Morhaliuk Ivan GL.05

**Projekt z Języków Symbolicznych**

Nr.Projektu 11 [Opis zadania](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/main/README.md)

**Dokumentacja**

**TEMAT PROJEKTU**

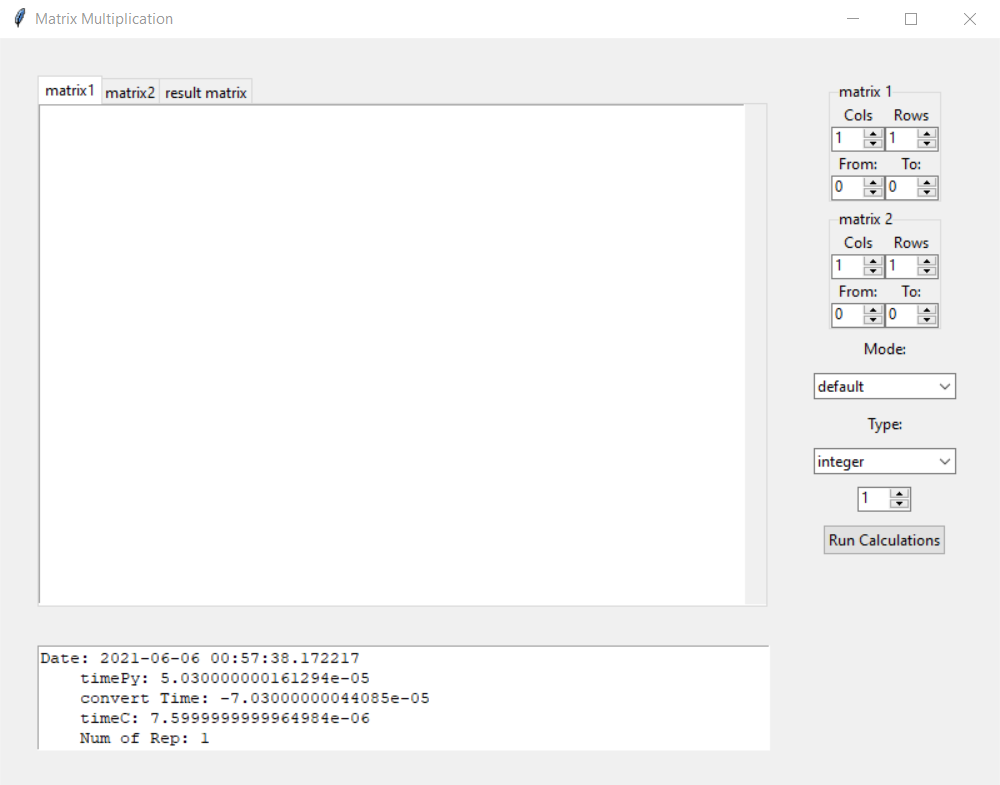
Projekt implementuje mnożenie dwóch macierzy za pomocą Python, tkinter oraz C/C++.Mnożenie realizowane jest w oparciu o podstawową matematyczną regułę mnożenia macierzy([Link](https://pl.wikipedia.org/wiki/Mnożenie_macierzy)). Macierze do programu. Macierze w programie albo generowane są na podstawie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

**FUNKCJONALNOŚĆ**

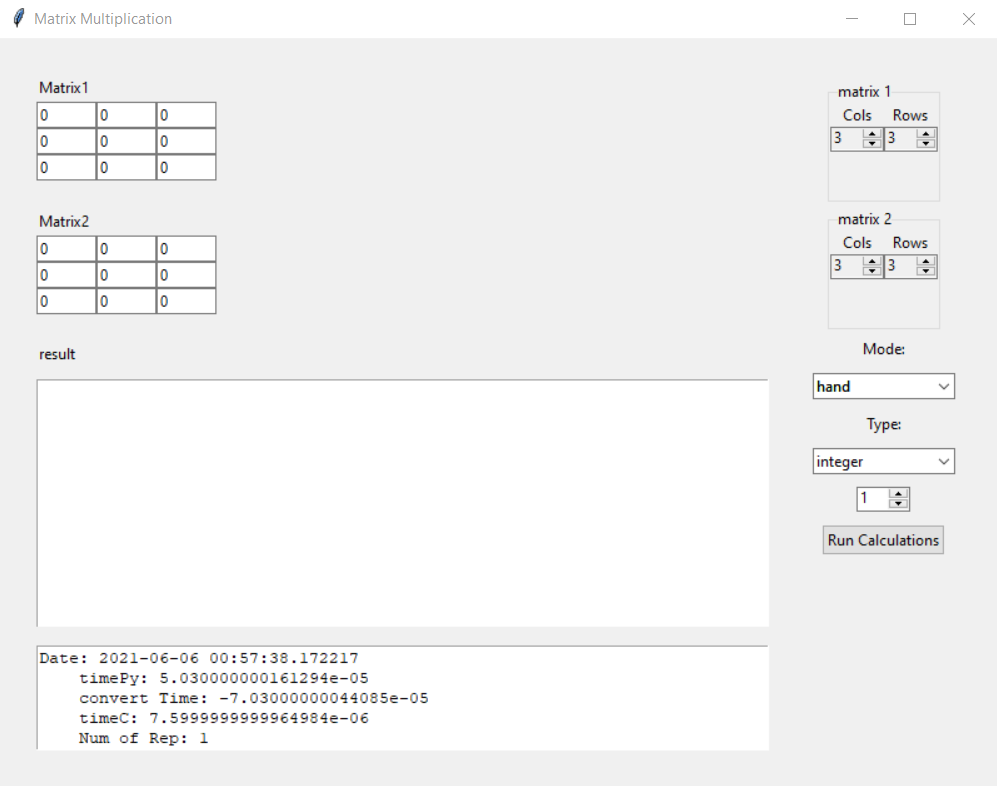
1. Tryb (default,hand)

W zależności od trybu główne okna wyglądają w następujący sposób:

-Tryb zwykły(default)

****

-Tryb ręczny(hand)



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 wierzów 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 wyprowadzenie 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 urochomieniem kalkulacji(Run Calculations).

1. Typ (Integer,Float)

W ramce konfiguracji znajduje się SelectBox do wyboru typu danych i działa on róznie 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 ‘ + ’ , ’ - ’ , ’ . ’ .

1. Liczba powtórzen

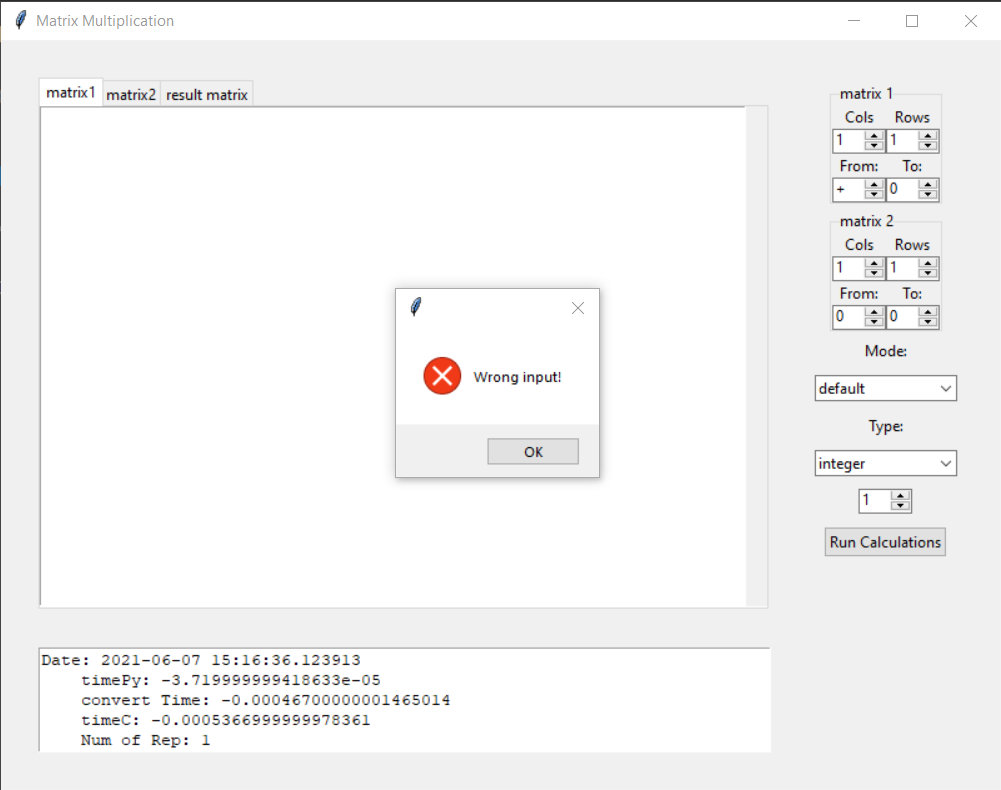
Jest to SpinBox do wprowadzenia liczby która odpowiada za to ile razy macierzy się przemnożą. Ograniczenia na ten objekt jest takie same jak i dla zwykłego pola dla trybu Integer,znaczy że możemy wprowadzać znaki 0-9 oraz znaki ‘+’ lub ‘–‘.

1. Uruchom obliczenia

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

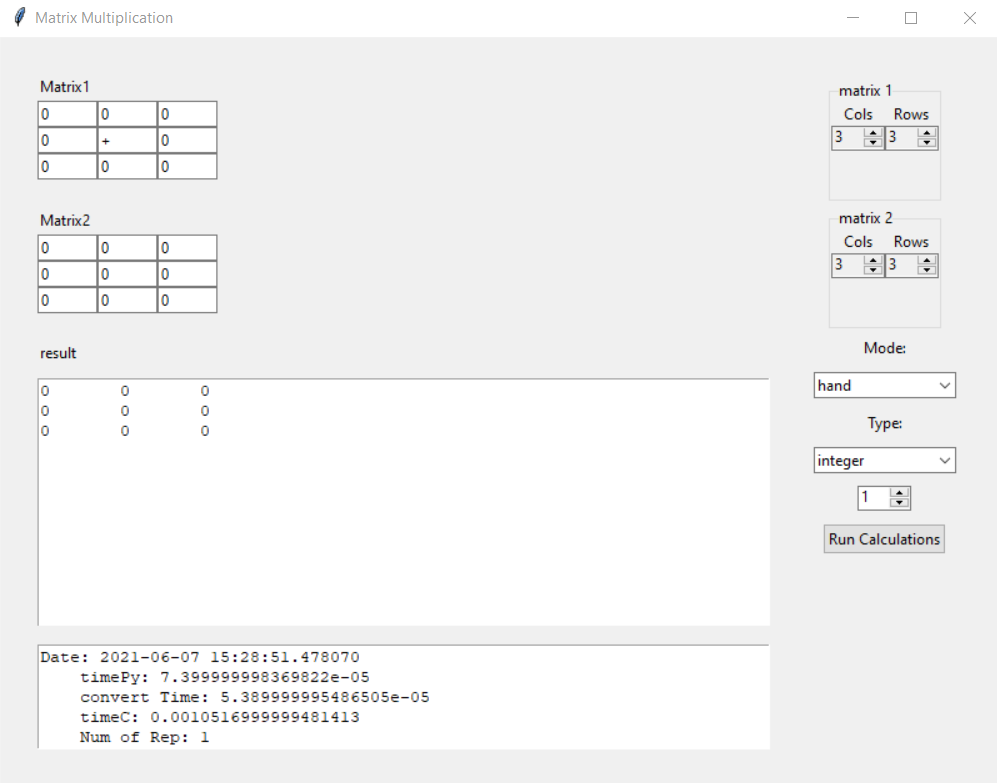
1. Błedy

**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

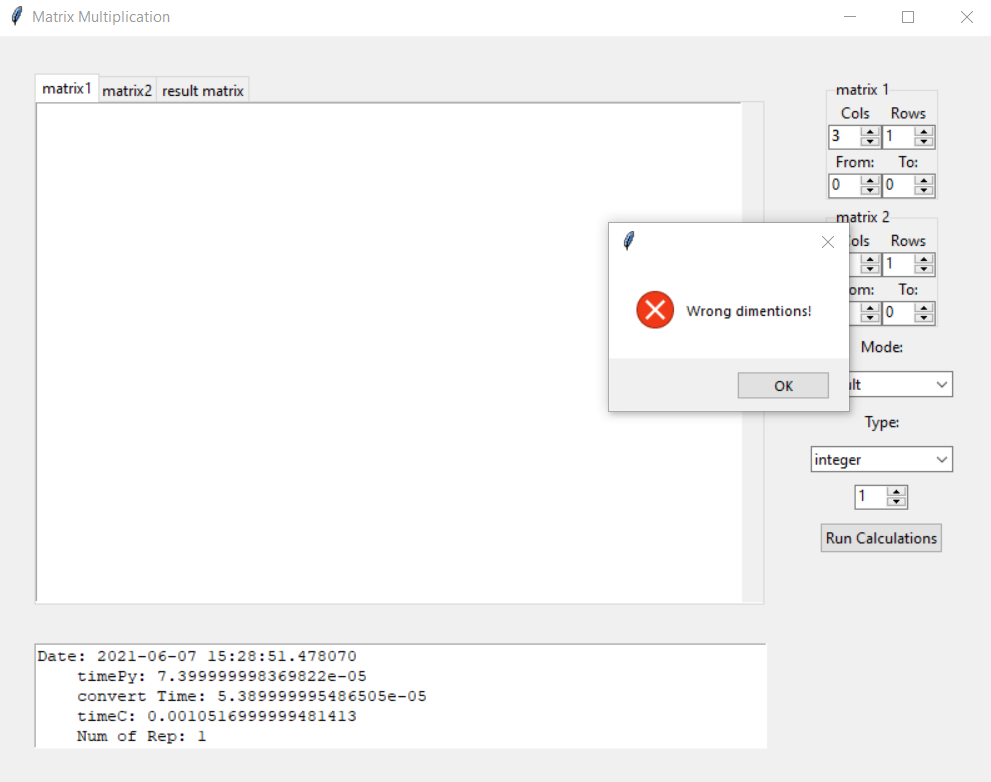


**Wyjątkiem** jest tylko pole które odpowiada za liczbę powtórzeń,kiedy podamy tam + - albo zero to wszystko jedno domyślnie ustawia się 1 do tego pola.

**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.

****

Również błąd wynika z powodu złych wymiarów (kiedy macierze nie da się przemnożyć).

****

**KLASY,SAMODZIELNE FUNKCJE I MODUŁY ZAWARTE W PROJEKCIE**

**Moduł GUI** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/tree/main/src/GUI)

**Klasa GUI [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/main/src/GUI/gui.py)**

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

**funkcja init [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/30852fa7c6984d7ee3d64e9d0cf53a4e0006db9e/src/GUI/gui.py" \l "L7)**

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

**funkcja checkNum** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/30852fa7c6984d7ee3d64e9d0cf53a4e0006db9e/src/GUI/gui.py#L45)

Zwraca Regular Expression dla liczb całkowitych.

**funkcja checkNum2** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/30852fa7c6984d7ee3d64e9d0cf53a4e0006db9e/src/GUI/gui.py#L48)

Zwraca Regular Expression dla liczb rzeczywistych.

**funkcja createMatrixesFrameElems** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/30852fa7c6984d7ee3d64e9d0cf53a4e0006db9e/src/GUI/gui.py#L114)

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**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/30852fa7c6984d7ee3d64e9d0cf53a4e0006db9e/src/GUI/gui.py#L161)

Definiuję pole tekstowe dla logów w ramce Logów.

**funkcja createConfigFrameElems** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/30852fa7c6984d7ee3d64e9d0cf53a4e0006db9e/src/GUI/gui.py#L164)

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. Lączy SelectBoxy do wyboru trybu oraz typu danych z funkcjami odpowiadającymi za zmiane trybu(changeMode) i zmiane typu(changeType).

**funkcja changeType** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/30852fa7c6984d7ee3d64e9d0cf53a4e0006db9e/src/GUI/gui.py#L51)

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

**funkcja changeMode** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/30852fa7c6984d7ee3d64e9d0cf53a4e0006db9e/src/GUI/gui.py#L61)

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

**funkcja changeSpin** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/30852fa7c6984d7ee3d64e9d0cf53a4e0006db9e/src/GUI/gui.py#L24)

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

**funkcja styles [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/30852fa7c6984d7ee3d64e9d0cf53a4e0006db9e/src/GUI/gui.py" \l "L215)**

Funkcja dodaje style do wybranych elementów.

**funkcja render** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/30852fa7c6984d7ee3d64e9d0cf53a4e0006db9e/src/GUI/gui.py#L220)

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

**Moduł matrixes** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/tree/main/src/matrixes)

**Klasa MnozenieMacierzy** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/main/src/matrixes/MnozenieMacierzy.py)

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

**Klasa MnozenieMacierzyPython** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/main/src/matrixes/MnozenieMacierzyPython.py)

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

**funkcja init [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/matrixes/MnozenieMacierzyPython.py" \l "L7)**

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

**funkcja printMatrix [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/matrixes/MnozenieMacierzyPython.py" \l "L13)**

Służy do wyprowadzenia macierzy na ekran.

**funkcja readFromEntries [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/matrixes/MnozenieMacierzyPython.py" \l "L19)**

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 func odpowiada za rzutowanie danych na inny typ(int lub float).

**funkcja fillMatrix [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/matrixes/MnozenieMacierzyPython.py" \l "L23)**

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 maicierzą liczb całkowitych i False jeżeli jest to macierz liczb rzeczywistych.

**funkcja ifMultipliable [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/matrixes/MnozenieMacierzyPython.py" \l "L32)**

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

**funkcja mnoz [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/matrixes/MnozenieMacierzyPython.py" \l "L44)**

Mnózy dwie macierzy danej klasy podanych jako argument i zwraca macierz wynikową.

**funkcja summ** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/matrixes/MnozenieMacierzyPython.py#L38)

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

**Klasa MnozenieMacierzyCpp** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/main/src/matrixes/MnozenieMacierzyCpp.py)

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

**funkcja init [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/matrixes/MnozenieMacierzyCpp.py" \l "L16)**

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

**funkcja takeMatrixes [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/matrixes/MnozenieMacierzyCpp.py" \l "L21)**

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 decyduja jaka funkcja będzie stosowana do przekształcenia (int\_list albo float\_list).

**funkcja int\_list [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/matrixes/MnozenieMacierzyCpp.py" \l "L9)**

Pomocnicza funkcja która zwraca listę przesłaną jako argument w postaci wskaźnika/wskaźników c\_int.

**funkcja float\_list** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/matrixes/MnozenieMacierzyCpp.py#L3)

Pomocnicza funkcja która zwraca listę przesłaną jako argument w postaci wskaźnika/wskaźników c\_float.

**funkcja convert [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/matrixes/MnozenieMacierzyCpp.py" \l "L27)**

Funkcja jest podobna do funkcji takeMatrixes,natomist przekształca macierz zawartą w objekcie danej klasy.

**funkcja printMatrix [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/matrixes/MnozenieMacierzyCpp.py" \l "L33)**

Służy do wyprowadzenia macierzy na ekran.

**funkcja areEqual [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/matrixes/MnozenieMacierzyCpp.py" \l "L39)**

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

**funkcja mnoz [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/matrixes/MnozenieMacierzyCpp.py" \l "L48)**

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](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/tree/main/src/dlls)**

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

**Plik start.c [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/main/src/dlls/start.c)**

**funkcja float\_mnoz [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/dlls/start.c" \l "L9)**

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**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/dlls/start.c#L1)

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

**funkcje int\_mnoz i int\_summa** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/dlls/start.c#L17)

Działają tak samo jak i float\_mnoz i int\_mnoz ale wykorzystują liczby całkowite.

**START PROGRAMU**

**Plik main.py [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/main/src/main.py)**

Zawiera głowną logikę programu i łączy wszystkie moduły. Po utworzeniu objektu 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](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/main.py" \l "L34)**

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 macierzy 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](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/main.py" \l "L23)**

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**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/main/src/testMatrixes.py)

**Klasa TestMatrixes [link](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/testMatrixes.py" \l "L5)**

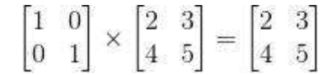
Zawiera testy jednostkowe do programu.

**funkcja test\_shouldReturnCorrectAnswerForPython\_1** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/testMatrixes.py#L7)

oraz

**funkcja test\_shouldReturnCorrectAnswerForC\_1** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/testMatrixes.py#L19)

Sprawdza działanie dla następnych danych:

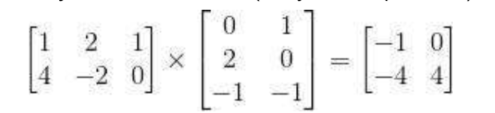


**funkcja test\_shouldReturnCorrectAnswerForPython\_2** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/testMatrixes.py#L37)

oraz

**funkcja test\_shouldReturnCorrectAnswerForC\_2** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/testMatrixes.py#L50)

Sprawdza działanie dla następnych danych:

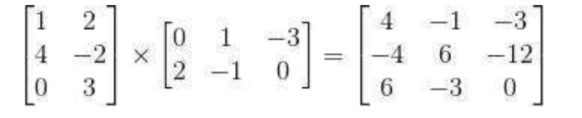


**funkcja test\_shouldReturnCorrectAnswerForPython\_3** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/testMatrixes.py#L68)

oraz

**funkcja test\_shouldReturnCorrectAnswerForC\_3** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/testMatrixes.py#L81)

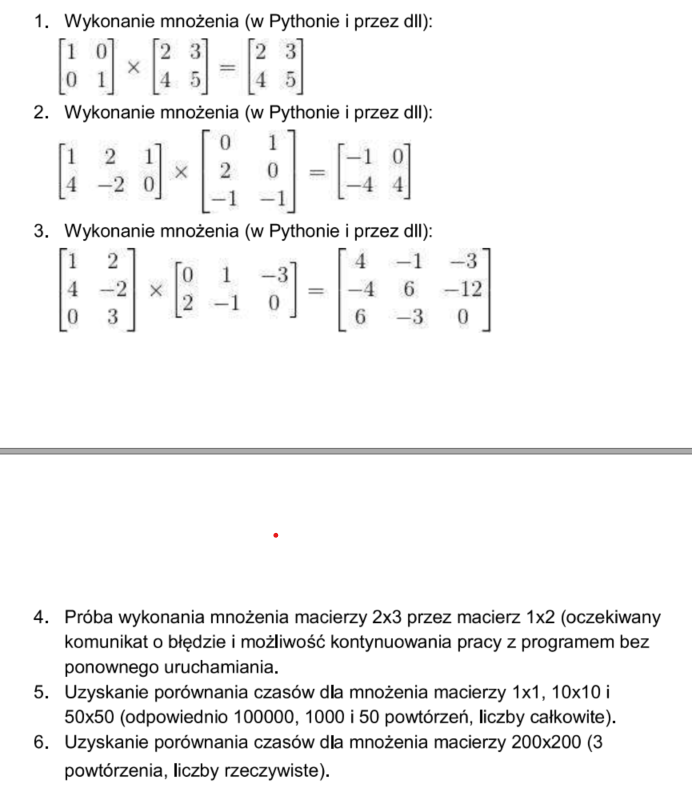
Sprawdza działanie dla następnych danych:



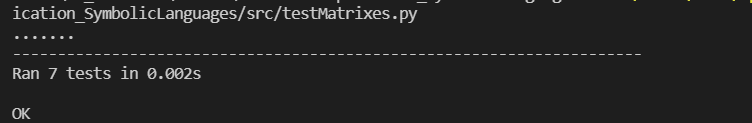
**funkcja test\_shouldRaiseException\_WhenWrongDimentions** [**link**](https://github.com/IvanMorhaliuk/MatrixMultiplication_SymbolicLanguages/blob/9f9bfcf436dd3d48078e5b39d02ce85a5e062064/src/testMatrixes.py#L99)

Sprawdza czy program rzuca wyjątek dla nieprawidłowych wymiarów

**TESTOWANIE**

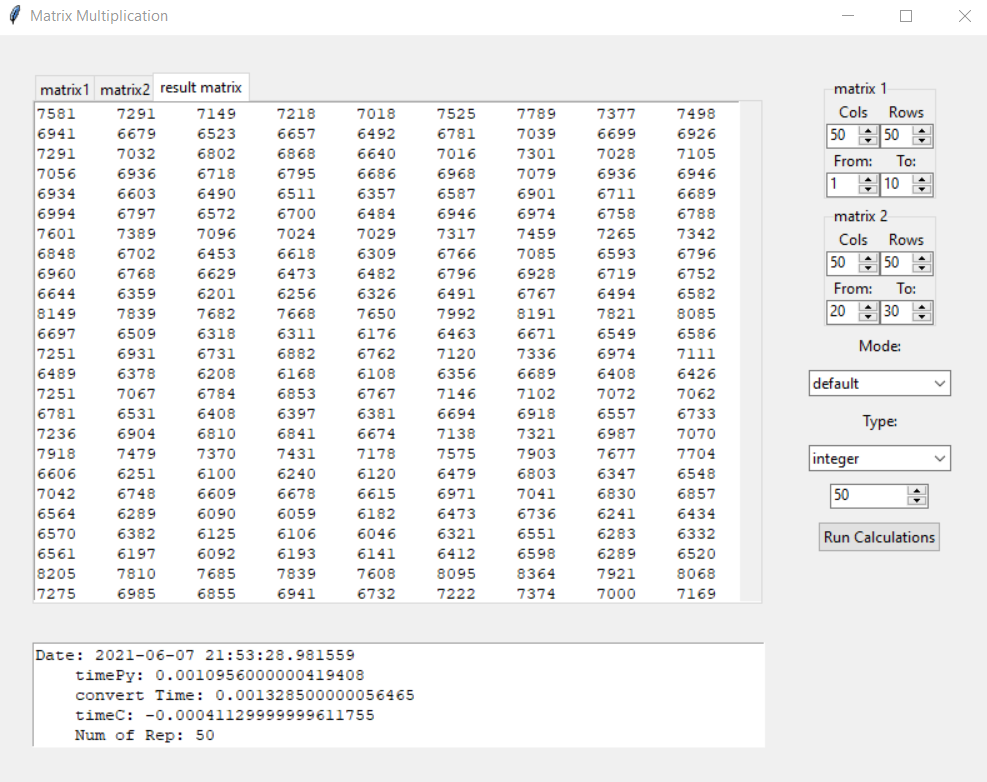


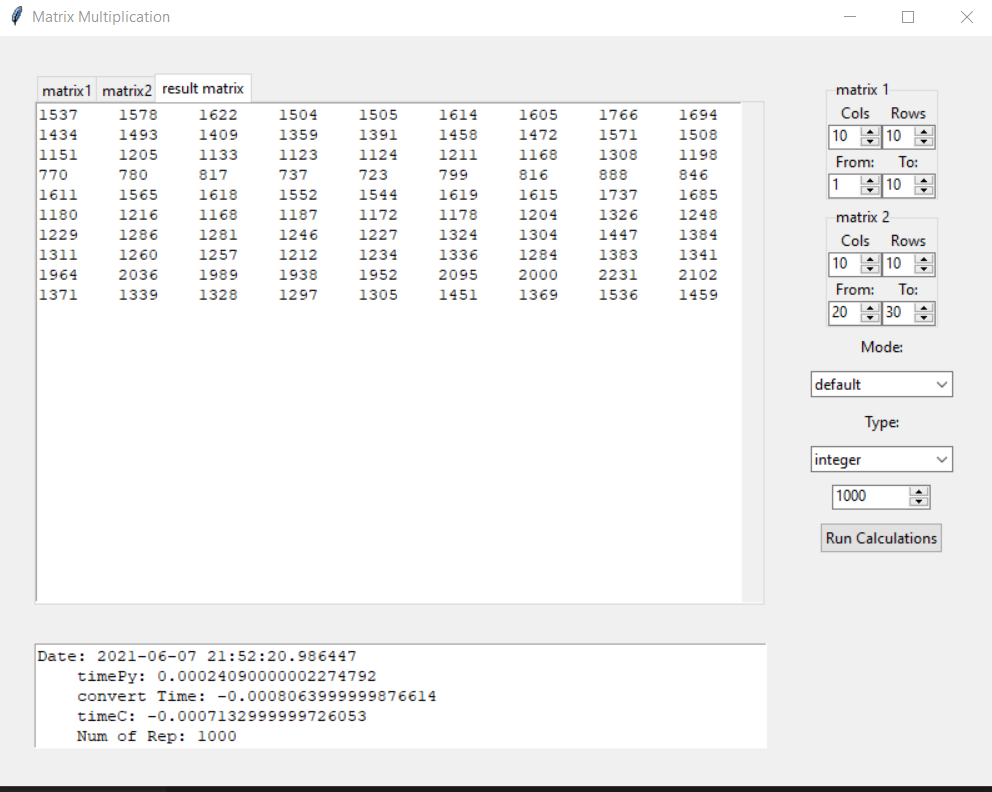
Dla zadań z 1-4 testy przechodzą:

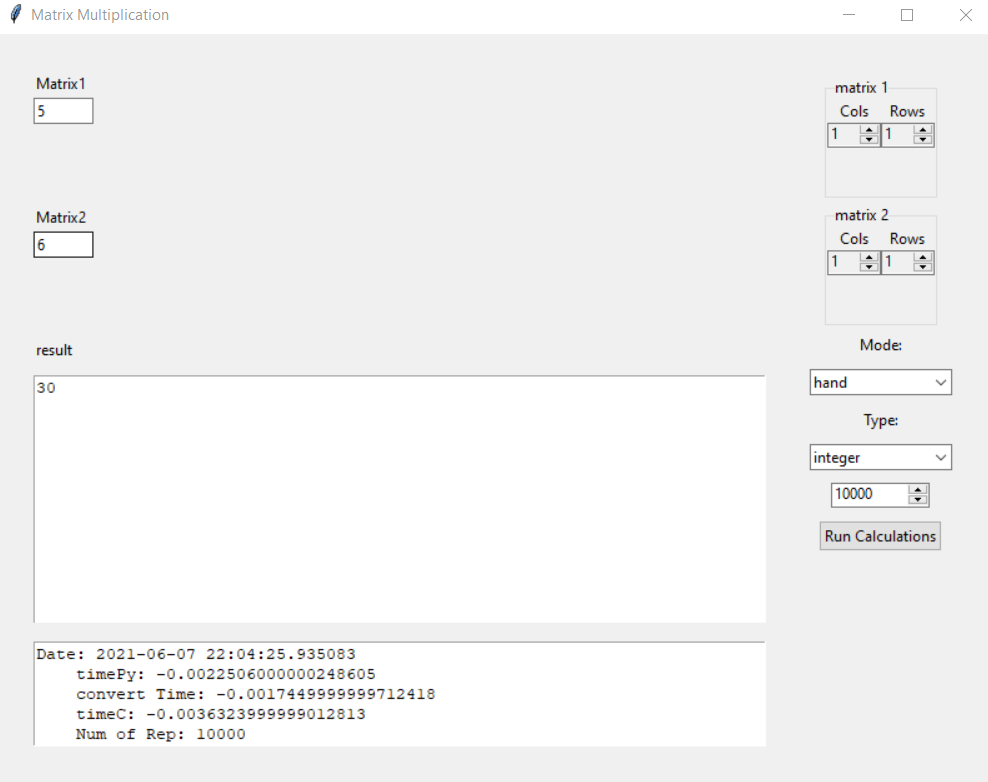


Natomiast do pozostałych zadań nie potrafiłem uzyskać porównanie czasowów w postaci testów więc po prostu uruchomiłem program i przedstawiam wyniki w postaci logów

Zadanie 5







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 tzeba czekać około 4 godzin.Domyślam się że to wynika z tego że użyłem pętli dla tego żeby powtarzać obliczenia. Może sytuacje poprawiło by użycie wątków lub jakichś innych mechanizmów z jakimi jeszcze nie jestem zapoznany.

Zadanie 6.

