

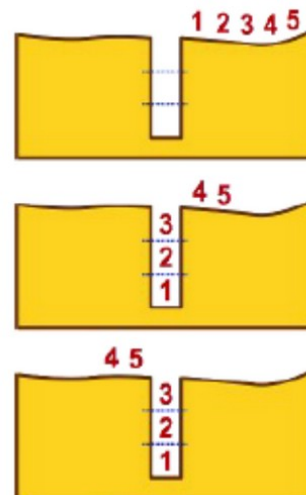
Törpék

Készíts egy programot, ami megoldja az alább leírt feladatot *unit tesztek* segítségével!

A törpék sokszor mennek csoportosan a sötét erdőn keresztül. Az erdőben az ösvények nagyon keskenyek, ezért sorban, egymás után haladnak, anélkül, hogy megelőznék egymást. A törpék egy gödrön a következőképpen jutnak át:

Először annyi törpe ugrik be, amennyi éppen betölti a gödröt. Ezután a csoport többi tagja áthalad a gödör felett. Miután a többiek áthaladtak, a törpék kimásznak a gödörből. Ezután a csoport tovább halad.

A kép azt mutatja, hogyan halad át 5 törpe egy olyan gödör fölött, amelyikbe 3 törpe fér bele.



1. Ábra: 5 törpe áthaladása egy 3 mély gödörön

Készíts alkalmazást, ami

- az inputból bekéri, hogy hány törpe van ($1 \leq n \leq 1000$)
- bekéri, hogy hány gödör van ($1 \leq g \leq 1000$)
- minden gödörről bekéri, hogy milyen mélyek ($1 \leq g_i \leq 8$)

A képes példa kódja (be- és kimenettel) így néz ki:

```
Hány törpe megy át az erdőn? 5
Hány gödör van az útjuk során? 1
Az 1. gödör milyen mély? 3
A törpék a következő sorrendben jönnek ki az erdőből: 4, 5, 3, 2, 1
```

Néhány tipp a fejlesztéshez:

- készíts egy *Class Library* projektet, ahol a törpék mozgását kezeled
- készíts egy *Console Application* projektet, ahol bekéred a felhasználtól az információt, majd megfelelően használod a *Törpe* osztályt, végül az eredményt ebben a projektben írod ki
- készíts egy *Unit Test* projektet, ahol a *Törpe* osztály viselkedését teszteled
- mivel tesztelés dolgozat, ezért azt fogjuk figyelembe venni, hogy a programod logikája összhangban van a tesztekkel, nem a tényleges helyes eredményt
- figyelj oda, hogy próbáld meg több oldalról megközelíteni a problémát: mik lehetnek nehéz esetek? Kezdd a könnyű esetekkel!
- változtasd meg a projekt alapértelmezett osztályainak nevét, hogy tükrözzék a funkciót!
- tesztekben ne akarj inputot bekérni vagy outputot kiíratni!
- figyelj a FIRST elvekre!
- a feladat megoldásához segítségül használhatsz [verem](#) adatszerekezetet