Cvičení č. 5 – podmínky, cykly, programování funkcí a dávek

October 16, 2019

Funkce vytvářejte v samostatných m-souborech. Kontrolujte počet (nargin) a vhodnost zadaných argumentů, jinak vypište chybovou hlášku (příkaz error). K textovým výpisům v průběhu funkce můžete využít např. příkaz disp (a funkci num2str). Kontrolujte správnost funkce nebo dávky na vhodných (i nevhodných) argumentech. Pište alespoň stručný popis funkce pro help.

1. Vytvořte funkci faktorial1:
   - má jeden argument: nezáporné celé číslo n (zkontrolujte vhodnost)
   - počítá faktoriál, $n!$, pomocí funkce prod
2. Vytvořte funkci faktorial2:
   - má jeden argument: nezáporné celé číslo n
   - počítá faktoriál, $n!$, pomocí násobení ve for-cyklu
3. Vytvořte funkci faktorial3:
   - má jeden argument: nezáporné celé číslo n
   - počítá faktoriál, $n!$, pomocí rekurzivního volání stejné funkce
4. Vytvořte funkci kombc:
   - má dva argumenty: čísla n a k (ověřte vhodnost)
   - počítá kombinační číslo $\binom{n}{k}$
   - Využijte některou z vlastních funkcí pro výpočet faktoriálu.
5. Vytvořte dávku cislo.m:
   - dávka vyzve uživatele k zadání jednoho čísla
   - pokud je zadáno číslo 0, dávka skončí
   - v opačném případě vypíše nápovědnou hlášku (příkaz disp) a opakuje se zadání čísla (tak dlouho, dokud není zadáno číslo 0)
6. Vytvořte funkci cislo2:
   - má jeden argument: číslo x
   - pokud je $x = 0$, funkce vrátí text nula

- pokud je $x \in \{1; 2; 3\}$, funkce vrátí text male kladne cislo
- pokud je $x = -1$, funkce vrátí text male zaporne cislo
- jinak funkce vrátí text jine cislo

7. Vytvořte funkci presmycka:
   - má jeden argument: textový řetězec x
   - pokud x není zadáno nebo jde o prázdný řetězec, funkce vyzve uživatele k zadání x
   - pokud je x stále prázdný řetězec, vypíše se chybová hláška
   - jinak funkce vrátí přesmyčku řetězce x získanou s využitím funkce randperm
8. Vytvořte funkci koreny:
   - 3 argumenty: koeficienty kvadratického polynomu $p(x) = a\,x^2 + b\,x + c$
   - podle hodnoty diskriminantu spočítá všechny tři kořeny
   - vypíše typ kořenů: jeden realny dvojnasobny, dva realne, nebo dva komplexne sdruzene
   - vypíše unikátní kořeny
   - vrátí unikátní kořeny
   - Funkci naprogramujte s využitím výpočtu pomocí diskriminantu, bez využití vestavěných funkcí pro práci s polynomy.
9. Vytvořte dávku davka.m:
   - dávka načte ze souboru data.mat proměnnou data
   - pokud je data číselná proměnná (tip: isnumeric), dávka vypíše typ proměnné (cislo, vektor, nebo matice), rozměry a obsah proměnné data
   - pokud je data textový řetězec (tip: ischar), dávka vypíše obsah řetězce data a počet jeho znaků bez započítání mezer
   - Dávku vyzkoušejte tak, že do proměnné data uložíte číslo, vektor, matici, nebo řetězec, proměnnou uložíte do souboru data.mat, spustíte dávku a zkontrolujete správnost výsledku.
10. Uvažujte následující náhodný pokus. V košíku je n stejných jablek. Přichází děti, postupně po jednom, a každé se náhodně rozhodne, jestli si vezme jedno jablko. Každé dítě se rozhoduje nezávisle na ostatních dětech a jablko si z košíku vezme s pravděpodobností 1/2. Pokus končí ve chvíli, kdy v košíku nezůstane žádné jablko. Výsledkem pokusu je N = počet dětí, které přijdou, dokud se košík nevyprázdní.

    Pro simulaci uvedeného pokusu vytvořte funkci jablka:
    - má jeden argument: nenulový počet jablek, n, v košíku
    - podle popisu výše spočítá počet dětí N

- V průběhu funkce si pro kontrolu vypisujte pořadí dítěte a jestli si jablko vzalo či nikoliv. Když bude funkce pracovat správně, tyto výpisy pak můžete zakomentovat.

11. Vytvořte dávku jablka_simulace.m pro simulační studii pokusu s jablky:
    - dávka nejdříve vytvoří nulový vektor X délky 1000
    - následně 1000krát provede výpočet funkce jablka pro $n = 20$ a výsledky postupně uloží do složek vektoru X
    - nakonec spočítá a vypíše základní číselné statistické charakteristiky vektoru X: výběrový průměr mean(X), výběrový rozptyl var(X), výběrovou směrodatnou odchylku std(X) a výběrový medián median(X)
12. Náhodný pokus spočívá v jednom hodu klasickou férovou mincí. Sledujeme, jestli při hodu padl líc.

    Vytvořte dávku mince.m pro simulační studii pokusu s hodem mincí:
    - dávka nejdříve vytvoří nulový vektor X délky 1000
    - následně 1000krát simuluje hod mincí a do složek vektoru X zapisuje hodnoty 1 (= padl líc) nebo 0 (= padl rub).
    - nakonec spočítá a vypíše výběrový průměr, výběrový rozptyl a výběrovou směrodatnou odchylku vektoru X
13. Vytvořte funkci elipsa:
    - má dva argumenty: kladná čísla a a b
    - vykreslí elipsu se středem v bodě $[0; 0]$ s poloosami a a b; využijte parametrického popisu elipsy
14. **Úkol na vypracování doma:**

    Vytvořte funkci polynom:
    - má jeden argument: vektor koeficientů obecného polynomu $p(x)$
    - spočítá a vypíše souřadnice průsečíku grafu $p(x)$ s osou y
    - spočítá a vypíše unikátní reálné kořeny polynomu $p(x)$
    - spočítá a vypíše souřadnice reálných stacionárních bodů polynomu $p(x)$ a u každého uvede jeho typ (lokalni maximum, lokalni minimum, nebo inflexni bod)
    - vykreslí graf polynomu $p(x)$ (modrá křivka)
    - do grafu přikreslí průsečík s osou y a kořeny $p(x)$ (modré body), lokální minima (zelené body), lokální maxima (červené body) a inflexní body (černé body)
    - Využijte vestavěných funkcí pro práci (vyhodnocování, derivování, hledání kořenů) s polynomy.
    - Tip: pro nalezení unikátních reálných kořenů prozkoumejte zápis unique(koreny(imag(koreny) == 0)).
    - Tip: k nalezení vhodného rozmezí x-ových hodnot pro vykreslení grafu $p(x)$ pomůže nalezení nejmenšího a největšího čísla ze sjednocení množiny kořenů a stacionárních bodů.