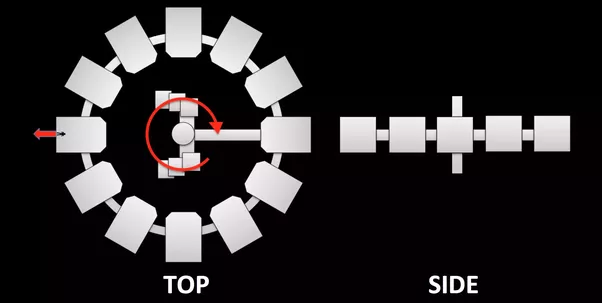
|  |
| --- |
|  |
| Körkörös Lista |
| Dokumentáció |

|  |
| --- |
| Balázs Ádám-Attila Babeș-Bolyai Tudományegyetem  2017.05.07. Adatszerkezetek |

**A feladat kijelentése**

Keretszöveg

2351–ben az emberiségnek sikerül megoldani az interplanetáris utazás problémáját, így 2360–ban útra indítanak egy óriási, önfenntartó űrhajót annak érdekében, hogy több bolygót kolonizáljunk és más galaxisokat közelebbről megvizsgálhassunk. Az űrhajó több modulból áll. A modulok az utazás útvonalának különböző pontjain fognak lekapcsolódni az anyahajóról, hogy a megfelelő bolygóhoz jussanak. Ugyanakkor, az utazás során aszteroidákból bányásznak nyers anyagokat amikből új modulokat tudnak építeni es csatolni az űrhajóhoz. Továbbá, a mesterséges gravitáció létrehozása érdekében az űrhajó struktúrája körkörös lesz, amint az alábbi ábrán látható.



Követelmények

Az úttörőknek szükségük lesz nyilvántartani minden modul lakosságát (foglalkozás, név, kor). Írjunk alkalmazást ezzel a céllal. Tegyük lehetővé a modulok lakosságának módosítását (személy hozzáadása, illetve eltávolítása a nyilvántartásból), új modul hozzáadását, illetve eltávolítását, a lakosság lekérdezését és ezen kívül pedig bármilyen más műveletet, amelyet hasznosnak találunk. Mivel az űrhajó struktúrája nagyon hasonló a körkörös listához, az adatok tárolására használjuk az előbb említett adatszerkezetet. Az űrhajó struktúrájából kiindulva a feladat megoldásához használjuk a Körkörös Lista AAT-t!

Bemeneti adatok

Legelőször a felhasználónak meg kell adnia, hogy hány modulból fog állni az űrhajó (természetes szám, legfennebb 12). Ezután minden modul esetén meg kell mondja, hogy hány személy fog tartozni ehhez a modulhoz (természetes szám, legfennebb 20), majd minden személynek a foglalkozását, nevét és életkorát (pl. Orvos Ferenc 34).

Kimeneti adatok

A felhasználó kérésére a képernyőn megjeleníthető az űrhajó egyetlen moduljának, vagy az összes moduljának a lakói. Az utóbbi esetben a lakók kiírása modulonként történik, nem pedig vegyítve.

Műveletek

**Beolvas:** a felhasználó választása szerint vagy állományból, vagy billentyűzetről, be lesz olvasva a modulok száma, illetve minden modul lakosának az adatai (foglalkozás, név, kor).

* Előfeltételek: –
* Utófeltételek: ha a beolvasott adatok helyesek, akkor ezeket tárolni fogja a memóriában és sikeres beolvasást jelző értéket térít vissza, különben hibajelző értéket térít vissza.

**Egy modul kiírása:** kiírja egy adott indexű modul lakosainak számát és mindeniknek az adatait (foglalkozás, név, kor).

* Előfeltételek: a körkörös lista ne legyen üres, a megadott index helyes legyen, azaz ne legyen nagyobb, mint a pillanatnyilag létező modulok darabszáma és ne legyen kisebb, mint 1.
* Utófeltételek: –

**Összes modul kiírása:** kiírja minden modul esetén a modul sorszámát, a lakóinak számát és mindenik lakónak az adatait (foglalkozás, név, kor).

* Előfeltételek: a körkörös lista ne legyen üres.
* Utófeltételek: –

**Üres–e:** ellenőrzi, hogy a körkörös lista üres vagy sem.

* Előfeltételek: –
* Utófeltételek: ha a modulok száma nulla, true értéket térít vissza, különben false értéket.

**Tele van–e:** ellenőrzi, hogy a körkörös lista maximális számú modult tartalmaz, vagy sem.

* Előfeltételek: –
* Utófeltételek: ha a modulok száma megegyezik a maximális megengedett modulok számával, akkor true értéket térít, különben false értéket.

**Modulok számának lekérdezése:** visszatéríti a pillanatnyilag létező modulok számát.

* Előfeltételek: –
* Utófeltételek: –

**Modul törlése:** egy adott indexű modult töröl a körkörös listából.

* Előfeltételek: a körkörös lista ne legyen üres, a megadott index helyes legyen, azaz ne legyen nagyobb, mint a pillanatnyilag létező modulok darabszáma és ne legyen kisebb, mint 1.
* Utófeltételek: –

**Adott indexű modul lekérdezése:** visszatérít egy mutatót az adott idnexű modulra.

* Előfeltételek: a megadott index helyes legyen, azaz ne legyen nagyobb, mint a pillanatnyilag létező modulok darabszáma és ne legyen kisebb, mint 1.
* Utófeltételek: –

**Új modul beolvasása:** létrehoz egy modult a felhasználó által megadott lakosok számával, és mindeniknek az adataival(foglalkozás, név, kor), majd visszatérít egy mutatót erre az új modulra.

* Előfeltételek: –
* Utófeltételek: a lakosos száma ne legyen negatív, a megadott kor természetes szám legyen

**Új modul beszúrása:**

* Előfeltételek: a körkörös lista ne legyen tele
* Utófeltételek: legyen sikeres az új modul beolvasása és a megadott hely ahova be szeretnénk szúrni is legyen helyes, vagyis a megadott index legyen helyes (azaz ne legyen nagyobb, mint a pillanatnyilag létező modulok darabszáma és ne legyen kisebb, mint 1), illetve a felhasználó választása arra nézve, hogy modul elé vagy után szeretné beszúrni, egyezzen meg a felkínált lehetőséges egyikével („ele” vagy „utan”).

**Adott modul után való beszúrás:** egy adott új modult beszúr a körkörös listába egy másik adott modul után.

* Előfeltételek: A megadott mutatók ne legyenek null mutatók és a megfelelő modulra mutassanak.
* Utófeltételek: –

**Adott modul elé való beszúrás:** egy adott új modult beszúr a körkörös listába egy másik adott modul elé.

* Előfeltételek: A megadott mutatók ne legyenek null mutatók és a megfelelő modulra mutassanak.
* Utófeltételek: –

**Új személy hozzáadása a nyilvántartáshoz:** beolvassa az új személy foglalkozását és nevét és hozzáadja a kiválasztott modul nyilvántartásához.

* Előfeltételek: –
* Utófeltételek: a megadott modul indexe helyes legyen, azaz ne legyen nagyobb, mint a pillanatnyilag létező modulok darabszáma és ne legyen kisebb, mint 1.

**Személy törlése a nyilvántartásból:** beolvassa a személy foglalkozását és nevét, majd az adott indexű modulból kitörli ezt.

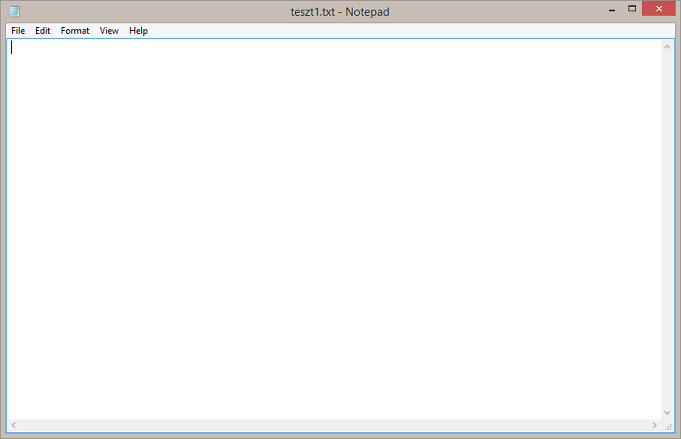
* Előfeltételek: a megadott index legyen helyes, azaz ne legyen nagyobb, mint a pillanatnyilag létező modulok darabszáma és ne legyen kisebb, mint 1.
* Utófeltételek: a megadott foglalkozású és indexű személy kell szerepeljen az adott indexű modul nyilvántartásában

**Év eltelése:** a modul összes lakójának az életkorát növeljük eggyel.

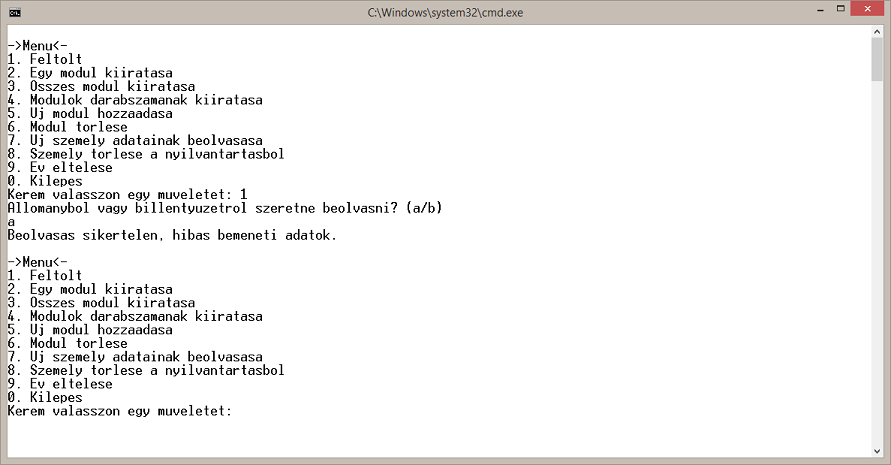
* Előfeltételek: –
* Utófeltételek: –

Tesztállományok

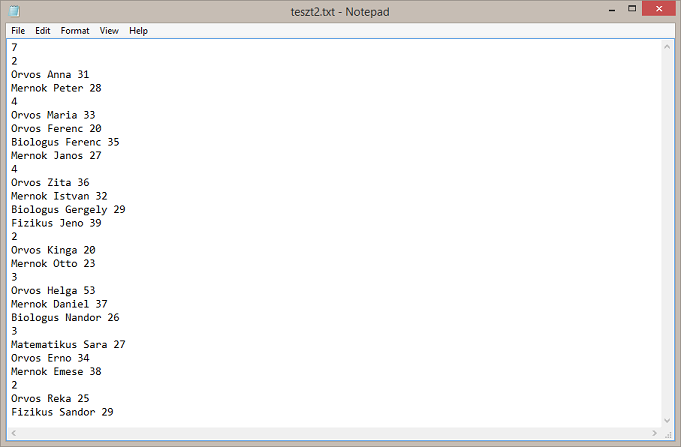
**teszt1** – üres állomány



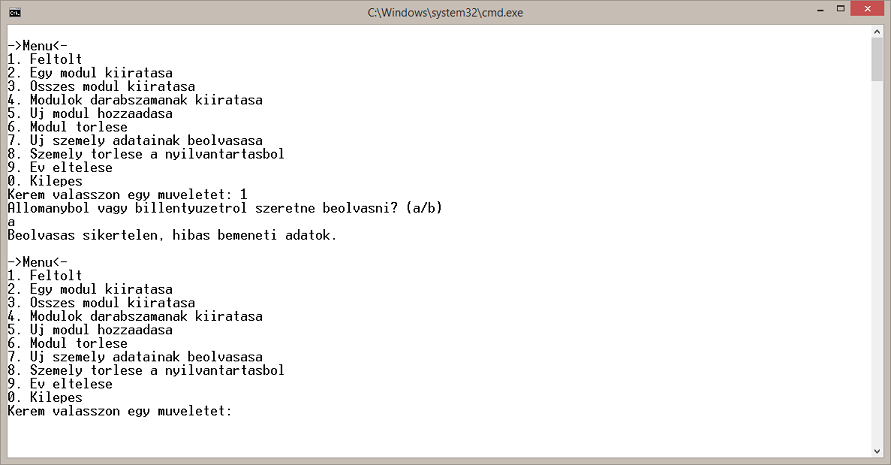
**Kimenet:**

****

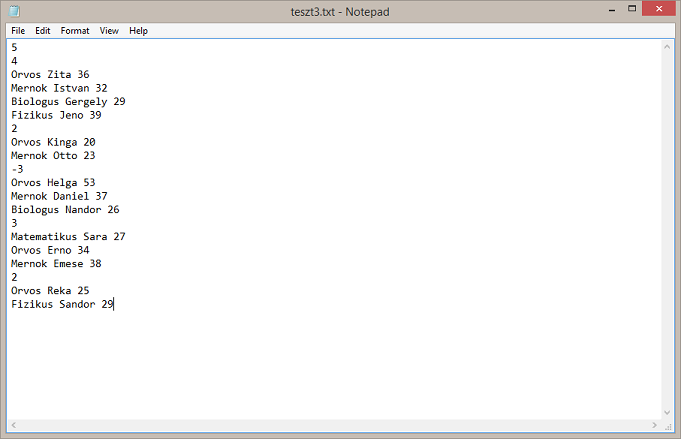
**teszt2** – a modulok számat nagyobb a maximális megengedett számnál (6)



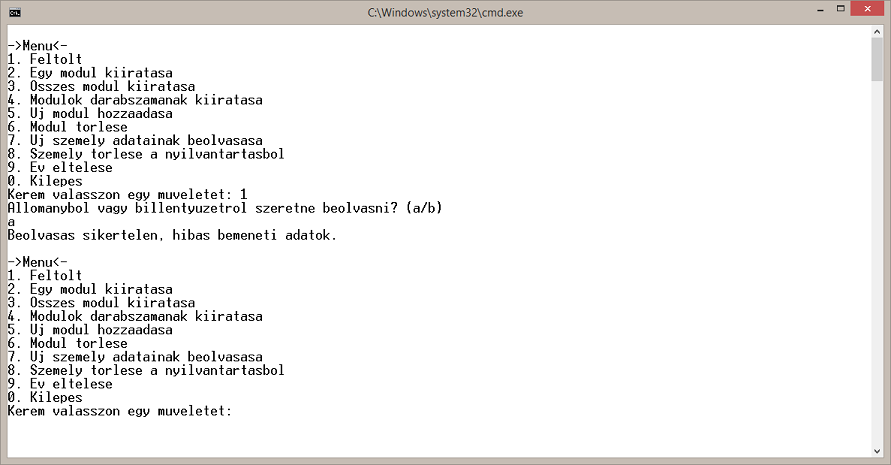
**Kimenet:**



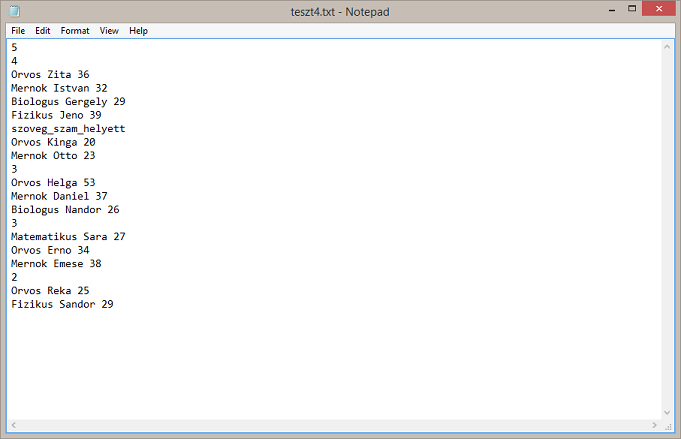
**teszt3** – az állományban lakos darabszámnak negatív érték szerepel

****

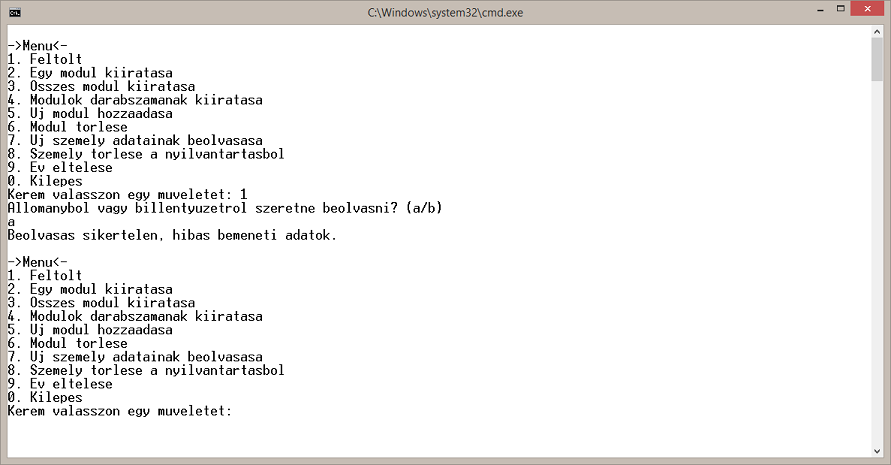
**Kimenet:**

****

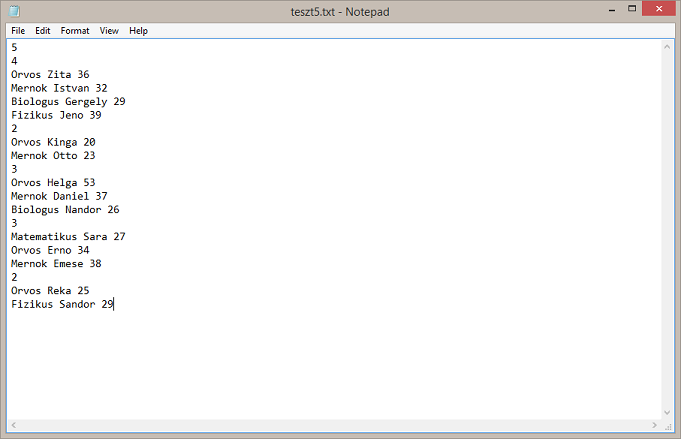
**teszt4:** az állományban lakos darabszáma helyén szöveg van szám helyett



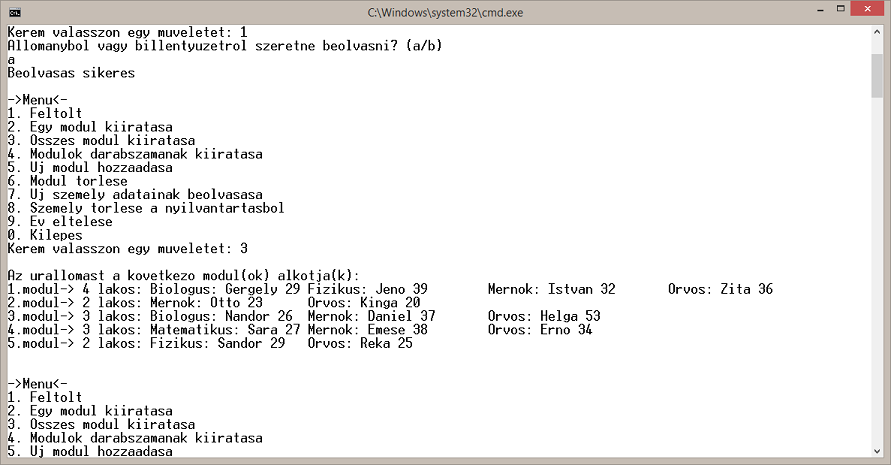
**Kimenet:**



**teszt5** – helyes bemeneti adatok

****

**Kimenet:**

****

Következtetések

Az alkalmazást több helyen lehet még fejleszteni, például a modulok lakosainak több adatait tárolni, mint csak a foglalkozásukat, nevüket és korukat, például haj szín, szem szín, magasság, betegségek és így tovább. Ugyanakkor, minden modulnak tulajdoníthatnánk egy sajátos nevet, például Alfa Modul, Gamma Modul. Továbbá, az alkalmazást felhaszálóbarátabbá lehetne tenni azáltal, hogy grafikus felhasználói felületet készítünk, viszont jelenleg is könnyen használható, mivel a menu opciói világosan kifejezik, hogy milyen műveleteket lehet elvégezni, ezeknek elvégzése közben pedig az alkalmazás kommunikál a felhasználóval és irányadó üzeneteket jelenít meg, hogy világos legyen milyen bemeneti adatokra vár.