

Készítse el az alábbi feladatokat TypeScriptben, ügyeljen arra, hogy a tesztesetek jól lefussanak, a változók típusa és a visszatérési érték megfelelő legyen, valamint a tiszta kód elveinek használatára is törekedjen a kód készítése során:

1. feladat - SZEKVENCIA

Készíts egy függvényt, ami egy paraméterként megkapott sebességből és idő értékből megadja az adott idő alatt megtett utat, az út a sebesség és az idő szorzata

Függvény neve: *MegtettUt()*

Paraméter(ek): *sebesseg (szám típusú), idő (szám típusú)*

Visszatérési értéke: *szám érték*

2. feladat - SZELEKCIÓ

Készíts egy függvényt, ami paraméterként 3 nap várható átlaghőmérsékletét kapja meg, ebből a 3 értékből el kell döntenie a kiadott hőségriadó szintjét.

- **1. fok (figyelmeztető jelzés):** ha a napi középhőmérséklet várhatóan egy napig eléri vagy meghaladja a 25°C-ot. Visszatérési érték: **1**
- **2. fok (riasztás):** ha a napi középhőmérséklet várhatóan legalább három egymást követő napon eléri vagy meghaladja a 25°C-ot. Visszatérési érték: **2**
- **3. fok (riadójelzés):** ha a napi középhőmérséklet várhatóan legalább három egymást követő napon eléri vagy meghaladja a 27°C-ot. Visszatérési érték: **3**
- **0. fok** minden egyéb esetben, NINCS hőségriadó! Visszatérési érték: **0**

Függvény neve: *HosegriadoSzint()*

Paraméter(ek): *nap1(szám típusú), nap2(szám típusú), nap3(szám típusú)*

Visszatérési értéke: *szám érték*

3. feladat - ITERÁCIÓ

Készíts egy függvényt, ami egy paraméterként megkapott osztó értékből és egy vizsgált szám típusú tömbből, megmondja, mennyi azoknak a számoknak a mennyisége, amiket az adott osztó maradék nélkül eloszt.

Függvény neve: *OszthatoSzamok()*

Paraméter(ek): *osztó(szám típusú), vizsgaltTomb(szám típusú tömb)*

Visszatérési értéke: *szám érték*

4. feladat - TUPLE

Készíts egy függvényt, **TUPLE** felhasználásával, érettségi feladat kiértékelésére, az érettségi 5 modulból áll, a modulok eredményeit egy szám típusú tömbben kapja meg a függvény, ebből meghatározhatja a tanuló által elért pontot.

A szerezhető érdemjegyek ponthatára a következő:

0-tól	39-ig	1
40-től	59-ig	2
60-tól	79-ig	3
80-tól	119-ig	4
120-tól	150-ig	5

Függvény neve: *Erettsegi()*

Paraméter(ek): *pontok (szám típusú tömb)*

Visszatérési értéke: *TUPLE: összpontszám(szám érték), jegy(szám érték)*

5. feladat - SZÖVEGKEZELÉS

Készíts egy függvényt, ami egy szöveget átalakít, úgy, hogy a benne lévő bizonyos betűket átalakítja számokká.

A kicserélendő betűk listája a következő

- i és I betű helyett 1-es
- o és O betű helyett 0-s
- a és A betű helyett 4-es
- e és E betű helyett 3-as

Függvény neve: *LeetKod()*

Paraméter(ek): *vizsgaltSzoveg(szöveg típusú)*

Visszatérési értéke: *szöveg típusú*

6. feladat – OBJEKTUM feldolgozás

Készíts egy interface-t, ami a mellékelt snookerAdatok.js állományban lévő adatok feldolgozását segíti.

Majd tölts fel az adatokat egy objektum típusú tömbbe, ehhez készítsen külön függvényt! Objektum feltöltő néven. A belőle létrehozott objektum: **snookerAdatok** néven legyen elnevezve.

Végül készítsen egy függvényt, arról, mennyi volt a legnagyobb nyeremény, amit nyertek.

Függvény neve: *LegtobbNyeremeny ()*

Paraméter(ek): *vizsgaltObjektum(objektum típusú tömb)*

Visszatérési értéke: *szám érték*

Feltöltés menete!!!:

A feladatokat egyben vagy külön-külön is elkészítheti.

(Egyben készítés esetén ne felejtse a rendszere mentést!)

A Modult egyben az összes, megoldáshoz felhasznált forrásfájl-t tartalmazva tömörítse (zip vagy rar) formátumban, és úgy töltsse fel gitHUB-ra az idő lejárta előtt! ts_modul néven, a megfelelő kiterjesztéssel!

A tömörített fájl tartalmazza legalább a következőket:

- **TypeScript kódot tartalmazó js állomány(ok) (KÖTELEZŐ)**
- **„lefordított” JavaScript kódot tartalmazó js állomány(ok) (KÖTELEZŐ)**
- **tesztfájl és teszfelület(opcionális) Virtuális Pacsi jár érte, plusz hamarabb javítom ki a feladatsort...**

kötelező részek nélkül a modul eredménye 0 pont!