Példatár a bevezetés a Matlab programozásába tárgyhoz

Sáfár Orsolya

1. Ciklusszervezés

1. Írjunk egy olyan `szorzoTabla(n,m)` nevű függvényt, melynek bemenete n és m pozitív egészek, és a kimenete egy mátrix, melynek elemei a $n \times m$-es szorzótábla értékei.

2. Írjunk egy olyan `kozepek(v)` nevű függvényt, amely a bemenetként kapott vektor elemeinek számtani, mértani és harmonikus közepét adja kimenetként egyetlen sorvektorban.

3. Írjunk egy olyan `sorozatRek(n)` nevű függvényt, melynek a bemenete n egy pozitív egész, és a kimenete az $a_{n+1} = 7 - \frac{10}{a_n}$, $a_1 = 4$ képlettel adott sorozat n-dik tagja.

4. Írjunk egy olyan `fibonacciSzorzat(n)` nevű függvényt, melynek bemenete n pozitív egész, és a kimenete a Fibonacci-sorozat n-edik és $n+1$-edik tagjának szorzata. (A Fibonacci sorozat első és második tagja 1, és minden elem az őt megelőző két elem összege.)

2. Feltételek kezelése

1. Írjunk egy olyan `idenMajus(n)` nevű függvényt, melynek bemenete n egy és 31 közötti pozitív egész, és kiírja, hogy idén május n-edike milyen napra esett.

2. Bridge-ben (egy pakli franciakártyával játszott játék) az egyik licitálási rendszerben a leosztásokat pontokkal értékelik. Egy ász 4 pontot ér, egy király 3-t egy dáma 2-t egy bubi egyet. Az indulóerő 13 (vagy több) pont. Írjunk egy olyan `bridgePont(v)` nevű függvényt, melynek bemenete egy 4 elemű vektor, amelyiknek első eleme az ászok száma, a második a királyoké, ... A függvény kimenete legyen 1 ha a leosztásban van indulóerő, 0 egyébként. A függvény visszatérési értéke legyen -1, ha 4-nél több van valamelyik figurából (ilyen leosztás ugyanis nem lehet).

3. Írjunk olyan `idosebb(ev1, ho1, nap1, ev2, ho2, nap2)` nevű függvényt, melynek bemenete 6 pozitív valós szám, melyek két ember születésnapját jelentik. A kimenet legyen 0, ha egyidősek, 1 ha az első ember idősebb, 2, hogy ha a második.

4. Írjunk egy olyan hanyadikelem(v,n) nevű függvényt melynek első bemenete egy számokból álló sorvektor, a második egy valós szám. A függvény határozza meg, hogy hányadik helyen szerepel először v-ben n, a kimenet legyen ezen index. Ha nem szerepel n a vektorban, akkor a kimenet legyen -1.

5. Írjunk olyan reciprokOsszeg(n) nevű függvényt, amelynek kimenete a legkisebb olyan k pozitív egész, melyre az $1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k}$ összeg nagyobb, mint a bemenetként kapott n szám. Ha 10^5-nél is több tag kellene, akkor a kimenet legyen -1.

6. Írjunk egy olyan sorozatRekb(eps) nevű függvényt, melynek a bemenete eps egy pozitív valós szám, és a kimenete az

$$a_{n+1} = 7 - \frac{10}{a_n}, \qquad a_1 = 4$$

képlettel adott sorozat első olyan tagja, amely közelebb van 5-höz mint eps.

7. Írjunk egy olyan fiboRem(n) nevű függvényt, amely megkeresi a Fibonacci-sorozat első n-el osztható tagját. A kimenet legyen ezen tag értéke és sorszáma egyetlen sorvektorban megadva. (A Fibonacci sorozat első és második eleme 1, a képzési szabálya az, hogy minden elem az őt megelőző 2 összege).

3. Logikai indexelés

1. Írjunk egy olyan csere(v,a,b) nevű függvényt, melynek 3 bemenete van: egy v vektor, és két valós szám: a és b. A függvény cserélje ki v minden a-val egyenlő elemét b-re.

2. Írjunk olyan transzformácio(A) nevű függvényt, amelynek bemenete egy A mátrix. A függvény kimenete legyen azon B mátrix, amelyet úgy kapunk, hogy A elemei közül kiválasztjuk a azokat, amelyeknek mindkét indexe páratlan, majd ezek közül a hárommal nem oszthatóakat 0-ra cseréljük.

3. Írjunk olyan erdekesElemek(v,n) nevű függvényt, amelynek bemenetei egy v vektor, és egy n szám. A függvény kimenete legyen az, hogy a v vektornak hány n-nél kisebb abszolút értékű eleme van.

4. Írjunk egy olyan szamolAtlag(v) nevű függvényt, amelynek bemenete egy sorvektor, amelyben valós számok vannak. A függvény kimenete legyen a v vektor 0 és -1 közé eső elemeinek összege.

5. Írjunk olyan mennyiDb(v) nevű függvényt amely meghatározza a bemenetként kapott v vektor páratlan indexű helyein lévő egész elemeinek számát.

6. Adott egy olyan mérés, ahol az eredményeknek a $[-1, 1]$ intervallumba kell esniük; ami nem ide esik, az hibás. Írjunk olyan terjedelem(v) nevű függvényt, amelynek bemenete egy mérési eredményeket tartalmazó vektor, kimenetei pedig a helyes mérési eredmények legkisebbike és legnagyobbika.

4. Véletlen számok

1. Írjunk egy olyan dobas nevű függvényt, amely meghívásakor generál 3 darab 0 és 1 közötti egyenletes eloszlású véletlen számot. Ha van a számok között olyan, amely nagyobb, mint a másik kettő összegének a duplája, akkor írja ki ezeket a számokat a képernyőre, ha nem, akkor generáljon új számhármast addig, amíg nem talál közöttük ilyen tulajdonságú számhármast, és írja ki ezen hármas a képernyőre.
2. Írjunk egy olyan hanyadikfiu(n) nevű függvényt, melynek bemenete n egy pozitív egész. A függvény generáljon egy véletlen 0-1 sorozatot addig, míg az első 1-es nem kapja. Jegyezze meg, hogy hanyadik kísérletre jött ki az első egyes (ez lesz a kísérlet hossza). Ezt a kísérletet ismételje meg összesen n-szer. A kimenet legyen a hosszak átlaga.
3. Írjunk olyan korLap(r) nevű függvényt, amelynek kimenete egy egyenletes eloszlású véletlen pont az origó középpontú, r sugarú körlapról.
4. Írjunk olyan doboKocka(n) nevű függvényt, melynek bemenete egy n természetes szám. A függvény dobjon n-szer egy szabályos dobókockával n-szer, majd ábrázolja egy hisztogramon, hogy melyik szám hányszor fordul elő.

5. Ábrák készítése

1. Írjunk olyan cosinus(a,b,n) nevű függvényt, melynek bemenete n egy pozitív egész, és a<b valós számok. A függvény ábrázolja egy képen az $t \in [a, b]$ intervallumon a $\sin(nt)$ függvényt a felső ábrán, és a $\cos(nt)$ függvényt az alsón.
2. Írjunk olyan felGomb(r) nevű függvényt, amelynek bemenete az r pozitív szám, és ábrázolja a $f(x, y) = r^2 - x^2 - y^2$ függvényt az $[-\frac{r}{2}, \frac{r}{2}] \times [-\frac{r}{2}, \frac{r}{2}]$ négyzet felett. Módosítsuk úgy a függvényt, hogy valódi félgömböt rajzoljon ki.

6. Interpoláció

1. Írjunk olyan linkoz(v1, v2) nevű függvényt, melynek első bemenete az alappontok egy vektorban sorban elrendezve, a második pedig a mért értékek vektora. A függvény határozza meg ezen értékekre legjobban illeszkedő egyenest, és ábrázolja is a pontokkal együtt. A függvény kimenete legyen az egyenes meredeksége.
2. Írjunk egy olyan legjobbParabola(v1,v2) nevű függvényt, melynek bemenetei v1 és v2 oszlopvektorok, a v1 tartalmazza a mérési pontokat, v2 a mérési értékeket. A függvény számítsa ki az ezen pontokra legkisebb négyzetes értelemben legjobban illeszkedő parabola ($ax^2 + bx + c$) együtthatóit, majd ábrázolja is a pontokat a parabolával együtt. A kimenet legyenek a parabola három együtthatója egyetlen sorvektorban, $[a, b, c]$ sorrendben.

3. Írjunk olyan interpolals(x,y,x0) nevű függvény Matlabban, amelynek bemeneti közül az első valós számokból álló sorvektor, a második bemenet pedig egy függvény helyettesítési értékei ezen pontokban szintén egy sorvektorban adva. A függvény kimenete legyen az ezen pontokra illesztett spline interpoláló polinom helyettesítési értéke az x0 pontban.

4. Írjunk olyan kozelit(t,v) nevű függvényt, amelynek bemenetei sorvektorok. A t sorvektorban megadott pontokban megmértük egy ismeretlen függvény értékeit, amelyről ismert, hogy $a\sin(t) + b\sin(2t) + c\mathrm{e}^{-t}$ alakú, a mérési eredmények szerepelnek a v vektorban. A függvényünk kimenete legyen az a, b, c számok egyetlen oszlopvektorban.

5. Írjunk olyan rungell(n) nevű függvényt, melynek bemenete egy n pozitív egész. A függvény interpolálja egy $(n-1)$-edfokú polinommal az

$$f(x) = \frac{1}{1+x^2}$$

függvényt az $[-5, 5]$ intervallumon vett n darab egyforma távolságra lévő ponton. Ábrázoljuk az eredeti függvényt és az interpoláló polinomot is a [-5,5] szakaszon.

7. Sztringek kezelése

1. Írjunk olyan romai(s) nevű függvényt, melynek bemenete egy sztring (most feltesszük, hogy csak az angol ábécé betűiből áll), a kimenete pedig egy olyan sztring, amelyben minden karakter ki van cserélve az ábécé következő betűjére, illetve a 'z' 'a'-ra.

2. Írjunk olyan kisBetuk(s) nevű függvényt, amelynek bemenete egy sztring. A függvény kimenete legyen a kisbetűk aránya az s-ben lévő betűk között (csak az angol ábécé betűit ideértve).

3. Írjunk olyan maganhangzo(s1) nevű függvényt amelynek bemenete egy s1 sztring, és a függvény kimenete, hogy hány magánhangzó van s1-ben (csak az angol ábécé magánhangzóit ideértve).

4. Írjunk olyan gyakorimg(s1) nevű függvényt amelynek bemenete egy s1 sztring, és a függvény kimenete, hogy melyik a leggyakoribb magánhangzó s1-ben (csak az angol ábécé magánhangzóit ideértve).

5. Írjunk egy mhar(s) nevű függvényt, melynek bemenete egy sztring, amely egy angol nyelvű szöveget tartalmaz. A függvény kimenete legyen a mássalhangzók és a magánhangzók aránya s-ben.

6. Írjunk egy mhtr(s) nevű függvényt, melynek bemenete egy sztring, amely egy angol nyelvű szöveget tartalmaz. A függvény kimenete legyen a mássalhangzó torlódások száma s-ben.

7. Írjunk olyan fd(s1) nevű függvényt, melynek bemenete egy s1 sztring. A függvény visszatérési értéke legyen az 'a' betűk aránya a sztring összes betűjéhez viszonyítva.

8. Írjunk egy olyan romaiSzam(s) nevű függvényt, melynek egyetlen bemenete egy sztring (ezt nem kell ellenőrizniük). Ha a bemenetként kapott sztring egy 1 és 89 közötti szabályos alakban írt római szám, akkor adja vissza az értéket int8 típusú számként, egyébként a kimenet legyen -1. (Segítség: könnyebb eldönteni, hogy mi egy szám római alakja, mint egy sztringről, hogy szabályos alakú római szám-e).

8. Struct és cell típusú tömbök

1. Írjunk olyan feldolgozs(v) nevű függvényt, amelynek bemenete egy v cell típusú sorvektor, amely tartalmazhat lebegőpontos számokat és sztringeket is, de más típusú adatot nem. A függvény kimenete legyen a v vektorban szereplő szigorúan 0 és 100 közötti számok darabszáma és átlaga (azt, hogy valami szám-e az isnumeric() függvénnyel ellenőrizhetjük).

2. Írjunk egy adatok(c) nevű függvényt, melynek bemenete egy c cell típusú változó, melyről tudjuk, hogy vegyesen vannak benne üres helyek, vektorok és mátrixok. A függvény kimenete legyen a c-ben lévő összes szám darabszáma.

3. Írjunk olyan cd(v) nevű függvényt, melynek bemenete egy cell típusú változó. Erről a változóról tudjuk, hogy első és utolsó oszlopában csak sztringek vannak, a többiben számok. A függvényünk kimenete legyen a páratlan indexű számokat tartalmazó sorokban szereplő számok összege.

9. Fájlok kezelése

1. Adott egy xls fájl, melynek első oszlopában nevek vannak, a másodikban a Neptun kódok, a következő 14-ben pedig a hetente elért házi pontszámok. Írjunk olyan hazi80(s) nevű függvényt, amelynek bemenete a fájl teljes neve sztringként, kimenete pedig a legalább 80 pontot elért hallgatók nevei egy cell típusú változóban.

2. Adott egy xls fájl, melynek első oszlopában nevek vannak, a másodikban a születési dátumok éééé.hh.nn formátumban. Írjunk egy felnotte(s) nevű függvényt, melynek bemenete a fájl teljes neve sztringként, a kimenete pedig a 18 évet betöltött emberek száma.

3. Írjunk olyan olvaso() nevű függvényt, mely megnyit egy bemenetként kapott fájlt. Ha nem sikerül megnyitni, akkor hibaüzenettel kilép, különben pedig kiírja a képernyőre a fájl első és utolsó sorát.

4. Írjunk olyan normal() nevű függvényt amely beolvassa az adatokhf.txt nevű fájlt, amelyben soronként egy darab szám található. A függvény számolja ki ezen számok

átlagát, és a kimenete legyen egy olyan vektor, amelyben minden szám helyére az adott szám mínusz összes szám átlaga kerül.

5. Adott egy txt fájl, amelyben soronként egy darab valós szám szerepel. Írjunk olyan nyereg(s) nevű függvényt, melynek bemenete a fájl teljes neve sztringként, a kimenetei pedig a páros indexű helyeken szereplő számok maximuma és a páratlan helyen lévő számok minimuma.

6. Írjunk olyan filed(s) nevű függvényt, melynek bemenete egy txt fájl teljes neve sztringként. Ebben a fájlban csak egész számok vannak, soronként azonos, de ismeretlen darabszámú. Olvassuk be egy megfelelő méretű mátrixba a számokat, majd konvertáljuk a legkisebb helyet foglaló olyan egész típussá a mátrixot, melybe még adatvesztés nélkül elférnek. A kimenet legyen ezen mátrix.

10. Perzisztens változók, függvény overload

1. Írjunk olyan mozgoatlag10(x) nevű függvényt melynek bemenete egy x valós szám, a kimenete pedig a függvény előző 10 hívásakor adott bemenetének átlaga és maximuma. Ha még nem hívtuk meg 10-szer a függvényt, akkor az addigi hívások átlagát és maximumát adja vissza.

2. Írjunk egy olyan korokc(r,x_0, y_0) nevű függvényt, amelyet ha 0 bemenettel hívunk meg, akkor kirajzol egy origó középpontú 1 sugarú kört, ha eggyel akkor egy origó középpontú r sugarú kört, ha hárommal, akkor egy (x_0, y_0) középpontú r sugarú kört.

3. Írjunk egy olyan statisztikak(v,c,d) nevű függvényt, melynek bemenete egy v oszlopvektor és két valós szám. Ha csak egy bemenettel hívjuk meg a függvényt, akkor a kimenet legyen a v-ben szereplő számok átlaga és terjedelme (a legkisebb és legnagyobb elem különbsége), ha hárommal, akkor pedig a v vektor elemei közül a c és d számok közé eső elemeinek átlaga és terjedelme.

4. Írjuk egy olyan modosit(v) nevű függvényt, melynek bemenete egy oszlopvektor, amely int8 típusú számokat tartalmaz. Ha egyszer hívjuk meg a függvényt, akkor a kimenet legyen egy olyan vektor, amelyben v elemei vannak lebegőpontos számként, ha másodjára (vagy ennél és többször) hívjuk meg, akkor a kimenetben v minden negatív elemét cserélje 0-ra a nemnegatív számokat pedig alakítsa át lebegőpontossá.

5. Írjunk olyan pefel5(a) nevű függvényt, amelynek bemenete egy valós szám. Első hívásakor a kimenete legyen a bemenet értéke, ha többször hívtuk meg, akkor a kimenet legyen az előző bemenet +1.

11. Lineáris algebra

1. Írjunk egy olyan normaOszlop(A) nevű függvényt, amelynek bemenete egy A mátrix. A kimenet legyen a mátrix oszlopai közül annak a sorszáma, amelyben a legkisebb a benne szereplő elemek abszolút értékének összege.

2. Egy mátrixot diagonálisan dominánsnak nevezünk, ha minden sorra igaz, hogy a főátlóban lévő elem abszolút értéke nagyobb, mint a sorban lévő többi elem abszolút értékének összege. Írjunk egy ddMatrixe(A) nevű függvényt, melynek bemenete egy négyzetes mátrix. A függvény döntse-el, hogy a bemenet diagonálisan domináns-e vagy sem, és ezt írja ki a képernyőre.

3. Írjunk egy olyan frob(A) nevű függvényt, amely kiszámítja a bemenetként kapott A mátrix Frobenius-normáját, legyen ez f_1, és a mátrix összes elemének négyzetösszegének a négyzetgyökét, legyen ez f_2. A függvényünk kimenete legyen $f_1 - f_2$. Segítség: egy A valós elemű mátrix Frobenius-normájának négyzete az $A \cdot A^T$ mátrix nyoma (a nyom a főátlóban szereplő számok összege).

4. Az $n \times n$-es Hilbert mátrix (i, j)-edik eleme $a_{i,j} = \frac{1}{i+j-1}$. Írjunk olyan myHilbert(n) nevű függvényt, amelynek bemenete egy pozitív egész, és amely megkonstruálja ezen mátrixot. A kimenete legyen a mátrix legnagyobb abszolút értékű sajátértéke, és a hozzá tartozó sajátvektor.

5. Írjunk egy savszelesseg(A) nevű függvényt, melynek bemenete egy négyzetes, nem csupa nullából álló mátrix. A függvény határozza meg a mátrix sávszélességét, azaz a főátlót nulladik átlónak számítva a legtávolabbi olyan ezzel párhuzamos mellékátló számát, amelyben van nem 0 elem.

6. Írjunk egy olyan stabile(A) nevű függvényt, amelynek bemenete egy A négyzetes mátrix. A függvény döntse el, hogy A stabil-e, és ezt írja ki a képernyőre. Egy mátrixot stabilnak nevezünk, ha az összes sajátértékének valós része negatív.

7. Írjunk olyan matrixK(n) nevű függvényt, amelynek bemenete egy n pozitív egész. A függvény készítse el *ritka* mátrixként azon mátrixot, melynek főátlójában csupa +1-es, az erre merőleges mellékátlójában csupa -1-es áll. Ha keresztezi egymást a két átló, akkor a metszéspontban álljon +1.

8. Írjunk olyan becslo(A) nevű függvényt, amelynek bemenete egy A négyzetes szimmetrikus mátrix (ekkor tudjuk, hogy minden sajátértéke valós). Az első Gersgorin–tétel segítségével konstruáljuk meg azon intervallumot a valós számegyenesen, amelybe biztos, hogy az összes sajátérték beleesik (a sajátértékek kiszámítása nélkül). A kimenet legyen ezen intervallum két végpontja.

9. Vegyük a 3×3-as Hilbert mátrixot, ahol $a_{i,j} = \frac{1}{i+j-1}$, legyen ez A, és a $b = [1; 1; 1]$ vektort. Oldjuk meg az $Ax = b$ lineáris egyenletrendszert, majd számítsuk ki az

$A\tilde{x} - b$ maradékvektort az $\tilde{x} = [1.5; -15; 21]$ vektorra. Keressünk magyarázatot a tapasztaltakra.

12. Differenciálegyenletek megoldása

1. Tegyük fel, hogy kenyeret sütünk egy kemencében, amikor kivesszük legyen a hőmérséklete 100°C. A Newton-féle lehűlési tétel értelmében egy test hőmérsékletének változási sebessége egyenesen arányos a a környezet és a test hőmérsékletének különbségével. Ez alapján $T(t)$-vel jelölve a kenyér hőmérsékletét a t. percben, és a környezet hőmérsékletét állandó 20° C-nek feltételezve a következő differenciálegyenletet írhatjuk fel:
$$T'(t) = k(20 - T(t)),$$
ahol a k arányossági tényező értéke legyen 10^{-6}. Állapítsuk meg, hogy hány perc alatt hűl ki a kenyér 40° C fokra.

2. Tekintsük az $x'(t) = ax(t)$, $x(0) = 1$ elsőrendű differenciálegyenletet, ahol $a \in \mathbb{R}$ valós paraméter. Ha tudjuk, hogy $x(2) = 0.1$ akkor mennyi lehet a értéke? Írjunk egy olyan `democ` nevű függvényt, melynek kimenete ezen becsült a érték. A megoldást egyetlen függvénnyel készítsük el.

3. Tekintsük az $x'(t) = ax(t)$, $x(0) = 1$ elsőrendű differenciálegyenletet, ahol $a \in \mathbb{R}$ valós paraméter. Ha tudjuk, hogy $a = 0.1$ akkor milyen t-re lesz először $x(t) = 2$? Írjunk egy olyan `demodd` nevű függvényt, melynek kimenete ezen t érték. A megoldást egyetlen függvénnyel készítsük el.

4. Tekintsük az $x''(t) = -ax(t)$, $x(0) = 0, x'(0) = 1$ másodrendű differenciálegyenletet, ahol $a \in \mathbb{R}$ valós paraméter. Ha tudjuk, hogy $a = 0.2$ akkor mennyi lehet $x(2)$ értéke? Írjunk egy olyan `demof` nevű függvényt, melynek kimenete ezen $x(2)$ érték. A megoldást egyetlen függvénnyel készítsük el.

5. Tekintsük az $x'(t) = 3x(t) + \alpha$, $x(0) = 1$ elsőrendű differenciálegyenletet, ahol $\alpha \in \mathbb{R}$ valós paraméter.

   (a) Számítsuk ki a megoldás értékét a $t = 3$ pontban, ha $\alpha = 1$.

   (b) Milyen t-re lesz először $x(t) = 1.5$ ha $\alpha = -1$?

   (c) Számítsuk ki α értékét, ha $x(1.5) = 10$!