

JavaScript Felmérés II.

A felmérő készítése során a megoldas.js fájlba dolgozzon.

A megoldáshoz hozza létre az alábbi függvényeket. Kérlek ügyelj a nevük pontos megadására, mivel a JavaScript nyelv ugye kis és nagybetű érzékeny!

hosszEllenor()

tajSzamEllenor()

tombTerjedelem()

legidosebbDolgozo()

fizetesEmeles()

A paraméterezésüket a feladatok leírásánál találod.

Valamint a függvényeket, vagy az elkészítésük előtt vagy utána folyamatábrával is próbáld meg elkészíteni, és ne feledd ezeket is a végső (feltöltött) fájlhoz csatolni! (Ajánlott)

A megoldáshoz csatolandó (egy fájlba tömörítve):

- **megoldas.js fájl** (mindenképp csatold!!!)
- **elkészített folyamatábrák** (ha van és volt rá idő)
- **A folyamatábrákat később is beküldheted a szokásos e-mail címre** (külön beszélünk róla)

A fájl első sorába megjegyzésként illeszd be a saját nevedet és csoportodat!

1. feladat: hosszEllenor() – Bemeneti paraméterek száma: 1 (szöveg)

Készítsen egy függvényt, amely ellenőrzi egy bemeneti karakterlánc hosszát, és amennyiben az nem megfelelő (nincs legalább 8 karakter) úgy a függvény hamis értékkel tér vissza, ellenkező esetben az függvény visszatérési értéke természetesen igaz lesz.

Kipróbálásra: „teszt” esetleg „feladateszt”

Visszatérési érték: false illetve true

2. feladat: tajSzamEllenor () – Bemeneti paraméterek száma: 1 (számsor szöveggént)

Készítsen egy függvényt mely szöveges bemeneti paraméterrel dolgozik, mely számok sorozata, összesen 9 karakter hosszú.

Ellenőrzés folyamata:

Ellenőrizni kell a karakter mennyiségét

(a)Majd összeadni a páros helyen lévő elemeket(4db), ezt megszorozni héttel

(b)Ezután összeadni a páratlan helyen lévő elemeket(4db), ezt megszorozni hárommal

Megjegyzés: a számsor utolsó elemét csak a következő lépésnél használjuk fel.

A feltétel igaz, ha a páros és páratlan számok végeredménye 10-zel való maradékos osztás után a tajszám utolsó elemének értékét adja vissza! **Azaz: $(a+b)\%10==tajszám[8]$**

Ellenkező esetben nem

Kipróbálásra: `tajSzamEllenor(040655330)` esetleg `tajSzamEllenor(111111111)`

Visszatérési érték: true illetve false

További igaz visszatérésű elemek: 037687210, 019536646

3. feladat: tombTerjedelem() – Bemeneti paraméterek száma: 1 (tömb)

Határozza meg egy vagy több függvény segítségével, egy adott tömb elemeinek terjedelmét. Terjedelem: a tömb legkisebb és a legnagyobb eleme közötti különbség.

A függvényt úgy készítse el, hogy bármilyen tömb paraméter esetén helyes megoldás adjon.

Kipróbálásra:

```
let vizsgaltTomb= [3, 5, 10, 16, 9];
```

```
tombTerjedelem(vizsgaltTomb);
```

Visszatérési érték: 13

4. feladat: legidosebbDolgozo() – Bemeneti paraméterek száma: 1 (objektum)

Készíts el egy függvényt, amelynek megadsz egy objektumot, melyben biztosan tartozik egy [kor] tulajdonság minden elemhez, abból kikeresse a „legidősebb” elemet, és visszaadja az objektumban található elemnek az indexét.

A függvényt úgy készítse el, hogy bármilyen [kor] attribútummal rendelkező paraméter esetén helyes megoldás adjon.

Kipróbálásra: A feladathoz mellékelt objektum

Visszatérési érték: *A legidősebb objektum elem indexe (jelenleg: 3)*

5. feladat: fizetesEmeles() – Bemeneti paraméterek száma: 1 (objektum)

Készíts egy függvényt, mely paraméterként bekér egy objektumot, melyben biztosan tartozik egy [fizetes] tulajdonság minden elemhez, melyben meghatározza mennyi az átlagfizetés.

Majd az átlag fizetés alatti emberek fizetését emeli 10%-kal. Ezekután visszaadja a módosított objektumot a programnak.

A függvényt úgy készítse el, hogy bármilyen [fizetes] attribútummal rendelkező paraméter esetén helyes megoldás adjon.

Kipróbálásra: A feladathoz mellékelt objektum

Visszatérési érték: *A módosított objektum*