

## Ćwiczenie 8: Wyrażenia logiczne. Elementy programowania

Wykonać poniższe ćwiczenia przy użyciu możliwie najkrótszych poleceń oraz bez stosowania instrukcji `if`, `for` i `while`

1. Przy pomocy funkcji `rand` wygenerować wektor  $x \in \mathbb{R}^{20}$  o elementach będących liczbami całkowitymi z przedziału  $[-3,10]$ . Następnie przy pomocy jednej instrukcji usunąć z niego wszystkie wystąpienia liczb 2, 5 i 8.
2. Wygenerować losowo wektor składający się ze 100 liczb całkowitych  $a_1, a_2, \dots, a_{100}$ . Usunąć z niego wszystkie elementy równe  $\max(a_1, a_2, \dots, a_{100})$ .
3. Przy pomocy funkcji `randn` (czym różni się ona od funkcji `rand`?) wygenerować wektor  $x \in \mathbb{R}^{25}$ . Jak przy pomocy jednej instrukcji zsumować wszystkie jego elementy dodatnie? Jedną instrukcją przekształcić elementy  $x_i$  wg reguły: jeżeli  $-1 \leq x_i \leq 0$ , to  $x_i$  należy pomnożyć przez 10; w przeciwnym razie należy  $x_i$  zamienić zerem.
4. Przetawić elementy wektora  $x$  tak, aby na samym początku znalazły się same liczby ujemne (w zachowaniu kolejności ich występowania w  $x$ ), a dopiero po nich wszystkie nieujemne (też z zachowaniem kolejności ich występowania w  $x$ ).
5. Dane są liczby rzeczywiste  $r_1, \dots, r_{17}$ , wśród których występują zarówno liczby ujemne, jak i nieujemne. Obliczyć  $x_1y_1 + \dots + x_sy_s$ , gdzie  $x_1, \dots, x_p$  są ujemnymi elementami ciągu  $r_1, \dots, r_{17}$  wziętymi w kolejności ich występowania, a  $y_1, \dots, y_q$  — nieujemnymi wziętymi w odwrotnej kolejności,  $s = \min(p, q)$ .
6. Przy pomocy funkcji `magic` wygenerować macierz  $A \in \mathbb{R}^{8 \times 8}$ . Zapisać instrukcję, która zliczy wszystkie liczby tej tablicy podzielne przez 3. Jak najprościej zamienić te liczby zerami (potrzeba w tym celu co najwyżej dwóch instrukcji)?
7. Dana jest tablica o rozmiarze  $n \times n$ . Jedną instrukcją określić numery kolumn, dla których:
  - (a) wszystkie elementy są zerami;
  - (b) wszystkie elementy kolumny są jednakowe;
  - (c) elementy kolumny tworzą ciąg niemalejący ( $b_{1i} \leq b_{2i} \leq \dots, b_{ni}$ ).
8. Dana jest tablica liczb rzeczywistych  $x_{ij}$  o rozmiarze  $m \times n$ . Uporządkować (czyli zamienić kolejnością) wiersze tej tablicy
  - (a) wg niemalejących sum elementów wierszy;
  - (b) wg niemalejących najmniejszych elementów wierszy;
  - (c) wg nierosnących największych elementów wierszy

Poniższe zadania dotyczą efektywności konstrukcji opartych na instrukcjach iteracyjnych.

1. Wygenerować losowo macierz  $X \in \mathbb{R}^{100 \times 100}$ . Sumę kwadratów jej elementów można wyznaczyć dwoma sposobami: stosując dwa razy funkcję `sum` i podnoszenie do potęgi „element-przez-element”, lub stosując dwie zagnieżdżone pętle `for`. Proszę porównać czas potrzebny do wyznaczenia tej sumy przy użyciu obydwu sposobów (do wyznaczenia czasu wykonywania sekwencji instrukcji można wykorzystać funkcje `etime` lub `cputime`).
2. W podobny sposób jak w poprzednim zadaniu porównać szybkość wykonania ciągu instrukcji

```
for i = 1:1000, u(i) = i^2; end
```

oraz

```
u=zeros(1,1000); for i = 1:1000, u(i) = i^2; end
```

Jak wytłumaczyć różnicę wyników?

W poniższych ćwiczeniach należy napisać odpowiednie skrypty. Jeżeli jest to konieczne (ale tylko wtedy), można użyć instrukcji warunkowej `if` oraz instrukcji iteracyjnych `for` lub `while` (nb. do czego służy instrukcja `break`?).

1. W programowaniu okazują się przydatne komendy `pause`, `echo`, `keyboard`, `input`, `disp`, `error`, `eval`, oraz `sprintf`. Proszę zapoznać się z nimi i przetestować.
2. Wyznaczyć pierwszą liczbę naturalną  $n$ , dla której  $n!$  ma więcej niż 100 cyfr.
3. Napisać ciąg instrukcji wyznaczający techniką Monte Carlo pole pewnej figury (np. koła jednostkowego o środku w początku układu współrzędnych).
4. Napisać skrypt powodujący utworzenie dziesięciu tablic o wymiarach  $5 \times 5$  i elementach losowych. Nazwami tablic mają być  $x_1, x_2, \dots, x_{10}$ .

*Wskazówka:* Wykorzystać funkcje `eval` i `int2str`.

W poniższych zadaniach należy napisać odpowiednie funkcje.

1. Czy w MATLABie parametry przesyła się do funkcji przez zmienną czy przez wartość? Jak wprowadza się zmienne globalne? Czy funkcja musi zwracać jakąś wartość? Czy liczba parametrów aktualnych przekazywanych do funkcji musi być taka sama jak liczba parametrów formalnych? Do czego służą funkcje `nargin` i `nargout`?
2. Do wykrywania błędów w napisanych funkcjach często okazuje się przydatny wbudowany debugger. Proszę zapoznać się z jego podstawowymi komendami poprzez wprowadzenie polecenia `help debug`.
3. Proszę napisać funkcję, która przekształca liczbę naturalną  $n$  wg następującego algorytmu: jeżeli  $n$  jest parzyste, to podziel je na dwa; w przeciwnym razie pomnóż je przez 3 i dodaj 1. Funkcja powinna rysować wykres kolejnych wartości  $n$  aż do momentu gdy  $n$  osiągnie wartość 1.
4. Napisać funkcję zwracającą tablicę wszystkich liczb pierwszych nie przekraczających zadanej granicy  $M$ . (*Wskazówka:* Zastosować algorytm *sita Eratostenesa*.)
5. Dana jest liczba naturalna  $n$ . Znaleźć wszystkie liczby pierwsze mniejsze od  $n$ , które mogą być przedstawione w postaci  $2^p - 1$ , gdzie  $p$  jest także liczbą pierwszą.

6. Dana jest liczba  $x \in \mathbb{R}$ . Otrzymać następującą tablicę o wymiarach  $10 \times 10$ :

$$\begin{bmatrix} 1 & x & \dots & x^8 & x^9 \\ x & 0 & \dots & 0 & x^8 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x^8 & 0 & \dots & 0 & x \\ x^9 & x^8 & \dots & x & 1 \end{bmatrix}$$

7. Dana jest liczba naturalna  $n$ . Wśród liczb  $1, \dots, n$  znaleźć wszystkie, których zapis jest identyczny jak ostatnie cyfry ich kwadratu (jak np.  $6^2 = 36$ ,  $25^2 = 625$  itd.).

8. Dana jest liczba naturalna  $n$ . Sprawdzić czy  $n!$  daje się przedstawić jako iloczyn trzech kolejnych liczb całkowitych.

9. Na podstawie podanej liczby naturalnej  $n$  funkcja powinna zwrócić następujące tablice:

$$A = \begin{bmatrix} 1/1! & 1/2! & \dots & 1/n! \\ 1/1!^2 & 1/2!^2 & \dots & 1/n!^2 \\ \vdots & \vdots & \dots & \vdots \\ 1/1!^n & 1/2!^n & \dots & 1/n!^n \end{bmatrix}, \quad B = \begin{bmatrix} 1 \cdot 2 & & & 0 \\ & 2 \cdot 3 & & \\ & & \ddots & \\ 0 & & & n(n+1) \end{bmatrix}$$

10. Wiadomości dyplomatyczne oraz wojskowe najwyższego szczebla, gdzie poufność jest nadzwyczaj ważna, szyfruje się w taki sposób, że ten, kto bezprawnie podłączy się do linii transmisyjnych, otrzyma jedynie losowo wyglądający ciąg liczb. Aby to osiągnąć, najpierw generuje się ciąg losowych cyfr dwójkowych, zwany kluczem, który znany jest wyłącznie nadawcy i odbiorcy. Nadawca zmienia swoją wiadomość na ciąg cyfr dwójkowych w zwykły sposób przez konwersję każdego znaku w jego standardowy 8-bitowy kod ASCII. Następnie umieszcza się ten ciąg znaków obok klucza i otrzymuje ciąg zakodowany, zmieniając każdy bit na przeciwny, jeżeli bit klucza jest 1, a pozostałe pozostawiając bez zmiany. Zakodowany łańcuch, który okazuje się losowym ciągiem binarnym, jest przekazywany adresatowi. Odebraną wiadomość dekoduje się, dokonując zmian w ten sam sposób, co przy zaszyfrowaniu, używając tego samego klucza, znanego odbiorcy. Oto przykład:

Klucz	0	1	0	0	0	1	1	
								Nadawca
Wiadomość	1	0	1	1	0	0	0	
Wiadomość zaszyfrowana	1	1	1	1	0	1	1	Wiadomość przekazywana
Klucz	0	1	0	0	0	1	1	
								Odbiorca
Wiadomość odczytana	1	0	1	1	0	0	0	

Napisać funkcję kodującą (**encode**) i funkcję dekodującą (**decode**) zadany ciąg znaków. Klucz ma być zmienną globalną.

Poniższe zadania polegają na napisaniu odpowiednich funkcji wykorzystujących rekurencję:

1. Napisać funkcję **pow(x, n)**, gdzie  $x \in \mathbb{R} \setminus \{0\}$ ,  $n \in \mathbb{Z}$ , która wyznaczy wielkość  $x^n$  wg formuły

$$x^n = \begin{cases} 1 & \text{gdy } n = 0 \\ 1/x^{|n|} & \text{gdy } n < 0 \\ x \cdot x^{n-1} & \text{gdy } n > 0 \end{cases}$$

2. Napisać funkcję **C(m, n)**, gdzie  $0 \leq m \leq n$ , która wyznacza współczynniki Newtona  $C_n^m = \binom{n}{m}$  wg formuły

$$C_n^0 = C_n^n = 1; \quad C_n^m = C_{n-1}^m + C_{n-1}^{m-1} \quad \text{dla } 0 < m < n$$

3. Napisać funkcję `root('f', a, b, tol)`, która metodą dzielenia odcinka na połowę określi z dokładnością  $tol$  pierwiastek równania  $f(x) = 0$  na odcinku  $[a, b]$ . Przyjąć, że jeżeli użytkownik poda  $tol \leq 0$  lub w ogóle nie określi zmiennej  $tol$ , to należy wziąć  $tol = eps$ . Ponadto przy zachodzeniu warunku  $f(a) \cdot f(b) > 0$  ma być wyświetlone odpowiednie ostrzeżenie.