

Második beadandó (2019/20/2)

A **gocpont** típusú pointer képes tetszőleges db. fertőzési adat nyilvántartására. Láncolt adatszerkezet fejeként tartja nyilván a fertőzötteket, az ún. nulladik személy adatait tartalmazó **beteg** típusú adatszerkezet címét tartalmazza. Minden személynél rögzítésre kerül az azonosítója, mely egy pozitív int típusú érték, valamint az is, hogy mely más személyeket fertőzött meg. Feltehetjük, hogy a fertőzöttek közül mindenki pontosan egy személytől kapta el a fertőzést, kivéve a nulladik személyt, akinek nincs fertőzője. Az is tudható, hogy egy fertőzött személy tetszőleges számú személyt megfertőzhetett (minimum nulla darab személyt). A tetszőleges azt jelenti számunkra, hogy csak a szabad memória (heap) szab határt. Ugyanazon azonosítóval rendelkező beteg több fertőzésben nem lehet érintett, ezt nem kell vizsgálni, ahogy azokat a kivételeket sem, amire a feladat nem hívja fel a figyelmet. Virtualizálja (implementálja ANSI C nyelven) a

1.
 - **gocpont**
 - **beteg**

absztrakt adatszerkezeteket (típusokat), és a következő funkciókat (függvényeket):

2.
 - Előfeltétel: a paraméterek helyesek
 - Utófeltétel: létrehozzák az adatszerkezeteket
 - Szintaktika:
 - `gocpont uj_gocpont(beteg nulladikbeteg)`
 - `beteg uj_beteg(int azonosito)`
3.
 - Előfeltétel: léteznek a megszüntetendő adatszerkezetek
 - Utófeltétel: megszüntetik az adatszerkezeteket
 - Szintaktika:
 - `void gocpont_torol(gocpont g)` Figyeljen rá, hogy a gócponthoz tartozó összes beteget meg kell szüntetni!!!
 - `void beteg_torol(beteg b)`
4.
 - Előfeltétel: nem NULL a gócpont
 - Utófeltétel: visszaadja az adott gócponthoz tartozó betegek számát
 - Szintaktika: `int letszam(gocpont g)`
5.
 - Előfeltétel: nem NULL a gócpont, létezik a beteg

- Utófeltétel: visszaadja a b beteg által megfertőzöttek számát, ha nincs ilyen beteg az adott gócpontú fertőzésben, akkor -1-et ad vissza
 - Szintaktika: `int fertozottek_szama(gocpont g, beteg b)`
- 6.
- Előfeltétel: nem NULL a gócpont, létezik mind a két beteg
 - Utófeltétel: ha a fertőző az adott gócponthez tartozik, akkor megfertőzi a fertőzöttet és egyet ad vissza, különben nullát
 - Szintaktika: `int uj_beteg(gocpont g, beteg fertozo, beteg fertozott)`
`int uj_beteg_fertozes(gocpont g, beteg fertozo, beteg fertozott)`
- 7.
- Előfeltétel: nem NULL a gócpont
 - Utófeltétel: kiírja az adott gócponthez tartozó összes fertőzött beteg esetében az azonosítóját, és hogy hány személyt fertőzött meg
 - Szintaktika: `void kiir_gocpont(gocpont g)`
- 8.
- Előfeltétel: nem NULL a gócpont, létezik a beteg
 - Utófeltétel: Ha az adott gócpontban nincs ilyen beteg, akkor 0 a visszaadott érték, különben 1.
 - Szintaktika: `int fertozott_e(gocpont g, beteg b)`
- 9.
- Előfeltétel: nem NULL a gócpont, létezik mind a két beteg
 - Utófeltétel: egyet ad vissza, ha b1 megfertőzte a b2 beteget, különben nullát
 - Szintaktika: `int megfertozte_e(gocpont g, beteg b1, beteg b2)`
- 10.
- Előfeltétel: nem NULL a gócpont, létezik a g gócpontban a két beteg
 - Utófeltétel: egyet ad vissza, ha b1 több személyt fertőzött meg, mint b2, ha kevesebbet akkor -1-et, ha ugyanannyit, akkor nullát.
 - Szintaktika: `int fertozobb_e(gocpont g, beteg b1, beteg b2)`

A forráskódot kell feltölteni (egy darab **main.c** állomány), csak szintaktikailag helyes kód kerül kiértékelésre. Kérdésekkel, észrevételekkel fórumban jelentkezzenek.