

1) "Szép tömb"

Egy tömböt akkor mondunk szépnek, ha csak azonos típusú elemeket tartalmaz. Írj egy `szepeTomb` nevű függvényt, amely egy tömböt vár paraméterben, és visszaadja, hogy ez a tömb azonos típusú elemeket tárol-e! Ha a paraméterben kapott érték nem tömb típusú, akkor hamissal térjen vissza a függvény!

Példa:

```
Input: ['alma', 'korte', 'szilva', 'barack', 'palinka']  
Return: true
```

```
Input: [10, 20, 30, 40, '50', 60, 70]  
Return: false
```

```
Input: []  
Return: true
```

```
Input: 'YEE HAW!'  
Return: false
```

2) Tömbelemek hatványozása

Írj egy `hatvanyoz` nevű függvényt, amely két paramétert vár: egy számokat tároló tömböt és egy számot! A függvény emelje a tömbben lévő számokat a második paraméterben kapott szám által meghatározott hatványukra, majd térjen vissza a hatványozás után kapott tömbbel!

Ha valamelyik paraméter típusa nem megfelelő, akkor a függvény egy üres tömbbel térjen vissza!

Példa:

```
Input: [1, 4, 7, 2, 5], 3  
Return: [1, 64, 343, 8, 125]
```

```
Input: 'buvarszivattyu', 5  
Return: []
```

3) Full stack

Jónás, a csokigyáros el szeretné készíteni a vállalkozásának a weboldalát, ezért webfejlesztőket keres. A munkára beadott jelentkezések elbírálásakor Jónás előnyben részesíti azokat a fejlesztőket, akik a front-endhez és a back-endhez is értenek ("full stack fejlesztők").

Írj egy `fullStack` nevű függvényt, amely két szöveget kap paraméterül: az első szövegben a front-end, míg a második szövegben a back-end fejlesztők nevei szerepelnek pontosvesszőkkel elválasztva! A függvény adja vissza egy tömbben azoknak a nevét, akik a front-end és back-end fejlesztők között is szerepelnek!

Példa:

```
Input: 'Zsiros B. Odon;Koaxk Abel;Heu Reka;Riz Otto', 'Riz Otto;Rabsz Olga;Trab Antal;Koaxk Abel;Winch Eszter'
```

```
Return: ['Koaxk Abel', 'Riz Otto']
```

```
Input: 'Citad Ella;Techno Kolos', 'Git Aron;Teask Anna'
```

```
Return: []
```

4) Valaki éppen gépel...

Jani egy chata alkalmazást fejleszt JavaScriptben. A chatben mindig kiírunk egy üzenetet arról, ha egy vagy több felhasználó éppen gépel. Írj egy `gepeles` nevű függvényt, amely egy szöveget kap paraméterül! Ez a szöveg pontosvesszőkkel elválasztva tartalmazza az éppen gépelő felhasználóknak a nevét. Figyelj arra, hogy néhány név elé és/vagy után néhány fölösleges whitespace karakter került, amiket először el kell távolítani!

- Amennyiben a paraméter típusa nem megfelelő vagy a paraméterben az üres string érkezik, akkor a függvény térjen vissza a `HIBA!` szöveggel!
- Ha pontosan 1 ember gépel, akkor térjünk vissza a `{username} éppen gépel...` szöveggel, ahol `{username}` a gépelő felhasználó neve!
- Ha pontosan 2 ember gépel, akkor térjünk vissza a `{username1} es {username2} éppen gépel...` szöveggel, ahol `{username1}` az első, míg `{username2}` a második felhasználó neve a pontosvesszőkkel elválasztott adatok közül!
- Ha 2-nél több ember gépel, akkor térjünk vissza a `{username1}, {username2} es {N} további felhasználó éppen gépel...` szöveggel, ahol `{username1}` és `{username2}` ugyanaz, mint az előző vázlatpontban, `{N}` pedig a többi gépelő felhasználónak a száma!

Példa:

```
Input: 'SuTi'
```

```
Return: 'SuTi éppen gépel...'
```

```
Input: 'szte4k;catman6      ;      Sziklas      ;      Aerosol;cservZenberg'
```

```
Return: 'szte4k, catman6 es 3 további felhasználó éppen gépel...'
```

```
Input: ''
```

```
Return: 'HIBA!'
```

5) Ötöslottó

A játék során 90 számból húznak ki 5 darab számot véletlenszerűen, és ezek közül kell a játékosoknak minél többet eltalálniuk. Erzsi néni ezen a héten a véletlenre bízta az öt szám bejelölését a lottószelvényen.

Írj egy paraméter nélküli `otoslotto` függvényt, amely generál 5 darab véletlenszámot 1 és 90 között, és visszaadja ezeket egy tömbben, **növekvő sorrendben!** Ügyelj arra, hogy a számok között ne legyen ismétlődés (minden számot csak egyszer jelölhetünk meg)!

6) E-mail cím generátor

Egy levelező szolgáltatással saját e-mail fiókot hozhatunk létre. A fiók létrehozásakor megadjuk a teljes nevünket és a születési évünket, majd ezekből a rendszer generál egy javasolt e-mail címet a következőképpen:

- vesszük a teljes névben szereplő összes szóközzel elválasztott névnek az első 3 betűjét, csupa kisbetűvel (ha egy ilyen string rövidebb 3 betűnél, akkor a teljes stringet vesszük)
- ezután vesszük a születési év utolsó számjegyét
- végül az így kapott azonosítóhoz hozzáfűzzük a `@duck.com` szöveget.

Írj egy `emailCimetGeneral` nevű függvényt, amely két paramétert vár: a teljes nevet és a születési évet, és visszatér a fenti szabályok alapján legenerált e-mail címmel! A születési évet ne legyen kötelező megadni, alapértéke legyen 1970! Hibakezeléssel nem kell foglalkoznod, feltesszük, hogy a paraméterek helyesek.

Példa:

```
Input: 'Richard Paul Astley', 1966
Return: 'ricpauast6@duck.com'
```

```
Input: 'Citad Ella'
Return: 'citell0@duck.com'
```

```
Input: 'Pablo Diego Jose Francisco de Paula Juan Nepomuceno Maria de los Remedios
Cipriano de la Santisima Trinidad Ruiz y Picasso', 1881
Return: 'pabdiejosfradepaujuanepmardelosremcipcodelasantriruiypic1@duck.com'
```

7) Számok csoportosítása

Írj egy `szamokatCsoportosit` nevű függvényt, amely egy olyan szöveget kap paraméterül, amelyben pontosvesszőkkel elválasztott számok szerepelnek! A függvény csoportosítsa a szövegben szereplő számokat "pozitív", "nulla" és "negatív" kategóriába! A csoportosítás eredményét egy object-ben kell visszaadni, a példában látható formátumban.

Példa:

```
Input: '7;-4;-1;0;2;5;-8'
Return: {pozitiv: [7, 2, 5], nulla: [0], negativ: [-4, -1, -8]}

Input: '-1;-2;-3;-4;-5'
Return: {pozitiv: [], nulla: [], negativ: [-1, -2, -3, -4, -5]}
```

8) Kérdőív

Az egyik egyetemen egy online kérdőívben kérdezik meg az informatikusokat, hogy melyik tantárgy számukra a legnehezebb. A kérdésre adott válaszokat elmentik, majd ebből készítenek statisztikát.

Írj egy `valaszokatOsszesit` nevű függvényt, amely egy tantárgyak nevét (stringek) tároló tömböt kap paraméterül! A függvény térjen vissza egy object-tel, amelyben tárold el, hogy melyik tantárgyra hány szavazat érkezett! A property-k a tantárgyak nevei, a hozzájuk tartozó értékek pedig a rájuk érkezett szavazatok száma.

A tantárgyak nevében a kis- és nagybetűket ne különböztess meg (tehát pl. `Kalkulus III` és `kalkulus iii` ugyanaz a tárgy), az eredmény object-ben minden tárgy neve csupa kisbetűvel szerepeljen!

Példa:

```
Input:
[
  'Indiszkret Matematika', 'Kalkulus III', 'Tavolito es valosagos szamitasok',
  'kalkulus iii', 'tavolito es valosagos szamitasok',
  'TAVOLITO ES VALOSAGOS SZAMITASOK', 'kalkulus iii', 'Kalkulus III',
  'Indiszkret Matematika', 'tAvOlItO Es vAlOsAgOs sZaMiTaSoK',
  'Kaveautomatakat es informális nyelvek', 'tavolito es valosagos szamitasok'
]

Return: {'indiszkret matematika': 2, 'kalkulus iii': 4, 'tavolito es valosagos szamitasok': 5, 'kaveautomatakat es informális nyelvek': 1}
```