

Programozás 1.

Beadandó feladatok 2020 tavasz

1. feladat

Írjon egy programot, amely segít eldönteni, hogy egy N. fordulós játék során ki lett a végső győztes. A program az alábbi értékeket kapja: N szám ($N \leq 10000$) ami a játék köreinek számát jelenti. Ezt követően beolvasásra kerül N sor, melyben szóközzel elválasztva megtalálható P1 és P2 játékos adott körben elért pontszáma ($1 \leq P1 \text{ pontja} \leq 1000$ és $1 \leq P2 \text{ pontja} \leq 1000$). Minden kört követően meg kell határozni, hogy az összesített pontok alapján melyik játékos vezet és mennyivel, ahogy az alábbi táblázatban is történik:

Kör	P1	P2	Ki vezet	Mennyivel vezet
1	140	82	Player 1	58
2	229	216	Player 1	13
3	319	326	Player 2	7
4	431	432	Player 2	1
5	519	522	Player 2	3

A játék végeztével az a játékos nyer, aki a játék során a legtöbb ponttal tudott vezetni. Ebben a példában tehát P1 lesz a nyertes.

Teszt Input: 5 140 82 89 134 90 110 112 106 88 90	Teszt output: P1 58
---	------------------------

2. feladat

Írjon egy programot, amelynek segítségével ki lehet számolni egy konkrét pénzösszeg kifizetéshez szükséges minimális darabszámú címletkészletet. Tehát, ha van lehetőségünk az alábbi címletekkel fizetni: 1Ft, 2Ft, 5Ft, 10Ft, 50Ft, 100Ft akkor egy bemenetként kapott összeget ezen címletek közül melyikkel és hány darabbal lehet kifizetni, úgy, hogy a lehető legkevesebb darabot használjuk fel. A program bemenetként először kap egy N számot ($1 \leq N \leq 1000$), ami a következő tesztesetek darab száma lesz, ezt követően kapja meg az N darab konkrét értéket V-t ($1 \leq V \leq 100000$) és output-ként minden értékhez meg kell határozni a pontos összeg kifizetéséhez szükséges címleteket és azok darabszámát.

Teszt Input: 3	Teszt output: 12db 100Ft
-------------------	-----------------------------

1200 500 242	5db 100Ft 2db 100Ft; 4db 10Ft; 1db 2Ft
--------------------	---

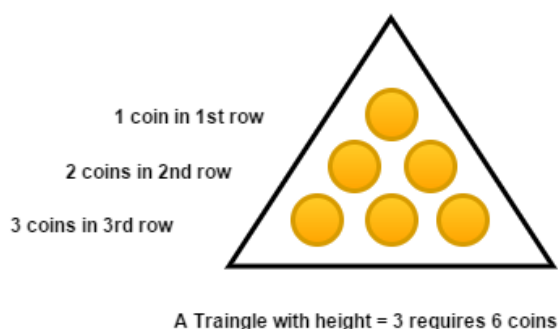
3. feladat

Írjon programot, amely infix operátorok alkalmazásával felírt matematikai kifejezéseket post fix alakúra alakít, ahogyan az a példákban is látszik. A program bementként először kap egy N számot ($1 \leq N \leq 100$), ami a következő tesztesetek darab száma lesz, ezt követően kapja meg az N darab konkrét matematikai kifejezést (maximum 400 karakter hosszú).

Teszt Input: 3 (a+(b*c)) ((a+b)*(z+x)) ((a+t)*((b+(a+c))^(c+d)))	Teszt output: abc*+ ab+zx+* at+bac++cd+^*
--	--

4. feladat

Írjon programot, amely egy adott mennyiségű labda esetén meg tudja határozni, hogy legfeljebb milyen magas piramist lehet belőle építeni (lásd a kép). A program bementként először kap egy N számot ($1 \leq N \leq 100$), ami a következő tesztesetek darab száma lesz, ezt követően kapja meg az egyes tesztekhez használatos golyók darab számát ($1 \leq DB \leq 10^9$). Amennyiben a golyók nem elegendőek a következő szint megépítéséhez, akkor azok veszendőbe mennek és a legmagasabb teljes szint magasságával kell visszatérni.



Teszt Input: 3 3 5 7	Teszt output: 2 2 3
----------------------------------	------------------------------

5. feladat

Írjon egy programot, amely megkapja N darab ($1 \leq N \leq 40$) pont (x,y) koordinátáját ($0 \leq |x| \leq 1000$ és $0 \leq |y| \leq 1000$). Két pont (a és b) távolságát számítsuk a következő képlet segítségével $\min(|a_x - b_x|, |a_y - b_y|)$. A program válasszon egy alkalmas kiinduló pontot és adja meg azt a bejárási útvonalat (az útvonalhoz a pontok sorszámát használjuk), amelynek segítségével minden

pontot pontosan egyszer érintünk és a bejárt távolság a lehető legkisebb. Ha több megoldás is van, akkor a pontok sorszáma szerint növekvő sorrend szerint történjen a bejárás.

Teszt Input #1: 2 1 2 0 0	Teszt output #1: 1, 2
Teszt Input #2: 3 3 3 0 0 0 3	Teszt output #2: 1, 3, 2 Note: 2,3,1 ugyan az a távolság, de az output az növekvő sorrendben van.

6. feladat

Írjon programot, amely bemenetként számsorozatot kap, illetve egy al-sorozatot is. A program döntse el, hogy az al-sorozat része-e a hosszabb sorozatnak. A program az alábbi bemeneteket kapja: először kap egy N számot ($1 \leq N \leq 1000$), ami a tesztesetek darab száma lesz, ezt követően kapja meg a konkrét teszt eseteket, amelyek mindig 4 sorból állnak. Az első sor a számsorozat hossza, a második sor a sorozat elemeit, a harmadik sor az al-sorozat hosszát és a negyedik sor az al-sorozat elemeit tartalmazza.

Teszt Input: 2 6 1 2 3 4 5 6 3 2 3 4 6 2 2 5 6 3 3 1 4 2 4 15	Teszt output: Yes No
--	----------------------------

7. feladat

Írjon programot, amely segít egy szobafestőnek meghatározni, hogy egy N ($1 \leq N \leq 100000$) szobás lakásban legalább, hány szobát kell kifesteni (mindig a minim számot adja meg), hogy minden szobának ugyan olyan legyen a színe. A ház szobáinak a színét a program bemenetként (sztringként) kapja, amely csak piros (R), zöld (G) és kék (B) lehet. A festő a szobák kifestéséhez szintén csak ezt a három színt használhatja, viszont ha egy falra ráfest egy másik színnel, akkor az alábbiak szerint alakul a fal színe:

- $R + B = G$
- $B + G = R$
- $G + R = B$

A bemenet első sora a teszt esetek darabszámát határozza meg, amely maximum 10 lehet.

Teszt Input: 3 3	Teszt output: 1 0
------------------------	-------------------------

RGR 3 RRR 3 RGB	2
-----------------------------	---

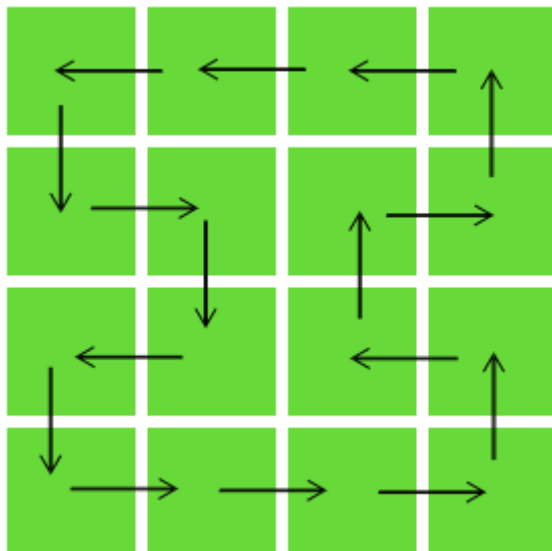
8. feladat

Írjon programot, ami segít Mórckának meghatározni, hogy pontosan mennyi időre van szüksége az összes N darab házfeladatának a megoldásához feltéve, ha Mórckica csak akkor tart szünetet amikor megoldotta a maradék feladatok legalább felét (ha N páros akkor $N/2$ után tart szünetet, ha N páratlan akkor $(N+1)/2$ után). Minden szünet pontosan B percig tart, viszont minden szünet után kétszer annyi ideig tart megoldani egy feladatot, mint korábban. Tehát ha az első szünet előtt egy feladat megoldása M perctartott, akkor az első szünet után 1 feladat $2 \cdot M$ percbe fog telni. A program bemenetként kapja a teszt esetek számát (max.: 100), majd minden sorban beolvassa az N , B , M -et ($1 \leq N, B, M \leq 10^8$).

Teszt Input: 2 9 1 2 123456 123456 123456	Teszt output: 45 131351258112
--	-------------------------------------

9. feladat

Írjon programot, amelynek segítségével eldönthető egy $N \times M$ -es méretű tábláról (N sor, M oszlop), hogy ha annak minden mezőjében elhelyezünk egy bábut, akkor azok mozgathatók-e úgy, hogy minden bábu elmozdul az eredeti helyéről, de nem lép nagyobb 1 mezőnél (lásd ábra). A program bementként először kap egy T számot ($1 \leq T \leq 5000$), ami a következő tesztesetek darab száma lesz, ezt követően kapja meg az N és M konkrét értékét ($1 \leq N, M \leq 50$).



Teszt Input: 2 3 3 4 4	Teszt output: NO YES
---------------------------------	----------------------------

10. feladat

Írjon programot, amely az alábbi módon mindig hibásan ad össze két számot. A program mindig megkapja először a teszt esetek számát ($1 \leq T \leq 100$), majd A és B összeadandó számokat ($1 \leq A, B \leq 10^9$).

	3	6
+	5	1
	8	7

Teszt Input:	Teszt output:
2	11
12 9	40
25 25	

11. Írjon egy függvényt, ami szimulálja 2 ember kő, papír, olló játékát. A függvény bemenetként megkapja, hogy hány fordulás a játék (n) és az nyer, aki többet nyer. A függvény térjen vissza a győztesrel és azzal, hogy az egyes fordulóknál ki mit választott és ki nyert. A döntetleneket újra kell játszani (nem számítanak), de ezeket a fordulókat is írja ki. Pl.: bo3 esetén egy lehetséges kimenetel.

1. játékos nyert.

1. forduló: Kő | Papír 2. játékos nyert

2. forduló: Olló | Olló Döntetlen (mj. újra játszá, de a kiírásban szerepel)

3. forduló: Olló | Papír 1. játékos nyert

4. forduló: Kő | Olló 1. játékos nyert

12. Írjon egy Függvényt, ami megtalálja a leghosszabb páros/páratlan vagy páratlan/páros alternáló sorozatot egy számban és visszatér ezzel a „részszámmal” és ennek hosszával. Abban az esetben, ha több ilyen is van, akkor az első ilyenrel térjen vissza. Pl a 34545488 sorozatban a 345454 páratlan páros alternáló és 6 hosszú.

13. Írjon egy függvényt, aminek a bemenete a csapatok száma és elkészíti egy random lejátszott sportesemény ágrajzát és ezt írassa ki egy txt-be. A txt-be kiírt ágrajznak legyen valamilyen stílusa. Pl. 4 csapat esetén.

```
Team1---
      |---> Team2---
Team2---
      |
      |---> Team2
Team3---
      |
      |---> Team3---
Team4---
```

14. Készítsen egy függvényt, ami képes felismerni, hogy a bemenetként kapott string az Morse kód vagy szöveg és ezt átalakítja a másikká. (Morse → Szöveg vagy Szöveg → Morse)
15. Írjon egy programot, ami a standard bemenetből kap meg név életkor és magasságot tartalmazó rendezett hármassokat (tuple) ((név, életkor, magasság)) nulla végjelig. Majd sorba rendezi őket. Elsőnek név szerint abc rendben, azonos név esetén az idősebbek vannak hamarabb, ha a koruk is megegyezik, akkor az alacsonyabbak kerülnek előrébb. Végül visszatér a sorba rendezett hármassokkal úgy, hogy mindegyik új sorban legyen.
16. Írj egy programot, ami egy input.txt-t kap bemenetként és visszatér, az output.txt-vel. A függvény helyettesítse az input szöveg betűit a betűk sorszámaival (a-1,b-2,c-3...) és írja ezt be az output-ba. A számok legyenek szóközzel elválasztva a szavak között legyenek csillagok és a többi karakter maradjon változatlan. Pl.: Hello World! → 8 4 12 12 15 * 23 15 18 12 4 !
17. Készíts egy függvényt, aminek az egyik bemenete egy természetes szám (n) és egy karakter (pl.: #) a másik. A függvény készítsen el egy n darab fokból álló lépcsőt úgy, hogy a megadott karakter legyen a lépcső „váza”. pl: lépcső(4,'#')

```

_
_#
_##
_###

```

18. Írjon függvényt, aminek a bemenete egy szám és el tudja dönteni, hogy a szám tartalmaz-e töréspontot. Ha igen, akkor térjen vissza a számmal, ami el van választva a töréspont mentén ellenkező esetben térjen vissza False-szal. Egy számban akkor van töréspont, ha ez első n számjegy összege megegyezik a maradék számjegyek összegével. Pl.: 35190 → 351|90; 351 → False
19. Készítsen programot, ami elsőnek megkapja, hogy mekkora legyen a rajztábla mérete (n x n), majd a standard bemenetből kapjon utasításokat, hogy merre rajzoljon. Minden utasítás után jelenítse meg a változást. (mint egy „nagyon light” comenius logo) Az utasítások: adott pontra lépés (akkor nem rajzol), jelenlegi rácponttól megadott hosszú egyenesek húzása fel, le, jobbra vagy balra.
20. Egy számot Harshad-számnak nevezzük, ha a szám osztható a számjegyei összegével. Írjon függvényt, ami el tudja dönteni egy számról, hogy Harshad-szám-e. A számot standard bemenetből kapja meg. Pl a 18 osztható 1+8 = 9-cel, így Harshad-számnak, de a 11 nem osztható 1+1 = 2-vel, így nem Harshad-számnak.

21. feladat: Írjon függvényt, amely a paraméterként átadott 3 számra (n, k, r) meghatározza n összegének ábrázolási módjait úgy, hogy: $A + B = n$, ahol $k \leq A \leq B \leq r$.

Input: n = 6, k = 2, r = 4

Output: 2

Magyarázat: Csak két módja van 6 reprezentálásának úgy, hogy $6 = A + B$, ahol $2 \leq A \leq B \leq 4$: $6 = 2 + 4$ és $6 = 3 + 3$.

22. feladat: Írjon függvényt, amely összead két törtet! Az eredményül kapott törtet egyszerűsítse is!

Input: (2, 3) és (4, 6) (Amelyek 2/3-nak és 4/6-nak felelnek meg.)

Output: (4, 3) (Amely 4/3-nak felel meg.)

23. feladat: Adott, egy egész számokat tartalmazó a vektor. Legyen a_j rákövetkezője a_i -nek, ha $i < j$ és $a_i < a_j$. Határozza meg a rákövetkezők számát minden elem esetén és adja vissza az összegét ezeknek a számoknak mod $(10^9 + 7)$.

Bemenet: $a = [1, 2, 3, 4, 5]$,

Kimenet: 10

Magyarázat:

1-es rákövetkezői: 2, 3, 4 és 5 ezek darabszáma 4.

2-es rákövetkezői: 3, 4 és 5 ezek darab száma 3.

3-as rákövetkezői: 4 és 5 ezek darabszáma 2.

4-es rákövetkezői: 5 ezek darabszáma 1.

Eredmény: $4 + 3 + 2 + 1 = 10 \bmod 10^9 + 7$

24. feladat: Néhány ember sorban áll egy parkban. Adott egy vektor, amely a parkban sorban álló emberek magasságát reprezentálja. Az emberek között fák lehetnek (-1 értékkel jelölve), amelyeket nem lehet mozgatni. Írjon függvényt, amely sorba rendezi az embereket sorrendben magasságuk alapján növekvő sorrendbe! Figyeljen arra, hogy a fák nem mozgathatóak.

Bemenet: $[-1, 150, 190, 170, -1, -1, 160, 180]$

Kimenet: $[-1, 150, 160, 170, -1, -1, 180, 190]$

25. feladat: Hozzon létre egy felhasználó által definiált dimenziójú mátrixot ($n \times m$)! Írjon függvényt, amely True értéket ad vissza, ha a mátrix minden sora és oszlopa rendezett! Ügyeljen arra, hogy a sorok növekvő és csökkenő sorrendben rendezettek is lehetnek!

Input: $n = 3, m = 4$, mátrix:

5	3	1	1
8	2	1	0
12	2	0	0

Output: True

26. feladat: Olvasson be két sztringet, a következő formátumban: "hh:mm:ss". Az egyik sztring (part) egy videóból megtekintett időtartamot jelöli, a másik (total) pedig a videó teljes időtartamát. Írjon függvényt, amely e két sztring esetén visszaadja, hogy a teljes videó mekkora részét sikerült már megnézni.

Input: part = "02:20:00"

total = "07:00:00"

Output: [1, 3] (A teljes videó 1/3 részét sikerült már megnézni.)

27. feladat: Írjon függvényt, amely paraméterként átadott mondatot átalakít úgy, hogy minden egymást követő betű után beilleszt egy kötőjelet!

Bemenet: "aba caba"

Kimenet: "a-b-a c-a-b-a"

28. feladat: Adott egy A vektor, amely 0 és 1 értékeket tartalmaz.

Input: [0, 1, 1, 1, 0]

Írjon függvényt, amely törli a vektor tartalmát. DE a vektorból 1 lépésben törölni csak 1 folytonos, azonos elemeket/elemet tartalmazó számsorozatot lehet! Hány lépést igényel, az összes elem törlése a vektorból? (Több megoldás is lehetséges.)

Magyarázat:

Az input vektor esetén az egyik lehetséges megoldás:

1. lépésben a 0. indexen lévő 0 elem törlése után a vektor tartalma: [1,1,1,0]
2. lépésben az így eredményül kapott vektor 1-es elemeinek törlése következik, az egy folytonos sorozat a vektorban. Az eredményül kapott vektor: [0]
3. lépésben az utolsó 0-s elem törlése a vektorból. Eredmény: []

Másik lehetséges megoldás:

- 1.lépésben a folytonos 1-eseket tartalmazó sorozat törlésére kerül sor. Az eredmény: [0, 0]
2. lépésben a [0, 0] vektor 2 darab 0-ból álló folytonos sorozatának törlése következik. Eredmény: []

Output: 3 vagy 2 (Elég egyet visszaadni.)

29. feladat: Írjon függvényt, amely egy titkosított sztringet dekódol.

A visszaféjtési szabály: $k[\text{encoded_string}]$,

ahol a szögletes zárójelek között megadott `encoded_string` egymás után k -szor ismétlődik.

Megjegyzés: k garantáltan pozitív egész szám

1. példa:
 - a. Input: "4[ab]"
 - b. Output: "abababab"
2. példa:
 - a. Input: "2[b3[a]]"
 - b. Output: "baaabaaa"
3. példa:
 - a. "z1[y]zzz2[abc]"
 - b. "zyzzzabcabc"

30. feladat: Hozzon létre egy karaktereket tartalmazó mátrixot, melynek dimenzióját ($n \times m$, ahol $m \neq n$) a felhasználótól kérje be. Írjon függvényt, amely a mátrixot kiegészíti egy csillagokból álló kerettel.

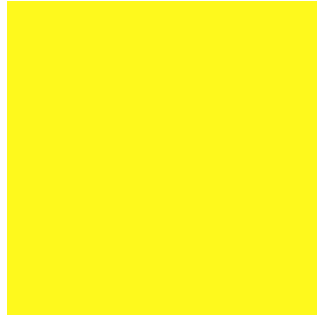
Input: ["abc",
"ded"]

Output: ["*****",
"*abc*",
"*ded*",
"*****"]

31. Készítsen programot amelyben egy dictionary-n belül tárol egy felhasználó által bekért méretű kék, egy piros és egy sárga képnek megfelelő mátrixot. Készítsen egy alprogramot, amely a következő bekért stringekre megjeleníti a neki megfelelő képet: sárga, lila, zöld, fehér. FONTOS! A megjelenítendő képeket a dictionary-ben szereplő képek kombinációjából állítsa elő

Bemenet: 200, 200, sárga

Kimenet:



32. Készítsen egy programot amely egy XO játék eredményéről eldönti, hogy X vagy O nyert, vagy esetleg Egyenlő a játék eredménye (X-eket nagy X, O-kat nagy O, E pedig az üres helyeket jelöli).

Példák: ["X", "O", "X",
"O", "X", "O",
"O", "X", "X"] -> "X" ["O", "O", "O",
"O", "X", "X",
"E", "X", "X"] -> "O" ["X", "X", "O",
"O", "O", "X",
"X", "X", "O"] -> "Egyenlo"

33. Készítsen programot, amely egy bemeneti string-ről eldönti, hogy az helyes e-mail cím-e vagy sem. A helyes cím alakja a következő: "azonosito@domain".

- azonosito:
 - o nem tartalmazhat szóközőket és a következő karaktereket: "_?!*;;" (vigyázat, pontot tartalmazhat)
- domain:
 - o a vége megfelel az engedélyezett formáknak amely egy bővíthető listában van tárolva. A lista: [".com", ".hu"]
 - o az eleje nem kezdődhet nagy betűvel és nem tartalmazhat szóközőket és a következő karaktereket: "_?!*;;"

Példák: asd.asd@vlaami.hu -> valid

Asd?@valami.com -> nem valid

asdasd@valami.gb -> nem valid

asd@Valami.com -> nem valid

[asd\(kukac\)valami.hu](mailto:asd(kukac)valami.hu) -> nem valid

34. Készítsen programot, amely eldönti egy bemeneti string-ről, hogy az helyes RGB(A) CSS szín-e.

A feltételek a következők: a string úgy kezdődik, hogy rgb vagy rgba. Ha rgb akkor 3 szám szerepel a zárójelek között, vesszővel elválasztva [0,255] intervallumon. Ha rgba akkor 4 szám, az első 3 a [0,255] intervallumon a negyedik pedig [0,1] intervallumon.

Példák: "rgb(0,0,0)" -> Igaz

"rgb(0,,0)" -> Hamis

“rgb(255,256,255)” -> Hamis
“rgba(0,0,0,0.1234567)” -> Igaz

35. Készítsen programot amely egy listát kap és megmutatja, hogy melyik elem hányszor szerepel benne. A könnyen áttekinthető vizualizációhoz használjon matplotlib-et. Ezen felül adja vissza a program azt az elemet amely többségben szerepel ha van ilyen. Többség, mikor az adott elem többször szerepel, mint: $n/2$ (ahol n a lista hossza)
Bemenet: [7,5,3,3,5,8,3,3,3,4]
Kimenet: 3 (6-szor szerepel egy 11 hosszú listában)
36. Készítsen programot ami kap egy sztringet és azt kódolja az alapján, hogy mely betűkből hány darab következik. A megoldáshoz használjon dictionary-t! (Feltételezhetjük, hogy a sztring amit kapunk csak nagy, az angol ABC betűit tartalmazza)
Bemenet: “AAAABBBCCDAAA”
Kimenet: “4A3B2C1D3A”
37. Készítsen egy programot, amely a listában szereplő 0 értékeket a végére tolja, és a nem nulla elemeket növekvő sorrendbe rendezi. A program vegye ki a listából azokat a nem nulla elemeket is amelyek kisebbek, mint az eredeti listában szereplő értékek átlaga.
Bemenet: [7,10,0,9,11,0,17]
Kimenet: [9,10,11,17,0,0]
38. Készítsen programot, amely kap 2 számot és kiszámolja ezek fordítottjának az összegét, majd annak kiírja a szintén fordítottját. Megjegyzés: nem minden számnak a fordítottja egyedi, pl. 31 a fordítottja a 13,130,1300,... összes számnak, szóval a 0-kat vágjuk le a végéről ilyen esetben.
Bemenet: 305, 794
Kimenet: Fordított = 503, 497 Összeg = 1000 FordítottÖsszeg = 1
39. Készítsen programot, ami összeadja egy szám számjegyeinek az összegét, majd ezt addig ismétli, ameddig nem egy számjegyű lesz az összeg. A program számolja ki az összes páratlan prím számjegy összegét, amit felhasznált a futtatás során.
Bemenet: 7548
Kimenet: Összeg1 = 7 + 5 + 4 + 8 = 24 Összeg2 = 2 + 4 = 6 PrimÖsszeg = 7 + 5 = 12
40. Készítsen egy programot, ami választ egy véletlen számot [1,100] intervallumon, majd lehetővé teszi a felhasználó számára, hogy kitalálja melyik számot generálta a program. Ha a felhasználó nem találja el a számot akkor segítsen neki és mondja meg, hogy a próbálkozásánál kisebb vagy nagyobb a generált szám. Készítsen a programhoz egy szöveges menüt amiben a felhasználó a billentyűzettel beírt betűkkel tud az egyes menüpontokban lépni. Menüpontok: New Game, Score, Difficulty, Quit. A Score menüpontra eléri a felhasználó az eddigi összes megnyert játékainak a számát (ehhez használjon filekezelést). A Difficulty-n belül állítani lehet 3 nehézségi szintet, amelyek tetszőleges próbálkozások számát jelölik. (Pl. Easy – 10-szer próbálkozhat a felhasználó, hogy eltalálja a számot). A Quit menu a játék bezárását eredményezi.
41. Írjon programot, amely ‘end’ végjelig kér be stringeket. A program írja ki a bekért stringek átlagos magánhangzósámát, valamint azokat a stringeket, amelyekben az átlagnál több magánhangzó szerepel.

Példa bemenet:

kecske, dio, fa, elefant, end

Példa kimenet:

Átlagos magánhangzós szám: 2

Szavak: elefant

42. Írjon programot, amely bekér három szót szóközzel elválasztva a felhasználótól. Egy függvény írja ki azt a szót, amelynek az angol abc-ben a legelől, illetve leghátul lévő betűi a legkisebb távolságra vannak egymástól.

Pl: alma: legelől az abc-ben: a -> 1. helyen, leghátul m -> 13. helyen, tehát a távolság 12

Példa bemenet:

alma pohar ablak

Példa kimenet:

ablak

43. Írjon programot, amely római számokat tud átkonvertálni arab számokra. A program addig kérje a római számokat, amíg azok összege nem éri el a 1000-et. A program végül írja ki soronként minden római és arab számot.

Példa bemenet:

CLXII

DCC

CD

Példa kimenet:

CLXII -> 162

DCC -> 700

CD -> 400

44. Írjon programot, amely bekér egy n számot, majd pedig n alkalommal soronként számokat szóközzel elválasztva. A program számolja a számsorban szereplő pozitív, illetve negatív számok szórását külön-külön. A program írjon ki egy üzenetet a felhasználónak, ha nem számolható valamelyik szórás.

Példa bemenet: n = 2

2 15 -3 4 -9

2 5 9

Példa kimenet:

Pozitív szórás: 7, negatív szórás: 3

Pozitív szórás: 2.86, negatív szórás nem számolható

45. Írjon programot, amely parancssori argumentumként bekér egy n számot és egy stringet. A programot az n -edik betűig minden betűt egyel többször írjon ki az előzőnél, az n -edik betű után pedig minden betűt egyel kevesebbszer az előzőnél.

Példa bemenetek:

3 kacsza
4 almafa

Példa kimenetek:

kaaccsssa
allmmaaaafffaa

46. Írjon programot, amely parancssori argumentumként bekér n darab számot, és $n-1$ darab műveleti jelet. A program tárolja el a számokat és műveleti jeleket egy-egy listában, majd írja ki a műveletek által generált eredményt a számsorozaton, szem előtt tartva a helyes műveleti sorrendet.

Példa bemenet:

4 2 3 1 + - *

Példa kimenet:

$4 + 2 - 3 * 1 = 3$

47. Írjon programot, amely először beolvas egy n számot, amely a beolvasandó sorok számát jelöli. Majd ezt követően n alkalommal olvasson be a program egy-egy SMS üzenet adatait. Egy sor tartalmazzon egy telefonszámot, a küldés időpontját (óra, perc) és az üzenet szövegét. A program írja ki egy txt állományba telefonszámok szerint csoportosítva a szöveges üzeneteket időpont szerint sorba rendezve.

Példa bemenet:

4
123456789 11 45 Szia, mikor jössz?
456721456 16 25 Kész az ebéd!
123456789 09 24 El kell menni a gyerekért az uszodába.
302145689 13 11 Hozd el a mintás pulcsimat!

Példa kimenet:

123456789
1. 09:24 El kell menni a gyerekért az uszodába.
2. 13:11 Szia, mikor jössz?
456721456
1. 16:25 Kész az ebéd!
302145689
1. 13:11 Hozd el a mintás pulcsimat!

48. Írjon programot, amely létrehoz egy random vektort, valamint bekér a felhasználótól egy kezdőindexet, egy végindexet és egy + vagy - jelet. A program + jel esetén rendezze a vektor elemeit a bekért indexek között növekvő sorrendben, - jel esetén csökkenő sorrendben. A program szűrjön be két | jelet a rendezett részsorozat elé, és mögé. A program kérje be a

felhasználótól a vektor méretét, valamint a random intervallum kezdő-, illetve végértékeit.

Példa bemenet: 10 elemű vektor 1 – 20 közötti értékekkel

vektor = [2,15,3,16,8,7,11,3,19,2]

3 8 +

Példa kimenet:

vektor = [2,15,3,|,3,7,8,11,16,19,|,2]

49. Írjon programot, amely létrehoz egy m egy $n \times m$ -es mátrixot ($n > 2$, $m > 2$). A program keresse meg a mátrixban az összes olyan 2×2 -es részmátrixot, amelynek a determinánsa nagyobb, mint 10. Az eredmény mátrixokat írja ki egy txt állományba.

Példa bemenet:

[4, 4, 8,

2, 6, 4,

1, 6, 6]

Példa kimenet:

[6, 4, [4, 4,

6, 6] 2, 6]

50. Írjon programot, amely először bekér egy n értéket, ami egy útszakast hosszát jelöli, és egy m értéket, ami egy sebességkorlát lesz. Majd soronként, 'end' végjelig beolvas két időpontot óra perc másodperc formátumban, szóközzel elválasztva, és egy rendszámot. Az első időpont az adott útszakaszra való belépés időpontja, a második pedig a kilépés időpontja az ellenőrzött útról. A program írja ki azoknak az autóknak a rendszámokat és sebességét, amelyek átlagsebessége meghaladta a megengedett m értékkel jelölt sebességet az ellenőrzött úton.

Példa bemenet:

500 50

11 22 35 11 23 05 ABC123

14 08 11 14 08 24 FGT546

09 02 45 09 03 55 VGT478

end

Példa kimenet:

ABC123 60km/h

FGT546 138.46km/h