

Problémamegoldás és algoritmusok

Beadandó projekt feladatok

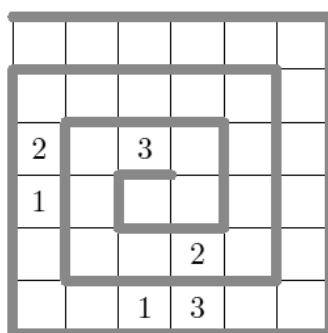
1. Készítsen sztring összeadó, kivonó, összehasonlító, szorzó algoritmusokat! **(4 fő)**
Az algoritmus paraméterként két pozitív egész számokat tartalmazó sztringet kap, melyekkel elvégzi a kívánt műveletet. Összeadás esetén visszatér a két szám összegét tartalmazó sztringgel. Kivonás esetén a két szám különbségét tartalmazó sztringgel. Összehasonlítás esetén a visszatérési érték legyen 1 ha az első szám a kisebb, legyen -1 ha a második paraméterként kapott szám a kisebb és 0, ha egyenlő a két szám. Szorzás esetén a szorzattal térjen vissza.
Valósítsa meg azokat az algoritmusokat is, amelyek a negatív számokat is kezelik!
2. Hozzon létre prioritásos sor, ritka mátrix és alsó háromszög mátrix adatszerkezetet! **(max 4 fő)**
Az algoritmus legyen képes adatszerkezetet létrehozni, adatot tárolni, adatot kiolvasni és adatot törölni, valamint az adatszerkezetet megjeleníteni.
Ritka mátrix esetében a program legyen képes beolvasni egy csv fájlban megadott mátrixot, amiről döntse el, hogy ritka mátrix-e, ha igen akkor tárolja a létrehozott adatszerkezetben.
3. Készítse el a multihalmaz implementációját Python nyelven! **(2 fő)**
Az algoritmus legyen képes létrehozni, tárolni és megjeleníteni a multihalmaz elemeit. Valósítsa meg a multihalmaz műveleteit: unió, max, min, metszet, eleme, multiplicitás, mind közös, üres halmaz-e.
4. NxN-es mátrix algoritmusok implementálása! **(max 4 fő)**
Készítsen algoritmust NxN-es mátrix létrehozására, tárolására, megjelenítésére. Valósítsa meg a mátrix összeadás, skalárral szorzás, mátrixszorzás, mátrix transzponálás műveleteket. Készítsen rekurzív algoritmust a mátrix determinánsának kiszámítására!
5. Éhes huszár probléma megoldása **(max 4 fő)**
Éhes huszár: 4x4-es táblán bal felső sarokból jussunk el a jobb alsó sarokig lóugrásban.
(A reprezentáló táblát egészítsük ki 4 sorral és 4 oszloppal a könnyebb megvalósítás kedvéért, mint egy 2 2 soros/oszlopos margó, ahova kiugorhat a ló, persze akkor nem érvényes a lépés, de nem kell vizsgálni, hogy túl lépünk a tömbön)
6. Rossz létrán feljutás. **(max 4 fő)**
Egy f fokú létrán bizonyos fokok annyira rozogák, hogy ha rálépünk, leszakadnak. Szerencsére tudjuk, hogy melyik fokok ilyenek, hova nem szabad lépni. Egy lépéssel legfeljebb 3 fokot tudunk lépni. Adjon algoritmust, ami meghatározza, hogy a létra aljától fel tudunk-e jutni a létra legfelső fokára! (Feltehető, hogy a legfelső fokra rá szabad lépni.) Az algoritmus lépésszáma legyen $c \cdot f$, ahol c valami fix konstans. Hogyan kell módosítani az algoritmust, hogy azt is kiszámolja, hogy hányféleképpen lehet feljutni a legfelső fokra?
7. Villanykörte mátrix. **(max 4 fő)**
Egy tanteremben fel van szerelve egy $n \times n$ -es tábla, melyen n^2 villanykörte helyezkedik el. A tábla minden egyes sorához illetve oszlopához tartozik egy-egy nyomógomb, mellyel a megfelelő sorban (oszlopban) található n darab villanykörte állapotát egyszerre lehet átváltoztatni az ellenkezőjére. (Egy gombnyomásra az adott sorban illetve oszlopban égő körték elalszanak, az alvók pedig kigyulladnak.) A szünet kezdetekor az összes körte leoltott állapotban van. Szünetben a nebulók össze-vissza nyomogatják a gombokat. Hány kapcsolással tudja a tanár visszaállítani az eredeti állapotot? (A gombok egyállapotúak, azaz nem látszik rajtuk, hogy megnyomták-e őket vagy sem.)

8. Labirintus. (max 4 főt)

Egy $m \times m$ méretű táblázat mezőin lépkedünk a bal alsó sarokból a jobb felső sarokba úgy, hogy egy lépésben a táblázatban vagy felfelé vagy jobbra egyet lépünk, de van néhány „tiltott” mező, ahova nem léphetünk. Adjon egy dinamikus programozást használó eljárást, ami meghatározza, hogy hányféleképpen érhetünk célba!

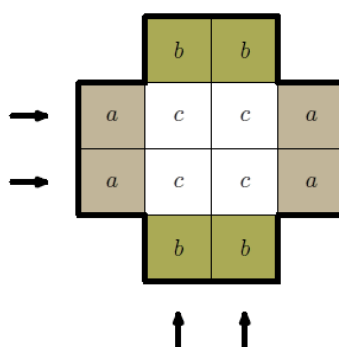
9. Búvös csiga. (max 4 fő)

Adott az ábrán látható 6×6-os tábla, rajta egy csigavonal és benne néhány szám. Írj olyan programot, amely 1-től 3-ig számokat helyez el az ábrában úgy, hogy a feladat végeztével minden sorban és oszlopban mindhárom szám pontosan egyszer szerepeljen, továbbá a bal felső sarokból elindulva, a csigavonalon haladva a számok mindig $-2-3-1-2-3-\dots$ sorrendben kövessék egymást!



10. Búvös huszonhat. (max 4 fő)

Írj olyan programot, amely a 1.15. ábrán látható kereszt alakú táblának a mezőibe elhelyezi 1-től 12-ig az egész számokat – minden cellába pontosan egy számot – úgy, hogy a tábla nyilakkal megjelölt soraiban és oszlopaiban, valamint a három különböző színű mezőcsoporton, amelyeket a, b és c betűkkel is megjelöltünk, az ott található számok összege a bűvös 26-ot adja eredményül! **(max 4 fő)**



11. Számpótlás. (max 4 fő)

Adottak az ábrán látható táblák, bennük néhány matematikai műveleti jel és néhány szám. Válasszon egy táblát!. Írj olyan programot, amely az összes lehetséges módon kitölti a táblát a 0-tól 9-ig terjedő számokkal úgy, hogy soronként és oszloponként a műveleteket sorrendben elvégezve, az egyenlőségjel után álló számokat kapjuk eredményül! Egy szám egy táblában többször is előfordulhat.

	-		+		=	8
-	■	×	■	+	■	
	+	3	-		=	1
×	■	:	■	-	■	
	+		-		=	0
=	■	=	■	=	■	
4		2		4		

(a)

	-		-		=	2
+	■	×	■	:	■	
	+	9	:		=	2
:	■	+	■	+	■	
	:		×		=	2
=	■	=	■	=	■	
2		2		2		

(b)

	-		+		=	4
-	■	+	■	×	■	
	+	5	+		=	6
×	■	-	■	+	■	
	+		-		=	6
=	■	=	■	=	■	
8		1		5		

(c)

	×		×		=	4
+	■	+	■	×	■	
	+	1	:		=	1
-	■	×	■	-	■	
	-		+		=	5
=	■	=	■	=	■	
4		9		1		

(d)

	+		-		=	5
-	■	×	■	+	■	
	×	6	:		=	4
:	■	-	■	:	■	
	+		-		=	6
=	■	=	■	=	■	
1		9		1		

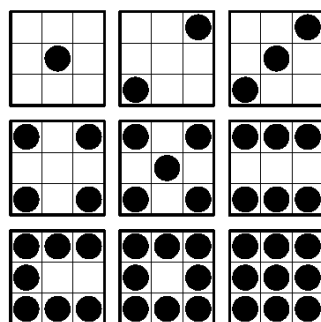
(e)

	-		+		=	6
-	■	×	■	+	■	
	×	6	-		=	9
×	■	:	■	-	■	
	+		-		=	6
=	■	=	■	=	■	
5		3		8		

(f)

12. Legyen pontos! (max 4 fő)

Adott az 1.21(a) ábrán látható kilenc féldominó. Írj olyan programot, amely elhelyezi ezeket a féldominókat az 1.21(b) ábrában úgy, hogy minden sorban, oszlopban, illetve a két átlóban annyi pont legyen, amennyi a sorok, oszlopok, illetve az átlók végén olvasható számok értéke! Az egyes elemeket elforgatni nem szabad.



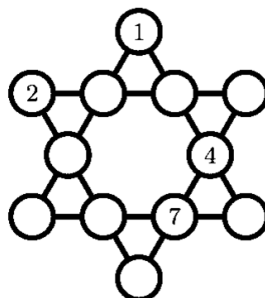
(a) A dominókészlet.

									6
									9
									4
									9
									5
									2
									5
									4
									3
									4
5	4	5	7	4	7	5	4	4	6

(b) Az üres táblázat.

13. Mágikus csillag 1. **(max 4 fő)**

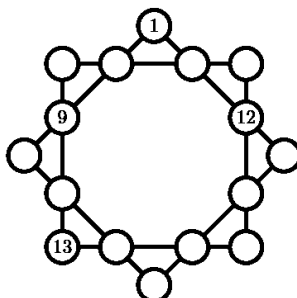
Adott az ábrán látható két mágikus csillag, bennük néhány előre beírt szám. Írj olyan programot, amely elhelyezi a csillagok üres köreibben 1-től 12-ig a még be nem írt számokat úgy, hogy az egyenes vonalak mentén elhelyezkedő számok összege 26 legyen!



(a) A vonalmenti összeg 26.

14. Mágikus csillag 2. **(max 4 fő)**

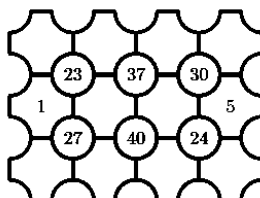
Adott az ábrán látható két mágikus csillag, bennük néhány előre beírt szám. Írj olyan programot, amely elhelyezi a csillagok üres köreibben 1-től 16-ig a még be nem írt számokat úgy, hogy az egyenes vonalak mentén elhelyezkedő számok összege 34 legyen!



(b) A vonalmenti összeg 34.

15. Számok a szomszédban. **(max 4 fő)**

Írj olyan programot, amely elhelyezi az 1-től 12-ig terjedő számokat az ábrában úgy, hogy a körökkel szomszédos mezőkbe kerülő számok összege megegyezzen a körben lévő számmal! Könnyítésül két számot előre beírtunk.



16. Varázsnégyszet. (max 4 fő)

Írj olyan programot, amely kitölti az ábra üres mezőit az 1-től 6-ig terjedő egész számokkal úgy, hogy minden sorban és oszlopban, valamint a színrel jelölt átlókban minden szám csak egyszer szerepeljen!

		6			4
	2		6		
	4				5
2				6	
		4		5	
5			3		

17. Számolás nevekkal. (max 4 fő)

A Mária, Lenke és Ilona nevekben az ábécé 10 különböző betűje szerepel. Ezeket a betűket számokká alakítva számtani műveletek végezhetők. Válasszon egyet a három kifejezés közül! Írj olyan programot, amely 0-tól 9-ig a megfelelő számokkal helyettesíti a betűket úgy, hogy az ábrán látható műveletek elvégzése után a megadott eredmények helyesek legyenek! Azonos betűk mindig azonos számokat szimbolizálnak.

$$\begin{array}{rcccccc}
 & M & Á & R & I & A \\
 + & L & E & N & K & E \\
 + & I & L & O & N & A \\
 \hline
 1 & 7 & 2 & 5 & 9 & 4
 \end{array}$$

$$\begin{array}{rcccccc}
 & M & Á & R & I & A \\
 + & L & E & N & K & E \\
 - & I & L & O & N & A \\
 \hline
 1 & 0 & 3 & 3 & 7 & 0
 \end{array}$$

$$\begin{array}{rcccccc}
 & M & Á & R & I & A \\
 - & L & E & N & K & E \\
 + & I & L & O & N & A \\
 \hline
 9 & 2 & 2 & 9 & 4
 \end{array}$$