

# Projekt z Języków Symbolicznych

Nr.Projektu 11 [Opis zadania](#)

Dokumentacja

## TEMAT PROJEKTU

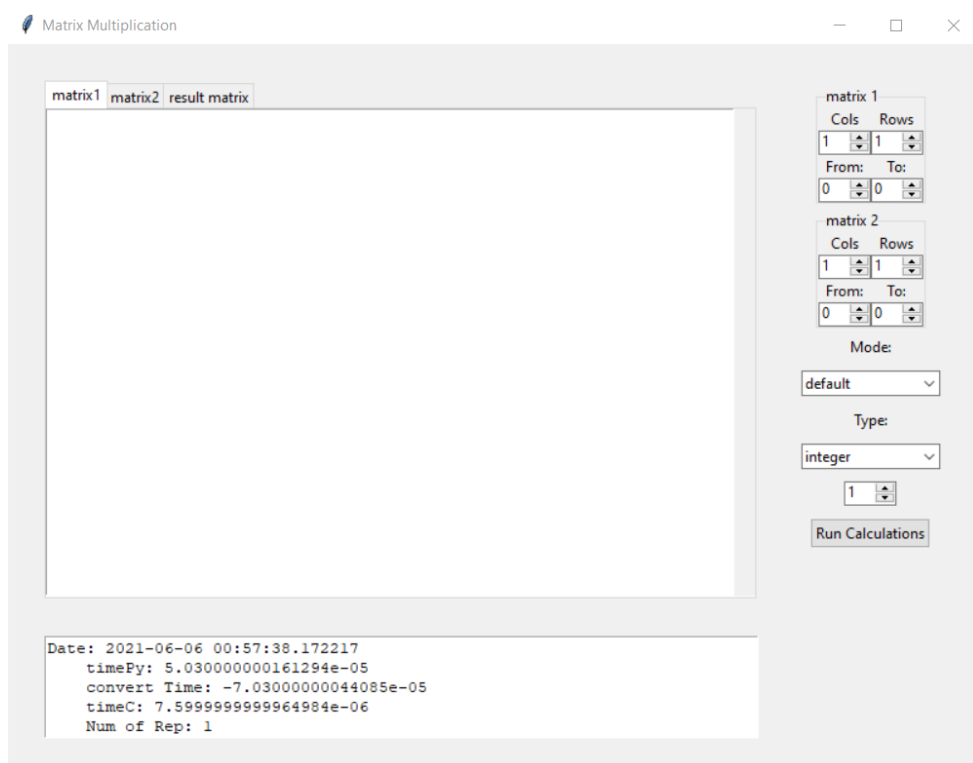
Projekt implementuje mnożenie dwóch macierzy za pomocą Python, tkinter oraz C/C++. Mnożenie realizowane jest w oparciu o podstawową matematyczną regułę mnożenia macierzy ([Link](#)). Macierze do programu. Macierze w programie albo generowane są na podstawie wprowadzonych numerów wierszy, kolumn i zakresu liczb, albo przy mnożeniu małych macierzy do 3x3 można wpisywać liczby ręcznie.

## FUNKCJONALNOŚĆ

1. Tryb (default, hand)

W zależności od trybu główne okna wyglądają w następujący sposób:

-Tryb zwykły (default)



-Tryb ręczny(hand)

Matrix Multiplication

Matrix1

0	0	0
0	0	0
0	0	0

Matrix2

0	0	0
0	0	0
0	0	0

result

matrix 1

Cols	Rows
3	3

matrix 2

Cols	Rows
3	3

Mode:

hand

Type:

integer

1

Run Calculations

Date: 2021-06-06 00:57:38.172217  
timePy: 5.030000000161294e-05  
convert Time: -7.03000000044085e-05  
timeC: 7.599999999964984e-06  
Num of Rep: 1

Okno główne składa się z trzech ramek: ramka z macierzami, ramka z konfiguracją i ramka z logami. **W trybie zwykłym** w ramce z macierzami mamy trzy notatniki do których są wypisywane dwie generowane macierzy i trzecia wynikowa, w ramce z konfiguracją mamy pola do wprowadzenia liczb wierszów i kolumn, oraz zakresu dla obu macierzy. **W trybie ręcznym** w ramce z macierzami mamy jeszcze trzy ramki dla każdej macierzy, w pierwszych dwóch są pola do wprowadzenia zawartości macierzy a w trzeciej jest pole tekstowe dla wyprowadzenia macierzy wynikowej. Ramka z konfiguracją też podlega zmianie, znikają pola do wprowadzenia zakresu. **Zawsze dla dwóch trybów** istnieją SelectBoxy do wyboru trybu, typu danych, liczby powtórzeń oraz przycisk z uruchomieniem kalkulacji (Run Calculations).

## 2. Typ (Integer, Float)

W ramce konfiguracji znajduje się SelectBox do wyboru typu danych i działa on różnie w zależności od trybu. Jeżeli korzystamy z trybu zwykłego to wtedy macierzy generowane są liczbami typu Integer lub Float. Ale jeżeli korzystamy z trybu ręcznego to zachodzą ograniczenia dla pól tekstowych takie że dla typu integer możemy wprowadzić tylko liczby 0-9 oraz znaki '+' lub '-', a dla typu float znaki 0-9 oraz '+', '-', '.', '.'.

### 3. Liczba powtórzeń

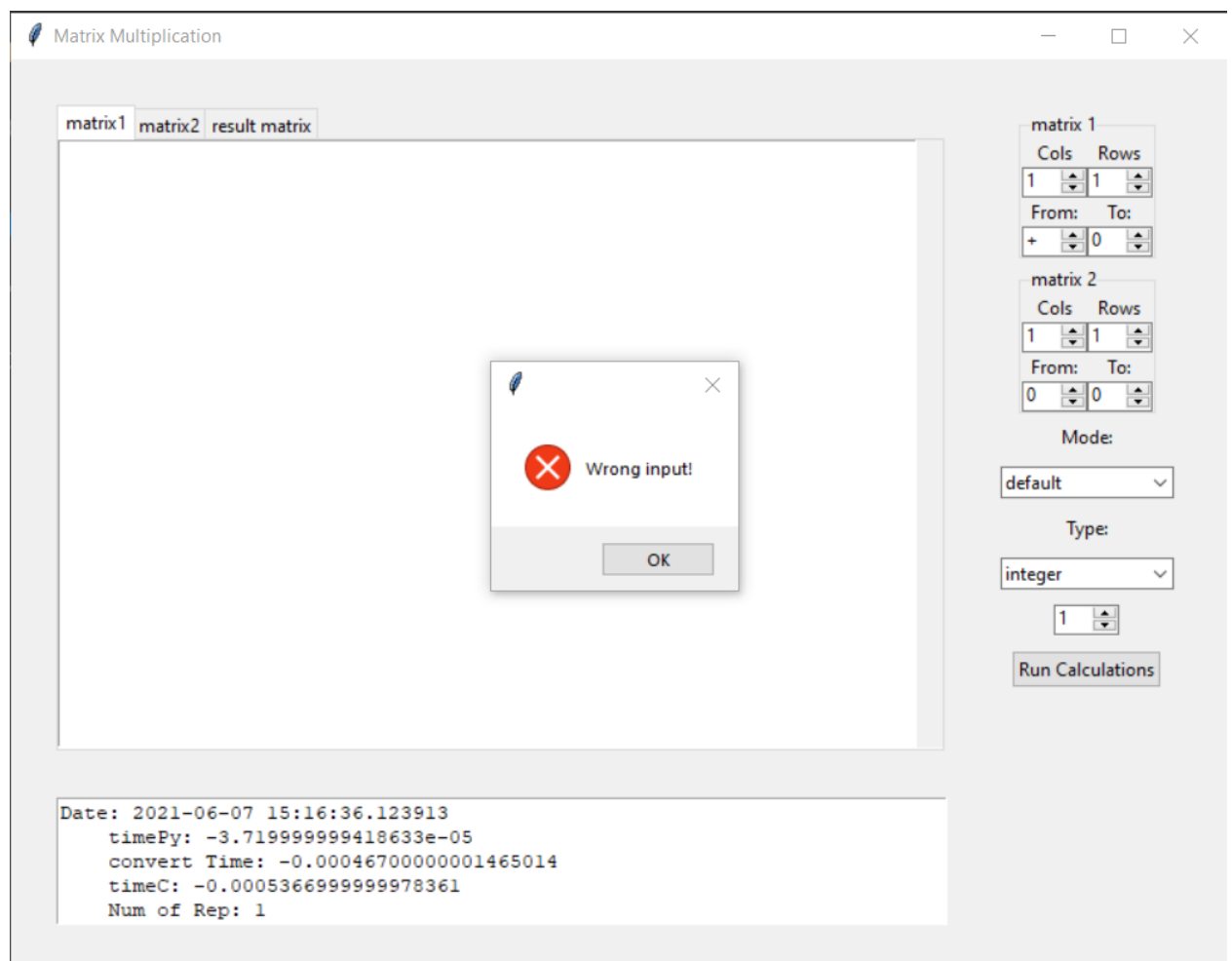
Jest to SpinBox do wprowadzenia liczby która odpowiada za to ile razy macierzy się przemnożą. Ograniczenia na ten obiekt jest takie same jak i dla zwykłego pola dla trybu Integer,znaczy że możemy wprowadzać znaki 0-9 oraz znaki '+' lub '-'.

### 4. Uruchom obliczenia

Przycisk uruchamia obliczenia dla wprowadzonych danych i wyprowadza wyniki i logi.

### 5. Błędy

**W trybie zwykłym** jeżeli wprowadzimy jakieś niepoprawne dane( '+' , '-' , '0' – dla pól do wyporu wierszy,kolumn) do wymienionych SpinBox'ów do dostaniemy komunikat 'Wrong input' w postaci okna dialogowego, ale możemy dalej korzystać z programu,po prostu musimy wprowadzić poprawne dane



**Wyjątkiem** jest tylko pole które odpowiada za liczbę powtórzeń, kiedy podamy tam + - albo zero to wszystko jedno domyślnie ustawia się 1 do tego pola.

**W trybie ręcznym** zachodzą ograniczenia na SpinBoxy i nie możemy tam wprowadzić danych, ale korzystając ze strzałek zmieniamy zawartość SpinBoxu w przedziale od 1 do 3. Wprowadzając niepoprawne dane do pól tekstowych do ustawienia zawartości macierzy mamy traktowane te dane jako 0, czyli wprowadzając + do jakiegoś pola z Macierzy 1 na przykład, program traktuje to miejsce jako zero i program działa.



## KLASY,SAMODZIELNE FUNKCJE I MODUŁY ZAWARTE W PROJEKCIE

### Moduł [GUI](#) [link](#)

#### Klasa [GUI](#) [link](#)

Zawiera wszystkie elementy głównego okna oraz posiada funkcje do tworzenia elementów ramki z macierzami,ramki z konfiguracją,ramki z logami oraz funkcje sprawdzające wprowadzone dane do pól,również jak i funkcje które zamieniają elementy przy wybraniu trybu.

#### funkcja [init](#) [link](#)

Definiuje podstawowe zmienne do tworzenia okna i podstawowe Framy do których będą umieszczane elementy,również zmienne dla sprawdzenia pól tekstowych które korzystają z Regular Expressions.Wywołuje funkcje które tworzą wszystkie niezbędne elementy.

#### funkcja [checkNum](#) [link](#)

Zwraca Regular Expression dla liczb całkowitych.

#### funkcja [checkNum2](#) [link](#)

Zwraca Regular Expression dla liczb rzeczywistych.

#### funkcja [createMatrixesFrameElems](#) [link](#)

Definiuję elementy dla ramki macierzy dla trybu zwykłego i trybu ręcznego,również ustala wartości początkowe dla pól tekstowych i nakłada na nich ograniczenia w postaci Regular Expressions.

#### funkcja [createLogsFrameElems](#) [link](#)

Definiuję pole tekstowe dla logów w ramce Logów.

#### funkcja [createConfigFrameElems](#) [link](#)

Definiuję elementy dla ramki konfiguracji dla trybu zwykłego i trybu ręcznego,również ustala wartości początkowe dla pól tekstowych i nakłada na nich ograniczenia w postaci Regular Expressions. Łączy SelectBoxy do wyboru trybu oraz typu danych z funkcjami odpowiadającymi za zmianę trybu(changeMode) i zmianę typu(changeType).

#### funkcja [changeType](#) [link](#)

Sprawdza zawartość SelectBoxu odpowiedzialnego za typy danych i na jego podstawie ustala ograniczenia dla SpinBoxów do wprowadzenia macierzy.

### funkcja `changeMode` [link](#)

Sprawdza zawartość SelectBoxu odpowiedzialnego za tryb i na jego podstawie usuwa elementy ramki dla Macierzy oraz ramki dla konfiguracji.

### funkcja `changeSpin` [link](#)

Funkcja tworzy zamienia stare SpinBoxy na nowe kiedy jest taka potrzeba (na przykład kiedy liczba wierszy musi się zmienić).

### funkcja `styles` [link](#)

Funkcja dodaje style do wybranych elementów.

### funkcja `render` [link](#)

Odpowiada za rawidłowe rozmieszczenie i wyświetlenie wszystkich elementów, również zawiera główną pętlę mainloop która uruchamia okno.

## Moduł `matrixes` [link](#)

### Klasa `MnozenieMacierzy` [link](#)

Jest to klasa abstrakcyjna albo interface który zawiera definicje trzech metod `init`, `printMatrix` i statycznej metody `mnoz`.

### Klasa `MnozenieMacierzyPython` [link](#)

Jest to klasa która dziedziczy po klasie `MnozenieMacierzy` i implementuje metody `init`, `printMatrix`, `mnoz`, oraz wykorzystuje standardową strukturę `list` do przechowywania macierzy i metody umożliwiające odczyt danych z gui.

### funkcja `init` [link](#)

Konstruktor klasy, przyjmuje dwie wartości (liczba wierszy i liczba kolumn) definiuje postać domyślną macierzy.

### funkcja `printMatrix` [link](#)

Służy do wyprowadzenia macierzy na ekran.

### funkcja `readFromEntries` [link](#)

Odpowiedzialna za odczyt danych z pól tekstowych w trybie ręcznym działania programu. Drugi argument jest to lista w której przechowywane wartości z pól tekstowych, trzeci argument funkcji odpowiada za rzutowanie danych na inny typ (int lub float).

### funkcja `fillMatrix` [link](#)

Funkcja uzupełnia macierz liczbami losowymi w przedziale który jest podawany w postaci drugiego i trzeciego argumentu. Trzeci argument jest to zmienna boolowska która przyjmuje wartość True jeżeli macierz jest macierzą liczb całkowitych i False jeżeli jest to macierz liczb rzeczywistych.

### funkcja `ifMultipliable` [link](#)

Sprawdza czy macierz na której funkcja jest wywoływana i macierz podana jako drugi argument mogą być przemnożone i zwraca True jeśli tak albo rzuca wyjątek z wiadomością „Wrong dimentions!” który jest obsługiwany w pliku głównym programu.

### funkcja `mnoz` [link](#)

Mnoży dwie macierze danej klasy podanych jako argument i zwraca macierz wynikową.

### funkcja `summ` [link](#)

Pomocnicza funkcja która zwraca sumę pomnożonych wybranego wiersza jednej macierzy na wybraną kolumnę innej macierzy.

## Klasa `MnozenieMacierzyCpp` [link](#)

Jest to klasa która dziedziczy po klasie `MnozenieMacierzy` i implementuje metody `init`, `printMatrix`, `mnoz`, oraz wykorzystuje standardową strukturę list do przechowywania macierzy i metody umożliwiające obliczenia w C/C++.

### funkcja `init` [link](#)

Konstruktor klasy, przyjmuje dwie wartości (liczba wierszy i liczba kolumn) definiuje postać domyślną macierzy.

### funkcja `takeMatrixes` [link](#)

Funkcja pobiera dane z innej macierzy która podawana jako drugi argument i przekształca ją na postać wskaźników C/C++ wykorzystując funkcje pomocnicze (`float_list`, `int_list`) zawarte w pliku. Trzeci argument jest zmienną boolowską która decyduje jaka funkcja będzie stosowana do przekształcenia (`int_list` albo `float_list`).

### funkcja `int_list` [link](#)

Pomocnicza funkcja która zwraca listę przesłaną jako argument w postaci wskaźnika/wskaźników `c_int`.

### **funkcja** `float_list` [link](#)

Pomocnicza funkcja która zwraca listę przesłaną jako argument w postaci wskaźnika/wskaźników `c_float`.

### **funkcja** `convert` [link](#)

Funkcja jest podobna do funkcji `takeMatrixes`, natomiast przekształca macierz zawartą w obiekcie danej klasy.

### **funkcja** `printMatrix` [link](#)

Służy do wyprowadzenia macierzy na ekran.

### **funkcja** `areEqual` [link](#)

Porównuje zawartości dwóch macierzy, przesyłanych jako argumenty i zwraca `True` jeżeli macierzy są takie same albo `False` w przeciwnym wypadku

### **funkcja** `mnoz` [link](#)

Przesyła macierze do dll biblioteki gdzie funkcje z języka `C/C++` mnożą dwie macierzy i wynik jest trzeciej macierzy, przesyłanej jako argument.

### **Moduł** `dlls` [link](#)

Zawiera plik źródłowy w języku `C/C++` bibliotekę dll utworzoną z tego pliku i pliku pośredniego z rozszerzeniem `.o`

### **Plik** `start.c` [link](#)

### **funkcja** `float_mnoz` [link](#)

Funkcja mnoży dwie macierzy przesyłane jako argumenty i wynik zapisuje w trzecią macierz która też jest przesyłana jako argument. Korzysta z funkcji pomocniczej `float_summa`.

### **funkcja** `float_summa` [link](#)

Pomocnicza funkcja która zwraca sumę pomnożonych wybranego wiersza jednej macierzy na wybraną kolumnę innej macierzy.

### **funkcje** `int_mnoz` i `int_summa` [link](#)

Działają tak samo jak i `float_mnoz` i `int_mnoz` ale wykorzystują liczby całkowite.

## **START PROGRAMU**



## Plik `main.py` [link](#)

Zawiera główną logikę programu i łączy wszystkie moduły. Po utworzeniu obiektu GUI odczytuje logi które znajdują się (src/logi/log.txt) o ile istnieją. Wiąże przycisk z funkcją Run i rysuje wszystkie elementy w oknie.

## funkcje `Run` [link](#)

Implementuje główną logikę programu.

- 1.Odczytuje liczbę powtórzeń
- 2.Odczytuje tryb
- 3.Tworzy macierze
- 4.Odczytuje typ danych
- 5.Odczytuje dane z pól tekstowych lub generuje dane i uzupełnia macierze
- 6.Sprawdza czy te macierze da się pomnożyć,jeśli nie to komunikat o błędzie z możliwością kontynuacji działania
- 7.Mierzy czas przekształcenia daych do C/C++
- 8.Mierzy czas obliczeń w C/C++
- 9.Mierzy czas obliczeń w Python
- 10.Wypisuje macierze
- 11.Zapisuje logi

## funkcje `printMatrix` [link](#)

Wypisuje macierz podaną jako drugi argument do Fram'u podanego jako pierwszy argument.

## funkcje `printToLogFrame`

Wypisuje log do LogFram'u oraz tworzy/zapisuje do logu otrzymane czasy obliczeń.

## Plik `testMatrixes.py` [link](#)

## Klasa `TestMatrixes` [link](#)

Zawiera testy jednostkowe do programu.

## funkcja `test_shouldReturnCorrectAnswerForPython_1` [link](#)

oraz

## funkcja `test_shouldReturnCorrectAnswerForC_1` [link](#)

Sprawdza działanie dla następnych danych:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$$

**funkcja** `test_shouldReturnCorrectAnswerForPython_2` [link](#)

oraz

**funkcja** `test_shouldReturnCorrectAnswerForC_2` [link](#)

Sprawdza działanie dla następnych danych:

$$\begin{bmatrix} 1 & 2 & 1 \\ 4 & -2 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ 2 & 0 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ -4 & 4 \end{bmatrix}$$

**funkcja** `test_shouldReturnCorrectAnswerForPython_3` [link](#)

oraz

**funkcja** `test_shouldReturnCorrectAnswerForC_3` [link](#)

Sprawdza działanie dla następnych danych:

$$\begin{bmatrix} 1 & 2 \\ 4 & -2 \\ 0 & 3 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & -3 \\ 2 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 4 & -1 & -3 \\ -4 & 6 & -12 \\ 6 & -3 & 0 \end{bmatrix}$$

**funkcja** `test_shouldRaiseException_WhenWrongDimentions` [link](#)

Sprawdza czy program rzuca wyjątek dla nieprawidłowych wymiarów

## TESTOWANIE

1. Wykonanie mnożenia (w Pythonie i przez dll):

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$$

2. Wykonanie mnożenia (w Pythonie i przez dll):

$$\begin{bmatrix} 1 & 2 & 1 \\ 4 & -2 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ 2 & 0 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ -4 & 4 \end{bmatrix}$$

3. Wykonanie mnożenia (w Pythonie i przez dll):

$$\begin{bmatrix} 1 & 2 \\ 4 & -2 \\ 0 & 3 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & -3 \\ 2 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 4 & -1 & -3 \\ -4 & 6 & -12 \\ 6 & -3 & 0 \end{bmatrix}$$

- 
4. Próba wykonania mnożenia macierzy 2x3 przez macierz 1x2 (oczekiwany komunikat o błędzie i możliwość kontynuowania pracy z programem bez ponownego uruchamiania).
  5. Uzyskanie porównania czasów dla mnożenia macierzy 1x1, 10x10 i 50x50 (odpowiednio 100000, 1000 i 50 powtórzeń, liczby całkowite).
  6. Uzyskanie porównania czasów dla mnożenia macierzy 200x200 (3 powtórzeń, liczby rzeczywiste).

Dla zadań z 1-4 testy przechodzą:

```
ication_SymbolicLanguages/src/testMatrixes.py
```

```
.....
```

```
-----  
Ran 7 tests in 0.002s
```

```
OK
```

Natomiast do pozostałych zadań nie potrafiłem uzyskać porównanie czasów w postaci testów więc po prostu uruchomiłem program i przedstawiam wyniki w postaci logów

## Zadanie 5

Matrix Multiplication

matrix1	matrix2	result matrix
7581	7291	7149
6941	6679	6523
7291	7032	6802
7056	6936	6718
6934	6603	6490
6994	6797	6572
7601	7389	7096
6848	6702	6453
6960	6768	6629
6644	6359	6201
8149	7839	7682
6697	6509	6318
7251	6931	6731
6489	6378	6208
7251	7067	6784
6781	6531	6408
7236	6904	6810
7918	7479	7370
6606	6251	6100
7042	6748	6609
6564	6289	6090
6570	6382	6125
6561	6197	6092
8205	7810	7685
7275	6985	6855
7218	7018	7525
6492	6781	7039
6868	6640	7016
6795	6686	7079
6511	6357	6587
6484	6946	6974
7029	7317	7459
6309	6766	7085
6482	6796	6928
6326	6491	6767
7650	7992	8191
6176	6463	6671
6762	7120	7336
6108	6356	6689
6767	7146	7102
6381	6694	6918
6674	7138	7321
7178	7575	7903
6120	6479	6803
6615	6971	7041
6182	6473	6736
6046	6321	6551
6141	6412	6598
7608	8095	8364
6732	7374	7000
7525	7789	7377
6781	7039	6699
7016	7301	7028
6968	7079	6936
6587	6901	6711
6946	6974	6758
7459	7265	7342
7085	6593	6796
6928	6719	6752
6494	6494	6582
8191	7821	8085
6671	6549	6586
7336	6974	7111
6689	6408	6426
7102	7072	7062
6918	6557	6733
7321	6987	7070
7903	7677	7704
6803	6347	6548
7041	6830	6857
6736	6241	6434
6551	6283	6332
6598	6289	6520
8364	7921	8068
7374	7000	7169

Date: 2021-06-07 21:53:28.981559

timePy: 0.0010956000000419408

convert Time: 0.0013285000000056465

timeC: -0.00041129999999611755

Num of Rep: 50

matrix 1

Cols Rows

50 50

From: To:

1 10

matrix 2

Cols Rows

50 50

From: To:

20 30

Mode:

default

Type:

integer

50

Run Calculations

Matrix Multiplication

matrix1	matrix2	result matrix
1537	1578	1622
1434	1493	1409
1151	1205	1133
770	780	817
1611	1565	1618
1180	1216	1168
1229	1286	1281
1311	1260	1257
1964	2036	1989
1371	1339	1328
1504	1505	1614
1359	1391	1458
1123	1124	1211
737	723	799
1544	1619	1615
1172	1178	1204
1227	1324	1304
1234	1336	1284
1952	2095	2000
1305	1451	1369
1605	1766	1571
1472	1571	1508
1168	1308	1198
816	888	846
1615	1737	1685
1204	1326	1248
1304	1447	1384
1284	1383	1341
2231	2102	
1536	1459	

Date: 2021-06-07 21:52:20.986447

timePy: 0.00024090000002274792

convert Time: -0.00080639999999876614

timeC: -0.0007132999999726053

Num of Rep: 1000

matrix 1

Cols Rows

10 10

From: To:

1 10

matrix 2

Cols Rows

10 10

From: To:

20 30

Mode:

default

Type:

integer

1000

Run Calculations

Matrix1

Matrix2

result  

30

```
Date: 2021-06-07 22:04:25.935083
timePy: -0.0022506000000248605
convert Time: -0.0017449999999712418
timeC: -0.00363239999999012813
Num of Rep: 10000
```

matrix 1  
Cols Rows  

1

1

matrix 2  
Cols Rows  

1

1

Mode:  

hand

Type:  

integer

10000

Run Calculations

Muszę zauważyć że dla liczby powtórzeń równej 1000 wydawało się że program się zawiesił, ale po 2-3 minutach program wszystko obliczył dał wynik, dla 10 000 program wykonywał się więcej niż 20 minut, próbowałem uruchomić go dla 100 000 i czekałem około 1.5 godziny zanim zatrzymałem obliczanie. Jeśli prawidłowo zrozumiałem, żeby wykonać obliczenie dla 100 000 to trzeba czekać około 4 godzin. Domyślam się że to wynika z tego że użyłem pętli dla tego żeby powtarzać obliczenia. Może sytuację poprawiło by użycie wątków lub jakichś innych mechanizmów z jakimi jeszcze nie jestem zapoznany.

Zadanie 6.

matrix1	matrix2	result matrix
2451.0011878516907	2445.41917352696	2480.102961200624
2532.966369114078	2524.021652195789	2535.1355761139835
2499.666579587216	2522.2769023136443	2546.509203917936
2381.198142811924	2410.0525600533147	2402.7998400382153
2446.843063467266	2439.639281764446	2466.2508693667437
2574.917973713495	2592.2391416722267	2594.56673829603
2534.2409393772527	2564.774569655856	2570.3799492651797
2627.2493052198665	2627.2742019652324	2633.7834430106386
2621.193112990364	2640.5274168208425	2665.7308477404963
2635.0029199590017	2597.692707590557	2626.602586716452
2508.4416222112172	2508.172721430406	2507.565049860943
2638.8887611022424	2679.3390268638677	2673.2511844468477
2524.0239492565447	2517.7582222507112	2512.6655039467605
2386.8908387214465	2416.957605185927	2415.5426561404533
2389.8718834893593	2405.664300604025	2405.5295347024903
2714.615711466419	2735.9510926404437	2733.2558029360325
2558.538278567904	2575.182046144487	2569.888081539071
2443.2866314619614	2448.774166855578	2446.968716189082
2470.5078962868606	2481.7112458694037	2502.737343890833
2513.219244426304	2524.949833867815	2527.3494780801348
2439.488758438563	2436.0516439919907	2442.5240179107454
2523.765634266827	2533.160499293101	2532.046741351413
2446.7795494338807	2468.0298146306777	2490.080914963537
2652.992803397266	2670.6851073404378	2670.994848687187
2566.5126550299183	2593.6472050716407	2575.685207509379

Date: 2021-06-07 22:21:37.476293  
timePy: 0.0027774000000135857  
convert Time: 0.0006768999999167136  
timeC: -5.599999985861359e-05  
Num of Rep: 3

matrix 1  
Cols Rows  
200 200  
From: To:  
0 1

matrix 2  
Cols Rows  
200 200  
From: To:  
20 30

Mode:

default

Type:

float

3

Run Calculations