

KURS PROGRAMOWANIA W JĘZYKU PYTHON

TYDZIEŃ 5 – MACIERZE I KONTROLA PRZEPŁYWU PROGRAMU

ZADANIA DLA CHĘTNYCH



[Importy do zadań:](#)

import numpy as np

Zadanie 1.

Stwórz dwie macierze wymiarowości 1x9. Przekształć je na macierze wymiarowości 3x3 za pomocą **np.reshape** zapisując wynik w nowych zmiennych i wyświetl ich postać pierwotną i przekształconą.

Zadanie 2.

Używając macierzy z zadania 1 wykonaj ich dodawanie, odejmowanie, mnożenie i dzielenie, zapisując wyniki działań do nowych zmiennych. Wyświetl za pomocą formatowanych stringów wyniki tych działań.

Zadanie 3.

Wybierz dowolne dwie macierze z poprzednich zadań. Używając pętli **for** przeiteruj jedną z nich i podnieś jej rzędy do kolejnych potęg, zaczynając od potęgi 2 i zapisz wyniki tych operacji jako nową macierz w nowej zmiennej. Podobnie postąp z drugą macierzą, potęgując jej rzędy zaczynając od potęgi (1/2). Następnie pomnóż i podziel wyniki tych operacji ze sobą. Wyświetl wszystkie wynikowe macierze.

Zadanie 4 i 5 na następnej stronie!

Zadanie 4.

Spróbuj podnieść dowolnie wybraną macierz do ujemnej potęgi. Jaki błąd został zwrócony?

Przy pomocy bloku **try...except** i pętli **while** zaimplementuj potęgowanie dowolnie wybranej macierzy wynikowej z poprzednich zadań macierzy z potęgą wczytywaną od użytkownika. Jeśli zostanie podana wartość ujemna zwróć błąd za pomocą **raise** i w odpowiedniej sekcji bloku przeiteruj przez macierz pętlami **for** i podnieś elementy macierzy do ujemnej potęgi ręcznie i wypisz wynik. W przeciwnym wypadku wykonaj operację normalnie i wypisz wynik.

Podpowiedź:

definiujemy listę do przechowywania wyniku

for rzed in macierz:

definiujemy listę tymczasową do przechowywania wyników na pojedynczych elementach
for element in rzed:

tu robimy coś z element i dodajemy wynik do listy tymczasowej
dodajemy listę tymczasową do listy wyników

Zadanie 5 *

Używając pętli **while** napisz kalkulator do operacji na macierzach. Kalkulator powinien na początku pobrać od użytkownika serię danych do stworzenia macierzy na których będzie działał. Po wprowadzeniu danych powinny być dostępne operacje dodawania, odejmowania, mnożenia i dzielenia danych wejściowych.

Należy też sprawdzać za pomocą **try...except** czy nie występuje błąd przy wprowadzonych danych i działaniach, a w wypadku błędu powinien być zwracany odpowiedni dla niego komunikat, i program powinien zakończyć pracę za pomocą **exit** z kodem -1.