МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ»

(национальный исследовательский университет)»

Институт №3 «Системы управления, информатика и электроэнергетика»

Кафедра 304 «Вычислительные машины, системы и сети»

Конструирование ПО

Отчет по курсовой работе по теме

«5.5 Сравнение временных параметров выполнения приложений в CLR и Python»

Выполнил студент группы М3О-325Бк-18:

Мазаев А.Г.

Проверил:

профессор к.304, к.т.н., Гагарин Андрей Петрович

Москва 2021

Оглавление

[Теория 3](#_Toc59847398)

[Вычисления 4](#_Toc59847399)

[Программа №1 4](#_Toc59847400)

[Программа №2 6](#_Toc59847401)

[Программа №3 11](#_Toc59847402)

[Программа №4 13](#_Toc59847403)

[Вывод 17](#_Toc59847404)

# Теория

**Python** — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой набор полезных функций.

Python — активно развивающийся язык программирования, новые версии с добавлением/изменением языковых свойств выходят примерно раз в два с половиной года. Язык не подвергался официальной стандартизации, роль стандарта де-факто выполняет [CPython](https://ru.wikipedia.org/wiki/CPython), разрабатываемый под контролем автора языка. В настоящий момент Python занимает второе место в рейтинге TIOBE с показателем 12,12 %. Аналитики отмечают, что это самый высокий балл Python за все время его присутствия в рейтинге.

Тем не менее, главное преимущество языка Python в том, что он прост и легок в изучении. Поэтому это один из самых популярных языков среди начинающих. Простой синтаксис помогает разработать даже сложный алгоритм для решения проблемы в течение короткого периода времени.

**CLR (**Common language runtime) — общеязыковая исполняющая среда. Она обеспечивает интеграцию языков и позволяет объектам благодаря стандартному набору типов и метаданным), созданным на одном языке, быть «равноправными гражданами» кода, написанного на другом.  
  
Другими словами CLR этот тот самый механизм, который позволяет программе выполняться в нужном нам порядке, вызывая функции, управляя данными. И все это для разных языков (c#, VisualBasic, Fortran). Да, CLR действительно управляет процессом выполнения команд (машинного кода, если хотите) и решает, какой кусок кода (функцию) от куда взять и куда подставить прямо в момент работы программы.

# **Вычисления**

# Используя однотипный код на двух разных языках выясним какой из них будет быстрее.

# **Программа №1**

3 раза находим произведение чисел.

**Проверка времени на языке Console line reading С#.**

**Код:**

class Program

{

static void Main(string[] args)

{

DateTime a = DateTime.Now;

{

Console.WriteLine(500.2345 \* 254.532);

Console.WriteLine(0.556 \* 2.534);

Console.WriteLine(545.144 \* 234343.43434);

}

//останавливаем счётчик

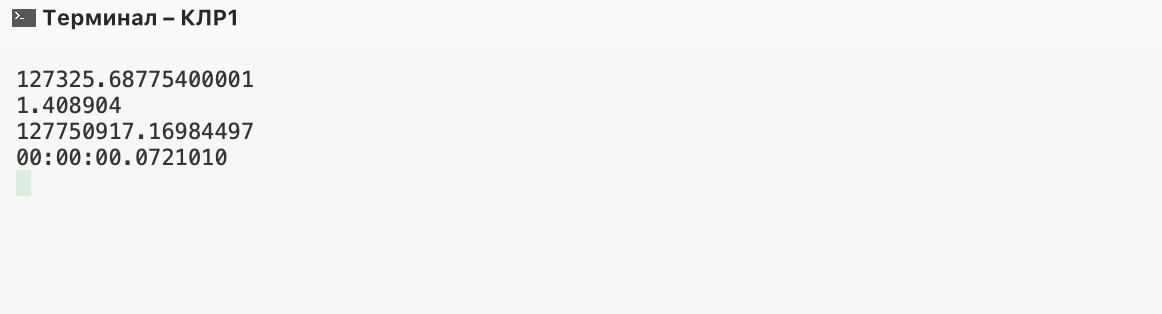
Console.WriteLine(DateTime.Now - a);

}

}

}

**Вывод:**

****

Время: 0.072 секунды.

**Проверка времени на языке Python.**

**Код:**

import time

def multiplication():

a=500.2345 \* 254.532

print(a)

b=0.556\*2.534

print(b)

c=545.144\*234343.43434

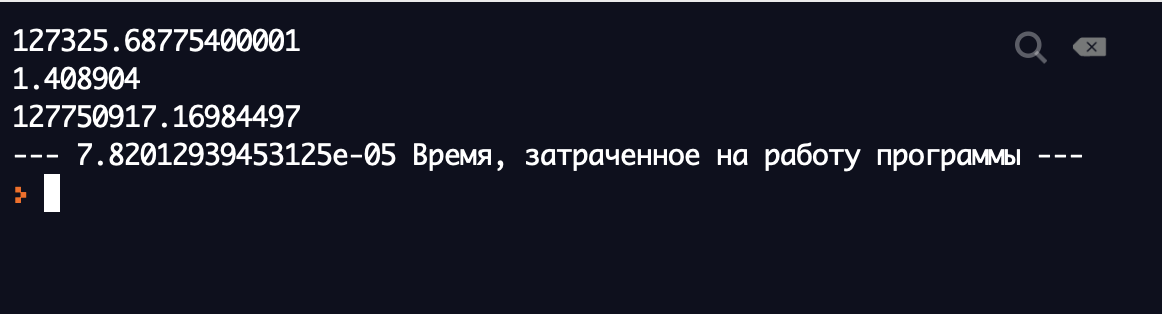
print(c)

start\_time = time.time()

multiplication()

print("--- %s Время, затраченное на работу программы ---" % (time.time() - start\_time))

**Вывод**

****

Время: **0,0000782 секунды.**

# 

# **Программа №2**

Считаем сумму от а до б включительно. А и б мы задаем самостоятельно,

a = 0, b = 2000000.

**Console line reading С#.**

**Код:**

using System;

using System.Diagnostics;

using System.Threading;

namespace CLR2

{

class Program

{

static int n\_min, n\_max;

static int sum;

static void Main(string[] args)

{

ввод();

сумма1();

вывод("Сумма чисел через цикл = ");

Console.ReadKey();

}

static void ввод()

{

Console.Write("начальное число = ");

n\_min = Convert.ToInt32(Console.ReadLine());

Console.Write("конечное число = ");

n\_max = Convert.ToInt32(Console.ReadLine());

}

static void сумма1()

{

DateTime a = DateTime.Now;

{

sum = 0;

for (int i = n\_min; i <= n\_max; i++)

sum = sum + i;

}

Console.WriteLine(DateTime.Now - a);

}

static void вывод(string z)

{

z = z + sum.ToString();

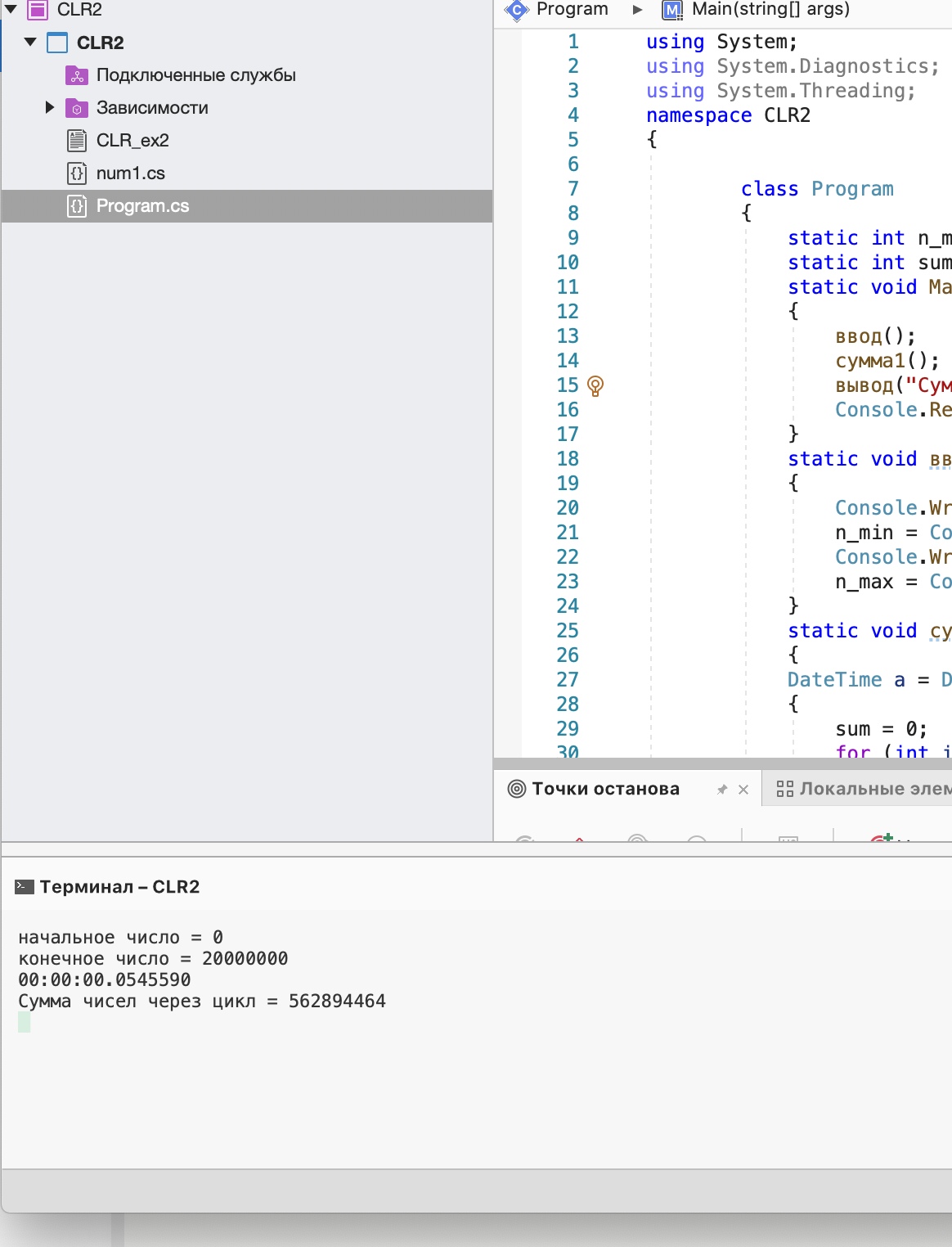
Console.WriteLine(z);

}

}

}

**Вывод:**



Время: **0.054 секунды**

**Проверка времени на языке Python.**

**Код:**

import time

a=int(input())

b=int(input())

def sum(a, b):

s=0

for i in range(a, b+1):

s=s+i

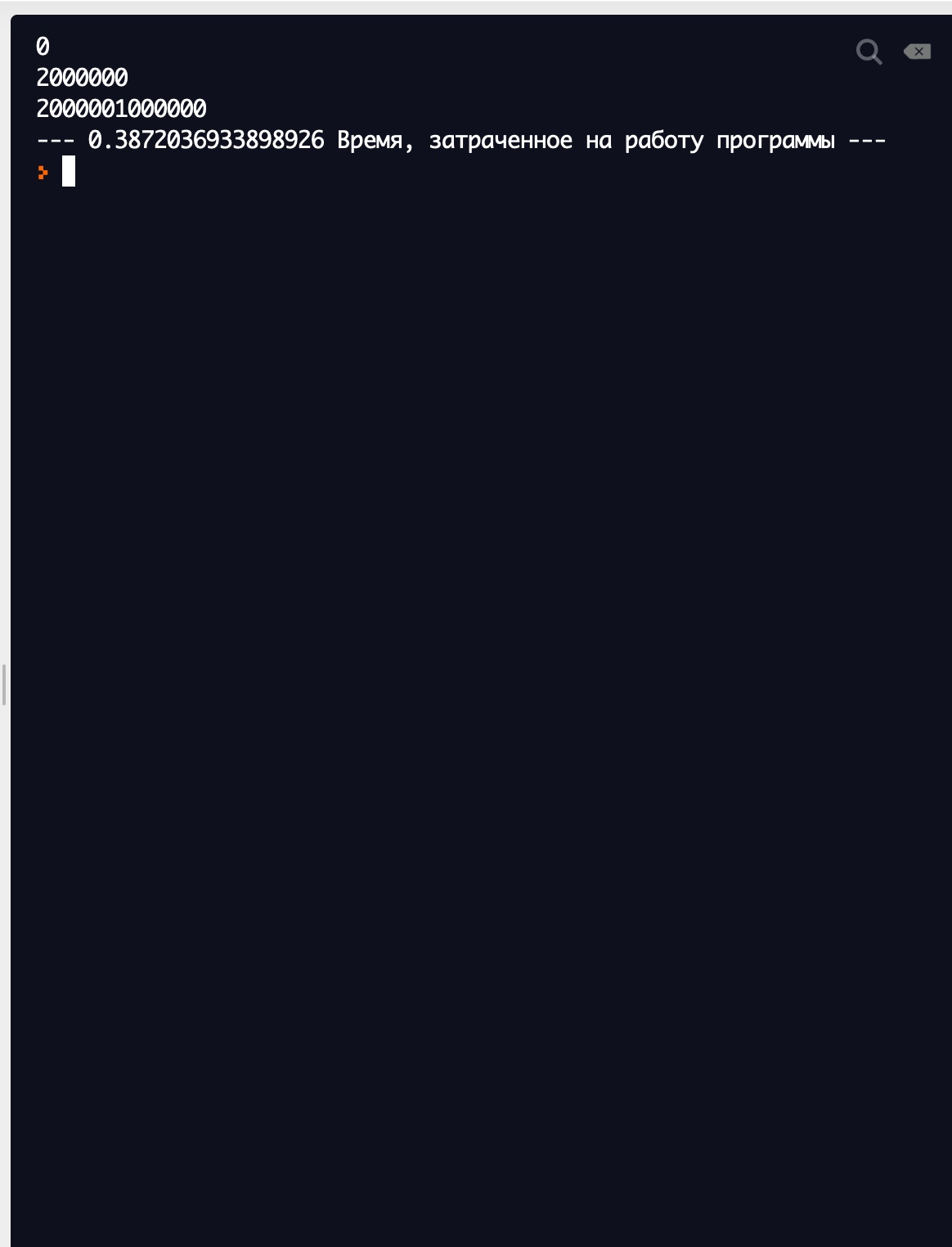
print(s)

start\_time = time.time()

sum(a,b)

print("--- %s Время, затраченное на работу программы ---" % (time.time() - start\_time))

**Вывод:**



**Время: 0.387 секунд**

# 

# **Программа №3**

Алгоритм Евклида.

**Console line reading С#.**

**Код:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Diagnostics;

using System.Threading;

namespace ConsoleApp38

{

class Program

{

static void Main(string[] args)

{

DateTime a = DateTime.Now;

{

int m, n, nod;

m = Convert.ToInt32(Console.ReadLine());

n = Convert.ToInt32(Console.ReadLine());

while (m != n)

{

if (m > n)

{

m = m - n;

}

else

{

n = n - m;

}

Console.WriteLine(DateTime.Now - a);

}

nod = n;

Console.WriteLine("НОД: " + nod);

}

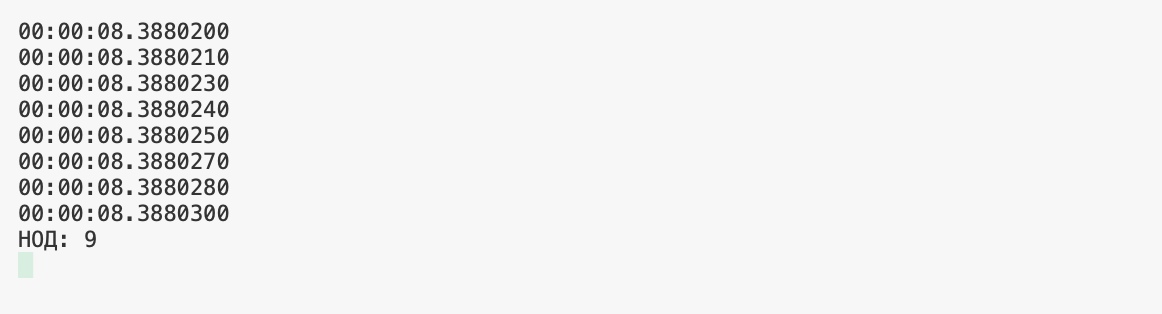
Console.ReadKey();

}

}

}

**Вывод**



**Время: 8.39 секунд**.

**Проверка времени на языке Python.**

**Код:**

import time

a=int(input())

b=int(input())

def nod(a, b):

while a > 0 and b > 0:

if a > b:

a=a-b

else:

b=b-a

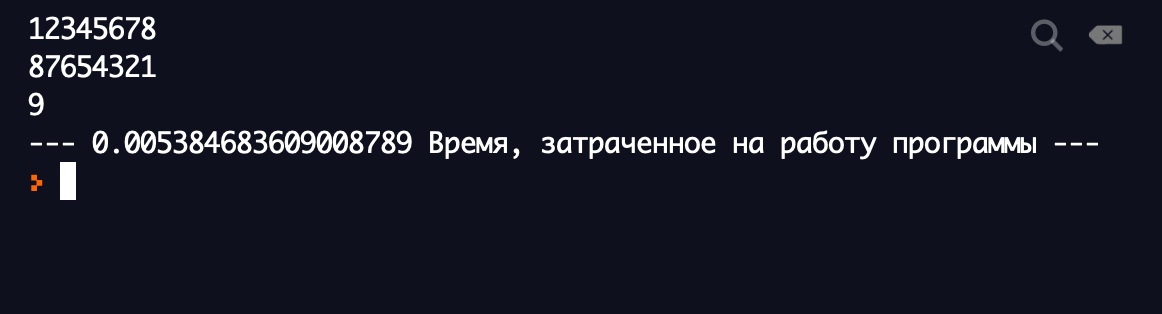
print(a)

start\_time = time.time()

nod(a,b)

print("--- %s Время, затраченное на работу программы ---" % (time.time() - start\_time))

**Вывод**



**Время: 0.005 секунд**

# **Программа №4**

Создание и перемножение двух матриц.

**Console line reading С#.**

**Код:**

using System;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

DateTime a = DateTime.Now;

{

Console.WriteLine("Введите размерность первой матрицы: ");

int[,] A = new int[Convert.ToInt32(Console.ReadLine()), Convert.ToInt32(Console.ReadLine())];

for (int i = 0; i < A.GetLength(0); i++)

{

for (int j = 0; j < A.GetLength(1); j++)

{

Console.Write("A[{0},{1}] = ", i, j);

A[i, j] = Convert.ToInt32(Console.ReadLine());

}

}

Console.WriteLine("Введите размерность второй матрицы: ");

int[,] B = new int[Convert.ToInt32(Console.ReadLine()), Convert.ToInt32(Console.ReadLine())];

for (int i = 0; i < B.GetLength(0); i++)

{

for (int j = 0; j < B.GetLength(1); j++)

{

Console.Write("B[{0},{1}] = ", i, j);

B[i, j] = Convert.ToInt32(Console.ReadLine());

}

}

Console.WriteLine("\nМатрица A:");

Print(A);

Console.WriteLine("\nМатрица B:");

Print(B);

Console.WriteLine("\nМатрица C = A \* B:");

int[,] C = Multiplication(A, B);

Print(C);

Console.WriteLine(DateTime.Now - a);

}

}

static int[,] Multiplication(int[,] a, int[,] b)

{

if (a.GetLength(1) != b.GetLength(0)) throw new Exception("Матрицы нельзя перемножить");

int[,] r = new int[a.GetLength(0), b.GetLength(1)];

for (int i = 0; i < a.GetLength(0); i++)

{

for (int j = 0; j < b.GetLength(1); j++)

{

for (int k = 0; k < b.GetLength(0); k++)

{

r[i, j] += a[i, k] \* b[k, j];

}

}

}

return r;

}

static void Print(int[,] a)

{

for (int i = 0; i < a.GetLength(0); i++)

{

for (int j = 0; j < a.GetLength(1); j++)

{

Console.Write("{0} ", a[i, j]);

}

Console.WriteLine();

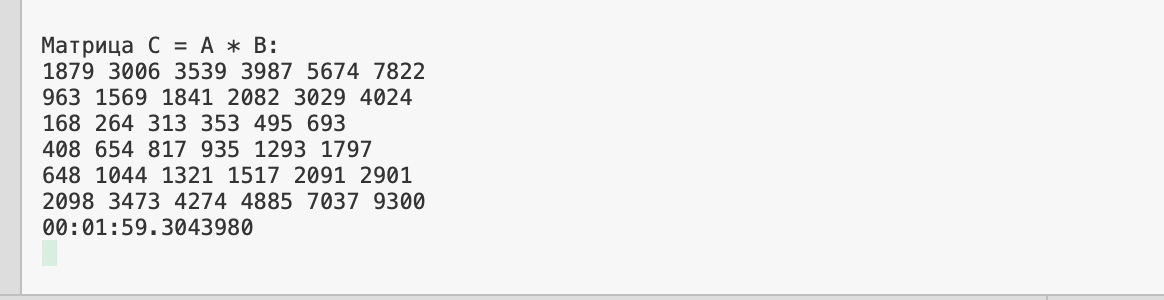
}

}

}

}

**Вывод:**



**Время: 119 секунд**

**Проверка времени на языке Python.**

import time

start\_time = time.time()

width = int(input())

height = int(input())

a = []

b = []

for i in range(width):

a.append([])

for j in range(height):

k=int(input())

a[i].append(k)

print(a)

for i in range(width):

for j in range(height):

print(a[i][j], end=' ')

print()

width1 = int(input())

height1 = int(input())

c = []

d = []

for i in range(width1):

c.append([])

for j in range(height1):

k=int(input())

c[i].append(k)

print(c)

if width == width1 and height1 == height:

for i in range(width):

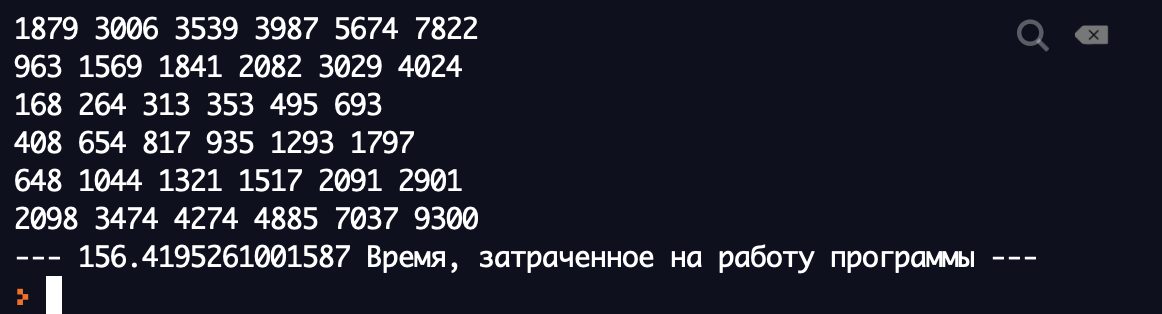
for j in range(height):

a[i][j]=a[i][j]\*c[i][j]

print(a)

print("--- %s Время, затраченное на работу программы ---" % (time.time() - start\_time))

**Вывод:**



**Время: 156.42 секунды**

# **Вывод**

Итоги работы программ по времени.

|  |  |  |
| --- | --- | --- |
| №/Язык | Console line reading С# | Python |
| 1 | 0.072 | 0.0000782 |
| 2 | 0.054 | 0.387 |
| 3 | 8.39 | 0.005 |
| 4 | 119 | 156 |

Если посмотреть на нашу таблицу, можно сделать вывод, что в определенных ситуациях языки работают по разному, в каком-то месте быстрее может сработать один язык, в каком-то месте другом. Но с учетом того, что я провел много испытаний, я считаю, что Console line reading С# быстрее. Но так как у С# есть сборщик ненужного материала, он может быть иногда может реализовывать дольше. Но два языка сравнивать на маленьких значениях не верно, нужно брать числа намного больше. Но данное исследование показывает, что Python при небольших значениях выдает результат быстрее, чем С#. Но если уже рассматривать большие значения, то Python уже не успевает за С#.