Vector, int Number)

{

int Left = 0, Right = Vector.Length - 1, Middle;

while (Left <= Right)

{

Middle = (Left + Right) / 2;

if (Vector[Middle] == Number) return true;

else if (Vector[Middle] > Number) Right = Middle - 1;

else Left = Middle + 1;

}

return false;

}

static bool IsPresent\_BinaryInstr(int[] Vector, int Number)

{

int Left = 0, Right = Vector.Length - 1, Middle;

while (Left <= Right)

{

Middle = (Left + Right) / 2;

if (Vector[Middle] == Number)

return true;

else

{

if (Vector[Middle] > Number) Right = Middle - 1;

else Left = Middle + 1;

}

}

return false;

}

static void LinearMaxInstr()

{

OpComparisonEQ = 0;

bool Present = IsPresent\_LinearInstr(TestVector, TestVector.Length - 1);

Console.Write("\t" + OpComparisonEQ);

}

static void LinearMaxTim()

{

double ElapsedSeconds;

long ElapsedTime = 0, MinTime = long.MaxValue, MaxTime = long.MinValue, IterationElapsedTime;

for (int n = 0; n < (NIter + 1 + 1); ++n)

{

long StartingTime = Stopwatch.GetTimestamp();

bool Present = IsPresent\_LinearTim(TestVector, TestVector.Length - 1);

long EndingTime = Stopwatch.GetTimestamp();

IterationElapsedTime = EndingTime - StartingTime; ElapsedTime += IterationElapsedTime;

if (IterationElapsedTime < MinTime) MinTime = IterationElapsedTime;

if (IterationElapsedTime > MaxTime) MaxTime = IterationElapsedTime;

}

ElapsedTime -= (MinTime + MaxTime);

ElapsedSeconds = ElapsedTime \* (1.0 / (NIter \* Stopwatch.Frequency));

Console.Write("\t" + ElapsedSeconds.ToString("F4"));

}

static void BinaryMaxInstr()

{

OpComparisonEQ = 0;

bool Present = IsPresent\_BinaryInstr(TestVector, TestVector.Length);

Console.Write("\t" + OpComparisonEQ);

}

static void LinearAvgInstr()

{

OpComparisonEQ = 0;

bool Present;

for (int i = 0; i < TestVector.Length; ++i) Present = IsPresent\_LinearInstr(TestVector, i);

Console.Write("\t" + ((double)OpComparisonEQ / (double)TestVector.Length).ToString("F1"));

}

Console.WriteLine("Size\tLMaxI\tLMaxT\tBMaxI\tBMaxT\tLAvgI\tLAvgT\tBAvgI\tBAvgT");

for (int ArraySize = 26843545; ArraySize <= 268435450; ArraySize += 26843545)

{

Console.Write(ArraySize); // tworzymy tablicę

TestVector = new int[ArraySize]; // wypełniamy tablicę

for (int i = 0; i < TestVector.Length; ++i) TestVector[i] = i; LinearMaxInstr(); // liniowe max instrumentacja

LinearMaxTim(); // liniowe max czas

BinaryMaxInstr(); // binarne max instrumentacja

//BinaryMaxTim(); // binarne max czas

LinearAvgInstr(); // liniowe średnia instrumentacja

//LinearAvgTim(); // liniowe średnia czas

//BinaryAvgInstr(); // binarne średnia instrumentacja

//BinaryAvgTim(); // binarne średnia czas

Console.Write("\n");

}

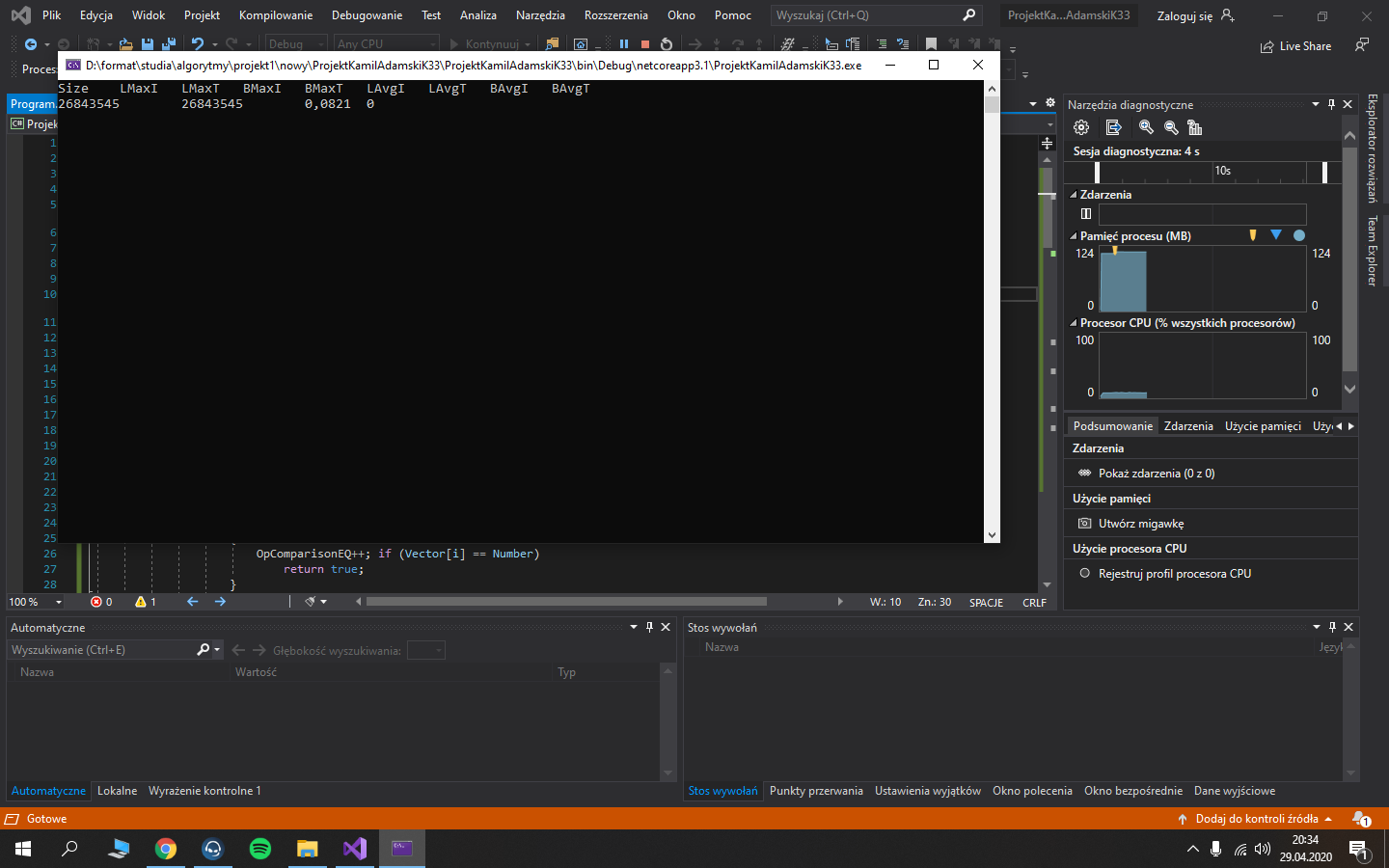
}

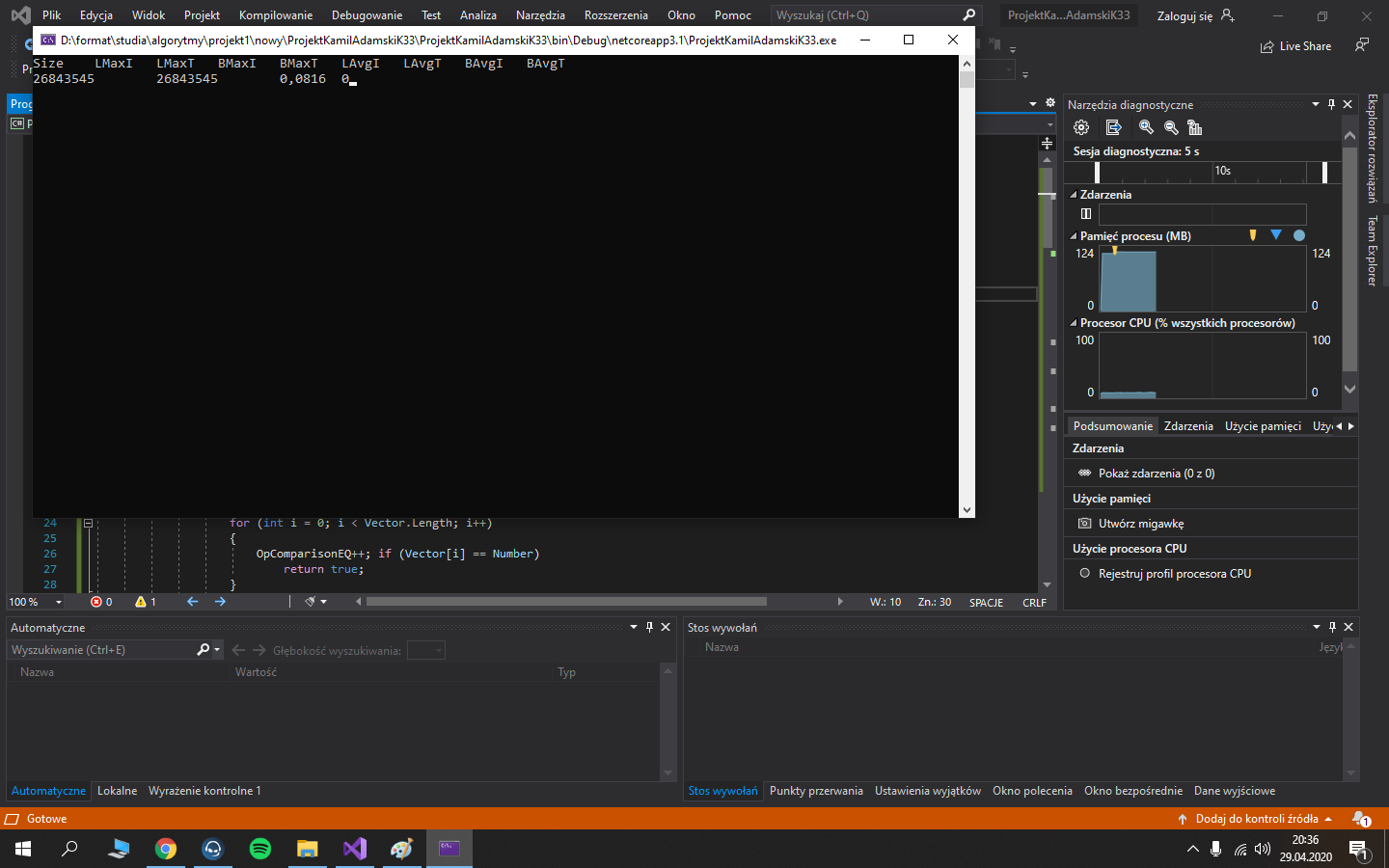
}

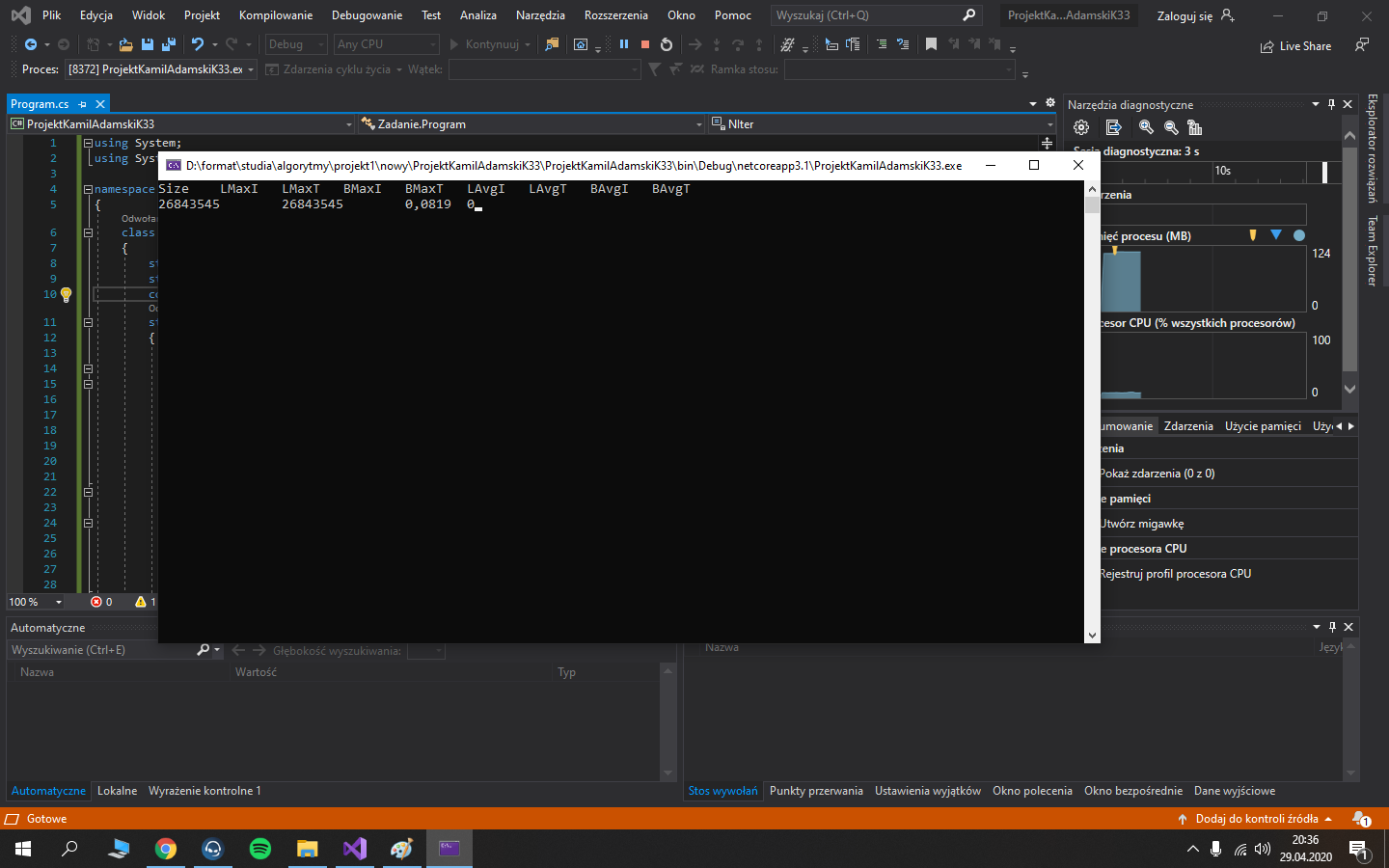
}

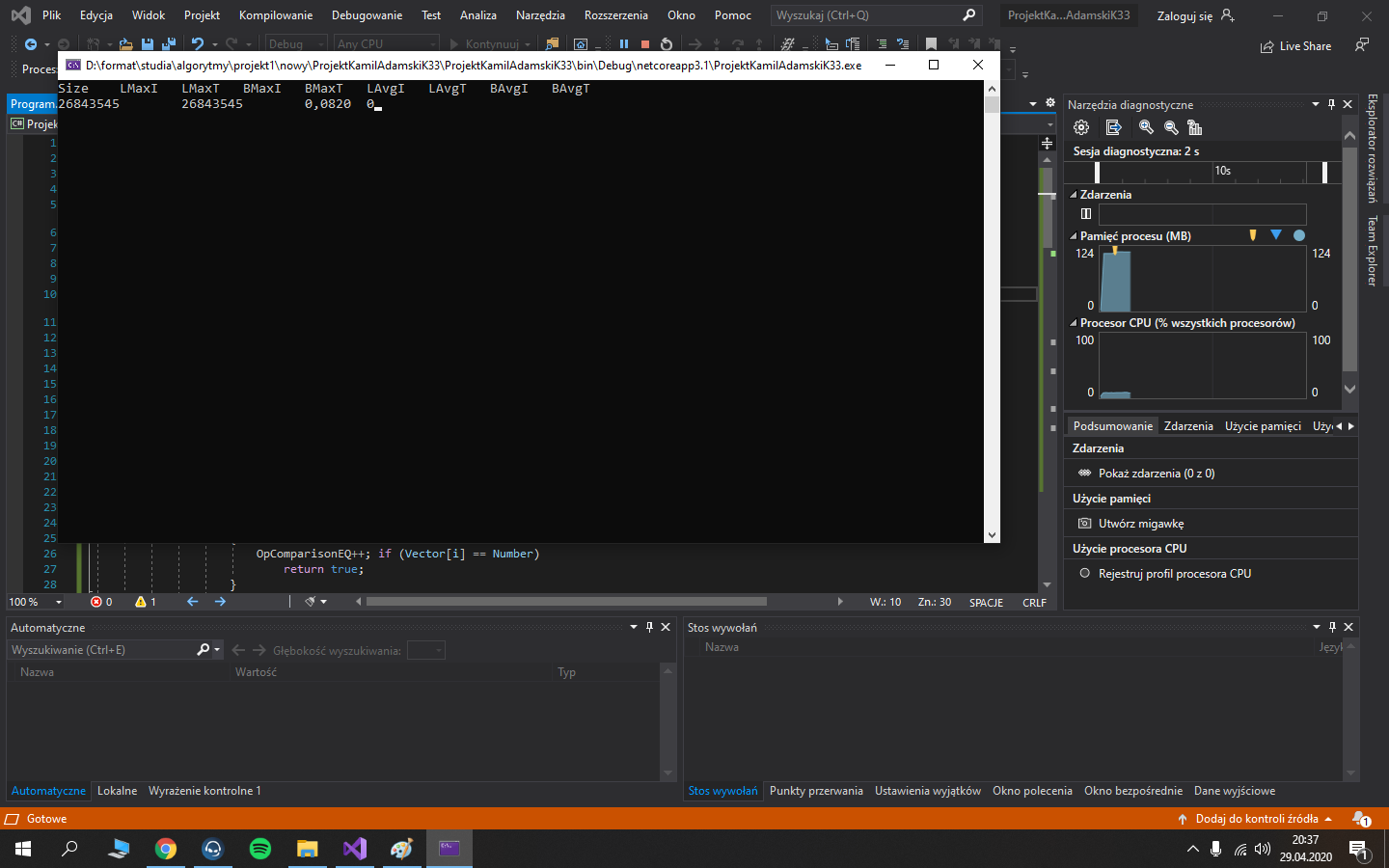
}

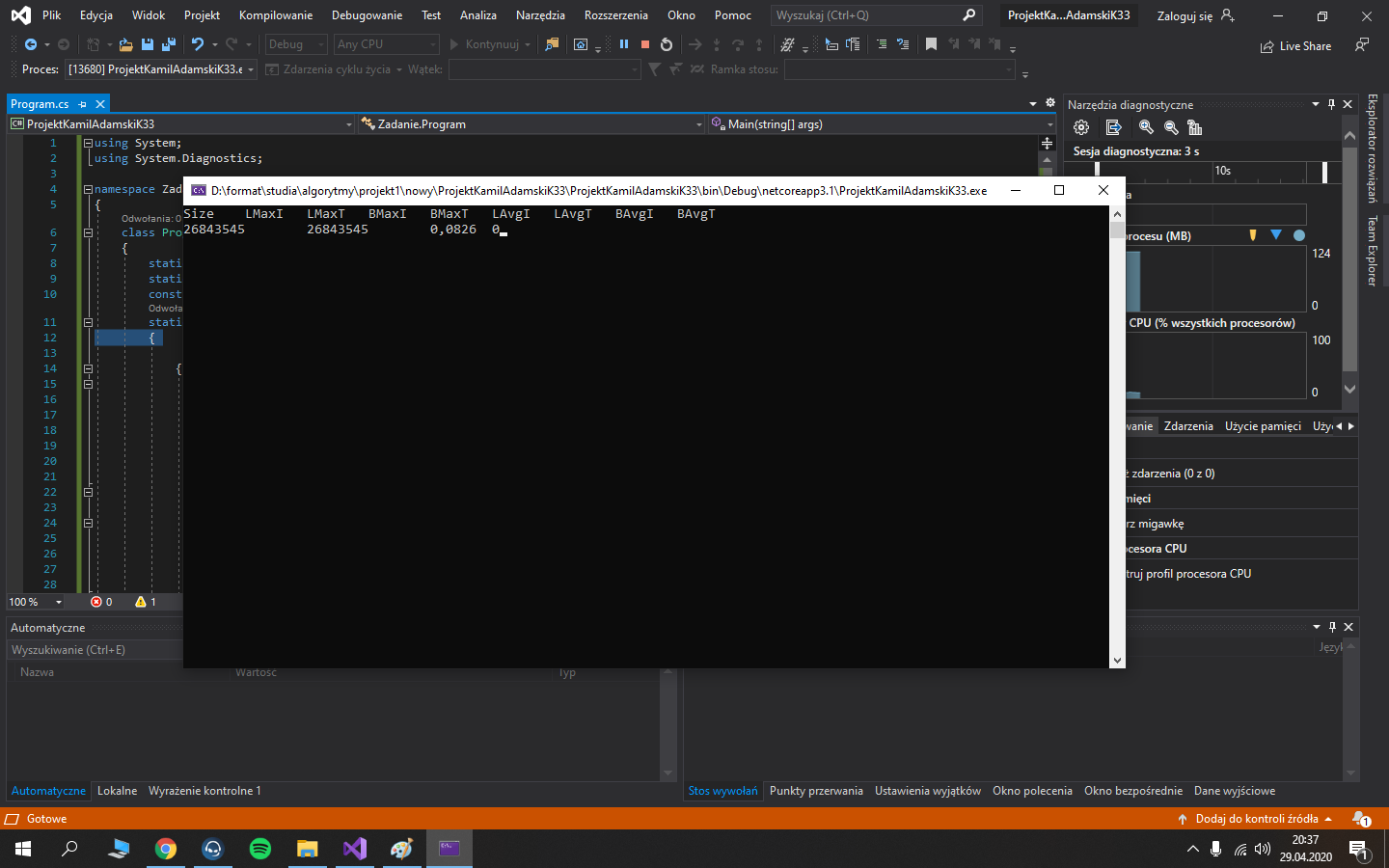
Wykonałem 5 prób pomiaru (ze względu na słabą jakość zdjęć w wordzie załączam je w folderze z projektem)











Wyniki pomiarów

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pomiar** | **Size** | **LMaxT** | **BMaxT** | **LAvgI** |
| 1 | 26843545 | 26843545 | 0,0821 | 0 |
| 2 | 26843545 | 26843545 | 0,0816 | 0 |
| 3 | 26843545 | 26843545 | 0,0819 | 0 |
| 4 | 26843545 | 26843545 | 0,0820 | 0 |
| 5 | 26843545 | 26843545 | 0,0826 | 0 |