

I. Vezérlési szerkezetek

- 1) A program olvasson be tetszőleges egész számot parancssori argumentumból, és határozza meg azok minimumát. Szóljon, ha ehhez nincs elég (legalább 2) bemenet.
- 2) A program olvasson be tetszőleges egész számot parancssori argumentumból, és határozza meg azok maximumát. Szóljon, ha ehhez nincs elég (legalább 2) bemenet.
- 3) A program olvasson be tetszőleges egész számot parancssori argumentumból, és határozza meg azok összegét. Szóljon, ha ehhez nincs elég (legalább 2) bemenet.
- 4) A program olvasson be tetszőleges egész számot parancssori argumentumból, és határozza meg azok átlagát. Szóljon, ha ehhez nincs elég (legalább 2) bemenet.
- 5) A program olvasson be tetszőleges egész számot parancssori argumentumból, és határozza
- 6) A program olvasson be tetszőleges egész számot parancssori argumentumból, és számolja meg, hogy összesen hány pozitív számot olvasott be. Szóljon, ha nem kap bemeneti paramétereket.
- 7) A program olvasson be tetszőleges egész számot parancssori argumentumból, és számolja meg, hogy összesen hány negatív számot olvasott be. Szóljon, ha nem kap bemeneti paramétereket.
- 8) A program olvasson be tetszőleges egész számot parancssori argumentumból, és számolja meg, hogy összesen hány nullát olvasott be. Szóljon, ha nem kap bemeneti paramétereket.
- 9) A program olvasson be tetszőleges egész számot parancssori argumentumból, és írja ki, hogy a páratlan, vagy a páros számok összege a nagyobb.
- 10) A program olvasson be két egész számot (x és y) parancssori argumentumból. Írja ki az x és y között található összes egész számot (x -> y felé haladva), valamint szóljon, ha a bemenet nem megfelelő formátumú (itt több problémát is vizsgálni kell).
- 11) Számtani sorozat (egész számokból) első n elemének kiszámolása tömb segítségével. A program a sorozat kezdőelemét, a differenciát, illetve n értékét parancssori argumentumból olvassa, valamint szóljon, ha nem megfelelő a bemeneti paraméterek száma.
- 12) Mértani sorozat (egész számokból) első n elemének kiszámolása tömb segítségével. A program a sorozat kezdőelemét, a kvócienszt, illetve n értékét parancssori argumentumból olvassa, valamint szóljon, ha nem megfelelő a bemeneti paraméterek száma.
- 13) Fibonacchi sorozat első n elemének kiszámítása tömb segítségével. n értékét a program parancssori argumentumként kérje, és szóljon, ha nem pontosan egy paramétert kapott.
- 14) Tetszőleges számú szó beolvasása parancssori argumentumból. A program számolja meg, hogy milyen hosszúságú szóból mennyi van, ezeket az értékeket tárolja tömbben (pl. 1 hosszú szó 3 db, 2 hosszú szó 0 db, 3 hosszú szó 4 db, stb.). A kapott értékeket írja ki, valamint szóljon, ha nincs megadva bemeneti szó.
- 15) A program olvasson be 3 számot parancssori argumentumból. Szóljon, ha nincs elég bemenete. Döntse el, hogy a számok lehetnek-e egy háromszög oldalhosszai. Ha igen, számolja ki annak kerületét és területét. A terület kiszámítására alkalmazza Heron képletét.
- 16) A program olvasson be egy n egész számot parancssori argumentumból. Szóljon, ha nincs bemenet. Dobjon egy hatoldalú kockával n alkalommal (véletlenszerű értékeket), és mentse tömbbe, hogy az egyes értékeket hányszor dobta ki. Ezután írja ki, hogy az különböző értékek az dobások hány százalékában fordultak elő.
- 17) A program olvasson be legalább 2 számot parancssori argumentumból, és szóljon, ha nincs elég bemenet. Vizsgálja meg, hogy a kapott számok sorrendje növekvő, csökkenő, vagy egyik sem, és ezt írja ki.

- 18) Beérkező rendelések összértékét szeretnénk meghatározni. Ehhez parancssorról a rendelések sorszámaikat kapjuk meg, ami tetszőleges darab egész szám formájában jön. Négy különböző rendelésünk van, az 1-es 1000 Ft, 2-es 5000 Ft, 3-as 7000 Ft és a 4-es 9000 Ft. Egy rendelés sorszáma többször is szerepelhet. Adjuk össze az érkező sorszámnak megfelelő értékeket, és az összeget írjuk ki. Ha olyan szám szerepel, ami nem rendes sorszám (nem 1,2,3,4), akkor minden esetben írjuk ki konzolra, hogy "Nem megfelelo sorszam". A program szóljon, ha nincs legalább 1 bemenet.
- 19) Készítsünk programot, ami kiszámolja egy szabályos sokszög területét. A program két bemenetet vár parancssori argumentumból: a sokszög oldalainak a számát, valamint az oldalhosszat. A probléma csak az, hogy a programozási nyelvünkben "elomlott" a szorzás, így ezt a műveletet nem használhatjuk a feladat megoldása során. A program szóljon, ha nem 2 bemenetet kapott.
- 20) Egy könyv olvasása közben kíváncsiak leszünk, hogy melyik lehet a benne található leghosszabb szó, ezért készítünk egy programot, ami eldönti ezt nekünk. A program a könyvben található szavakat kapja bemenetnek parancssori argumentumból (tetszőleges mennyiségű szó). Írjuk ki, hogy melyik a bemenetben található leghosszabb szó (ha több leghosszabbat is találunk, akkor bármelyiket kiírhatjuk). A program szóljon, ha nem kapott bemenetet.
- 21) Matek ZH-ra készülés közben elomlott a számológépünk, ezért elhatározzuk, hogy írunk egy programot, ami helyettesíteni fogja. A számológépünk mindig két értéken végez el műveletet, ezért a program pontosan 3 bemenetet kapjon parancssori argumentumból: az első számot, az operátort, valamint a második számot (pl. $3 + 1$). Írjuk ki, hogy mi lesz a megadott kifejezés eredménye. A program szóljon, ha nem 3 bemenetet kapott. (tipp: szorzásnál $*$ helyett más karaktert használjunk bemenetnek, pl. x)
- 22) Az előző feladatban elkészített számológépünk nagyon jól működik, de rövidesen rájövünk, hogy hosszabb kifejezéseket is ki szeretnénk számolni. Írunk tehát egy új programot, ami az alábbi felépítésű bemenetet kapja: műveletek száma (n), kezdőérték, majd $n \cdot \{\text{operátor, operátorhoz tartozó érték}\}$. (pl. $3 \ 1 + 2 \times 2 - 1 \rightarrow 3$ művelet, 1 a kezdőérték, és először $+2$ -t végezzük el, majd $\times 2$ -t, majd -1 -t). A program számolja ki az így kapott műveletrend eredményét. A műveleteket abban a sorrendben hajtsa végre, ahogy a bemenetben szerepelnek. A program szóljon, ha nem megfelelő a bemenet.
- 23) Jancsi és Juliska az erdőben sétálva minden kereszteződésnél morzsákat szór el. Jancsi az első kereszteződésben 3 szemet, Juliska a második kereszteződésben 2 szemet. Minden további kereszteződésben annyi szemet szórnak el, mint az előző kettőben összesen. A sétájuk az n -edik kereszteződésben ér véget, ugyanis itt egy életnagyságú mézeskalács házzal találják szemben magukat, de mielőtt bemennek, még itt is elszórnak valamennyi morzsát. Mennyi morzsát szórtak el a ház előtt? A kereszteződések számát (n értékét) parancssori argumentumból kapjuk bemenetnek. A program írja ki az utolsó kereszteződésben elszórt morzsa mennyiségét. A program szóljon, ha nem kapott bemenetet.
- 24) Kisboltunkba egy szállítmány dinnye érkezik, amit átvétel előtt leellenőrzünk, lemérjük mindegyiket. A dinnyék súlyai lesznek a program bemenete, ezeket parancssori argumentumból olvassuk be (tetszőleges darab egész szám). Tudjuk, hogy ha egy dinnye súlya páros szám, és osztható 3-al is, akkor biztos rohadt lesz, ezeket nem szeretnénk átvenni. A program állapítsa meg a súlyok alapján, hogy hány problémás dinnye található a szállítmányban. A program szóljon, ha nem kapott bemenetet.
- 25) Készítsünk programot, ami megadott háromszögek közül kiválasztja, hogy melyiknek a legnagyobb a területe. A program parancssori argumentumból kapja a bemenetét az alábbi módon: először a háromszögek darabszáma (n), utána hármassával az oldalhosszak ($n \cdot \{\text{oldal1 oldal2 oldal3}\}$) következnek (pl. $2 \ 1 \ 2 \ 3 \ 3 \ 4 \rightarrow 2$ háromszög, az egyik oldalai 1,2,3, a másik oldalai 3,3,4). Állapítsuk meg, hogy hanyadik háromszögnek a legnagyobb a területe. A program szóljon, ha nem megfelelő a bemenet.
- 26) Reklámozni szeretnénk a legújabb termékünket, és készítünk egy programot, ami elektronikus hirdetőtáblaként megteszi ezt nekünk. A hirdetőtábla egymás után szövegeket fog kiírni a képernyőre: vagy a termék nevét, vagy a termék árát. A program 3 bemenetet kap parancssori argumentumból: az első a termék neve, a második a termék ára, a harmadik pedig egy szám, hogy összesen hány sort szeretnénk kiírni a táblára. Alapesetben a hirdetőtábla mindig a termék nevét írja ki, de minden 5. kiírás a termék árát fogja kiírni ehelyett. A program szóljon, ha nem 3 bemenetet kap.

II. Metódusok

- 1) Készíts programot, amely a parancssori argumentumból tetszőleges darab egész számot olvas be. Szóljon, ha nincs legalább 1 bemenet, és lépjen ki. A programnak legyen egy metódusa, ami egy int tömböt vár paraméternek, és visszaadja a tömbben található számok összegét. Ezt a visszaadott értéket írja ki.
- 2) Készíts programot, amely a parancssori argumentumból tetszőleges darab egész számot olvas be. Szóljon, ha nincs legalább 1 bemenet, és lépjen ki. A programnak legyen egy metódusa, ami egy int tömböt vár paraméternek, és visszaadja a tömbben található számok átlagát. Ezt a visszaadott értéket írja ki.
- 3) Készíts programot, amely a parancssori argumentumból tetszőleges darab egész számot olvas be. Szóljon, ha nincs legalább 1 bemenet, és lépjen ki. A programnak legyen egy metódusa, ami egy int tömböt vár paraméternek, és visszaadja a tömbben található legkisebb elem értékét. Ezt a visszaadott értéket írja ki.
- 4) Készíts programot, amely a parancssori argumentumból tetszőleges darab egész számot olvas be. Szóljon, ha nincs legalább 1 bemenet, és lépjen ki. A programnak legyen egy metódusa, ami egy int tömböt vár paraméternek, és visszaadja a tömbben található legnagyobb elem értékét. Ezt a visszaadott értéket írja ki.
- 5) Készíts programot, amely a parancssori argumentumból tetszőleges darab egész számot olvas be. Szóljon, ha nincs legalább 1 bemenet, és lépjen ki. A programnak legyen két metódusa: mindkettő egy-egy int tömböt vár paraméternek, és visszaadja a tömbben található legkisebb, illetve legnagyobb elem értékét. A visszaadott értékekről döntse el, hogy a legnagyobb elem osztható -e a legkisebb elemmel, és írjon ki igen/nem szöveget ettől függően.
- 6) Készíts programot, amely a parancssori argumentumból tetszőleges darab egész számot olvas be. Szóljon, ha nincs legalább 1 bemenet, és lépjen ki. A programnak legyen egy metódusa, ami egy int tömböt vár paraméternek, és visszaadja, hogy hány páros szám található a tömbben. Ezt a visszaadott értéket írja ki.
- 7) Készíts programot, amely a parancssori argumentumból tetszőleges darab egész számot olvas be. Szóljon, ha nincs legalább 1 bemenet, és lépjen ki. A programnak legyen egy metódusa, ami egy int tömböt vár paraméternek, és visszaadja, hogy hány páratlan szám található a tömbben. Ezt a visszaadott értéket írja ki.
- 8) Készíts programot, amely a parancssori argumentumból tetszőleges darab egész számot olvas be. Szóljon, ha nincs legalább 1 bemenet, és lépjen ki. A programnak legyen egy metódusa, ami két paramétert vár: egy int tömböt, valamint egy logikai értéket, hogy a páros, vagy páratlan számokat akarjuk megszámolni a tömbben. A metódus adja vissza, hogy a logikai paraméter által meghatározott típusú számokból hány darab található a tömbben. A metódust kétszer hívd meg: számolja össze a páros és páratlan értékeket is. A program írja ki, hogy páros, vagy páratlan számokból volt több a bemenetben.
- 9) Készíts programot, amely a parancssori argumentumból tetszőleges darab egész számot olvas be. Szóljon, ha nincs legalább 2 bemenet, és lépjen ki. Az első bemenet egy lépésköz értéke lesz, minden további bemenet kerüljön egy int tömbbe. A programnak legyen egy metódusa, ami egy int tömböt, valamint egy egész lépésközt vár paraméternek, és összeadja tömb azon elemeit, amit a lépésköz határoz meg (az első elemmel kezdve). Hívja meg ezt a metódust a beolvasott értékkel és tömbbel, a visszaadott eredményt írja ki.
- 10) Készíts programot, amely a parancssori argumentumból tetszőleges darab egész számot olvas be. Szóljon, ha nincs legalább 2 bemenet, és lépjen ki. Az első bemenet egy lépésköz értéke lesz, minden további bemenet kerüljön egy int tömbbe. A programnak legyen egy metódusa, ami egy int tömböt, valamint egy egész lépésközt vár paraméternek, és összeadja tömb azon elemeit, amit a lépésköz határoz meg, és párosak (az első elemmel kezdve). Hívja meg ezt a metódust a beolvasott értékkel és tömbbel, a visszaadott eredményt írja ki.
- 11) Készíts programot, amely a parancssori argumentumból tetszőleges darab egész számot olvas be. Szóljon, ha nincs legalább 2 bemenet, és lépjen ki. Az első bemenet egy lépésköz értéke lesz, minden további bemenet kerüljön egy int tömbbe. A programnak legyen egy metódusa 3 paraméterrel: egy int tömb, egy egész lépésköz, valamint egy logikai változó, hogy balról, vagy jobbról járja be a tömböt. A metódus adja össze a tömb azon elemeit, amit a lépésköz határoz meg (a kezdő elem a logikai értéktől függően a legelső/legutolsó legyen). Kétszer hívja meg ezt a metódust a beolvasott értékkel és tömbbel: balról és jobbról is határozza meg az összeget. Döntsd el, hogy melyik irányból kapott nagyobb értéket, és ezt írja ki.
- 12) Készíts programot, amely a parancssori argumentumból tetszőleges darab egész számot olvas be. Szóljon, ha nincs legalább 3 bemenet, és lépjen ki. Az első és a második bemenet egy-egy index érték lesz, minden további bemenet kerüljön egy int tömbbe. A programnak legyen egy metódusa, ami egy int tömb valahányadik elemét határozza meg. Ehhez 3 paramétert kérjen: az tömböt, a keresendő elem indexét, valamint egy logikai értéket, hogy balról, vagy jobbról keressen a tömbben. A program hívja meg ezt a metódust kétszer: az első beolvasott indexel balról keress a tömbben, a másodikkal pedig jobbról. Mindkét eredményt írd ki.

III. Rekurzív metódusok

FONTOS: Az alábbi feladatok esetében sehol nem használhatsz ciklusokat (**while**, **for**)! A feladatokat minden esetben rekurzív elven kell megoldani úgy, hogy a metódus más paraméterezéssel újra meghívja saját magát.

- 1) Írj programot, ami parancssori argumentumból egy egész számot olvas be. Szóljon, és lépjen ki, ha nincs bemenet. A programnak legyen egy metódusa, ami egy int paramétert vár. A metódus ne adjon vissza semmit, de rekurzívan írja ki az összes számot 1 és a kapott paraméter között növekvő sorrendben.
- 2) Írj programot, ami parancssori argumentumból egy egész számot olvas be. Szóljon, és lépjen ki, ha nincs bemenet. A programnak legyen egy metódusa, ami egy int paramétert vár. A metódus ne adjon vissza semmit, de rekurzívan írja ki az összes számot a kapott paraméter és 1 között csökkenő sorrendben.
- 3) Írj programot, ami parancssori argumentumból egy egész számot olvas be. Szóljon, és lépjen ki, ha nincs bemenet. A programnak legyen egy metódusa, ami egy int paramétert vár. A metódus rekurzívan adja össze az összes számot 1 és a kapott paraméter között, és adja vissza az eredményt. A metódus által visszaadott értéket írd ki.
- 4) Írj programot, ami parancssori argumentumból egy egész számot olvas be. Szóljon, és lépjen ki, ha nincs bemenet. A programnak legyen egy metódusa, ami egy int paramétert vár. A metódus rekurzívan számolja ki a kapott paraméter faktoriálisát, és adja vissza az eredményt. A metódus által visszaadott értéket írd ki.
- 5) Írj programot, ami parancssori argumentumból két egész számot olvas be. Szóljon, és lépjen ki, ha nincs 2 bemenet. Az első szám egy hatványozandó érték (alap), a második pedig a hatványkitevő. A programnak legyen egy metódusa, ami két int paramétert vár: alapot és kitevőt. A metódus rekurzívan számolja ki a kapott paraméterek alapján a hatványt, és adja vissza az eredményt. A metódus által visszaadott értéket írd ki.