Metody numeryczne

Laboratorium 1 – Wprowadzenie do GNU Octave, proste ćwiczenia

GNU Octave: https://www.gnu.org/software/octave/, obecna wersja to 5.1.0 z lutego 2019 r.
Dokumentacja online: https://octave.org/doc/interpreter/
Dokumentacja w jednym pliku PDF: https://octave.org/octave.pdf

Ćwiczenia – uruchamianie prostych skryptów

1.1. Napisać skrypt test.m, w którym utworzymy macierz A=[1 2 3; 4 7 0]. Do zmiennej (wektor dwuelementowy) rozA przypiszemy jej rozmiar. Po uruchomieniu skrypt powinien wyświetlić:

```
Macierz A
  1  2  3
  4  7  0
ma rozmiar: 2  3
```

1.2. Do skryptu test.m dodać instrukcję, która narysuje wykres funkcji sinus dla x= [-2pi, 2pi]. Wyświetlanie informacji o macierzy A zakomentować (znak % na początku linii). Przetestować.

1.3. Napisać skrypt trojkat.m, w którym przypiszemy trzem zmiennym (a, b i c) wartości 3.17, 4.5, i 5. Z wzoru Herona proszę wyliczyć pole (warto stworzyć sobie zmienną pomocniczą p). Skrypt powinien wyświetlać informację na dwa sposoby (korzystając z funkcji disp oraz printf):

```
Trójkąt o bokach a=3.17, b=4.5, c=5 ma pole P=7.0084
Trójkąt o bokach a=3.17, b=4.500000, c=5.000000 ma pole P=7.0084
```

Ćwiczenia – instrukcja warunkowa

2.1. W skrypcie trojkat2.m utworzyć trzy zmienne (a, b, c) mające dowolne długości, reprezentujące boki trójkąta. Wykorzystać instrukcję warunkową do sprawdzenia, czy z boków o podanych długościach można zbudować trójkąt. Dla obu możliwych rozwiązań wyświetlić właściwą informację.

Ćwiczenia – pętle (for, while, …) i macierze

3.1. Korzystając z pętli for albo while wyświetlić 200 razy napis "n. Będę się pilnie uczył.", gdzie zamiast n, powinien być numer wyświetlanej linii (od 1 do 200).

3.2. Korzystając z pętli for i funkcji printf() wyświetlić tabliczkę mnożenia, mniej więcej taką:

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

$A=\begin{bmatrix} 1 & 8 & -3 \\ 2 & 1 & 0 \\ 9 & 0 & 8 \end{bmatrix}$ Ćwiczenia wykonać w skrypcie macierz.m.

3.3. Korzystając z pętli for, utworzyć macierz B zwiększając każdy element macierzy A o 3. Porównaj macierz B wynikiem operacji A+3.

3.4. Korzystając z pętli for, utworzyć macierz C podnosząc do potęgi 2 wszystkie elementy leżące na głównej przekątnej (taki sam indeks wiersza i kolumny). Sprawdzić wynik.

3.5. Korzystając z pętli for utworzyć macierz D równą sumie macierzy B i C.
Porównać macierz B z wynikiem operacji B+C.

3.6. Korzystając z pętli, instrukcji warunkowej i funkcji isprime(), sprawdzającej, czy podana liczbą jest liczbą pierwszą, wyświetlić wszystkie liczby pierwsze mniejsze od 100. Zadanie rozwiązać na dwa sposoby, za pomocą pętli for i pętli while. Porównaj wynik z poleceniem: primes(100).

Ćwiczenia – własne funkcje

4.1. Utworzyć plik hello.m. W pliku utworzyć funkcję hello(), która wyświetli napis: "Witaj roku akademicki!". Przetestować działanie.

4.2. W tym samym pliku poniżej utworzyć funkcję suma() z jednym parametrem będącym macierzą 3x3, która policzy sumę wszystkich elementów macierzy i zwróci ją. Funkcję dla macierzy magic(3) wywołać w funkcji hello() i wyświetlić zwróconą wartość.