

## Első beadandó (2019/20/2)

A **"havirendeles"** képes *maximum 1000* db. **"rendeles"** pizzarendelésnek tartalmazására. A **"rendeles"** egy rendelést tartalmaz, mely rögzíti a futársorszámát, a napot, s melyik fajta pizzából hány darabot vitt ki. A futár sorszáma *egyjegyű pozitív szám*, a nap *1-től 30-ig* terjedhet, a **"pizza"** tárolja a pizza fajtáját és darabszámát: a darab *1..9* számok, a fajta az *'A', 'B', 'C', 'D', 'E', 'F'* karakterek valamelyike lehet. Virtualizálja (implementálja ANSI C nyelven) a

1.
  - **havirendeles**
  - **pizza**
  - **rendeles**

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

2.
  - Előfeltétel: a paraméterek helyesek
  - Utófeltétel: létrehozzák az adatszerkezeteket
  - Szintaktika:
    - havirendeles uj\_havirendeles(rendeles r)
    - rendeles uj\_rendeles(int futar, int nap, pizza p)
    - pizza uj\_pizza(int db, char fajta)
3.
  - Előfeltétel: léteznek a megszüntetendő adatszerkezetek
  - Utófeltétel: megszüntetik az adatszerkezeteket
  - Szintaktika:
    - void havirendeles\_torol(havirendeles h)
    - void rendeles\_torol(rendeles r)
    - void pizza\_torol(pizza p)
4.
  - Előfeltétel: a nap 1-30 közötti, h-ban kevesebb, mint 1001 rendelés van.
  - Utófeltétel: visszaadja az adott napon legkelendőbb pizza fajtáját. Ha nem volt aznap kiszállítás, akkor 'X' a visszaadott érték. Több maximum esetén 'Y' kerül visszaadásra.
  - Szintaktika: char napi\_max(havirendeles h, int nap)

5.
  - Előfeltétel: a nap 1-30 közötti, h-ban kevesebb, mint 1001 rendelés van.
  - Utófeltétel: visszaadja az adott napon kiszállított pizzák számát.
  - Szintaktika: `int napi_ossz(havirendeles h, int nap)`
6.
  - Előfeltétel: h-ban kevesebb, mint 1001 rendelés van.
  - Utófeltétel: kiírja az adott hónapban teljesített rendeléseket napokra lebontva, az adott napon futáronként összesítve, pl.
 

2. nap

2. futar: A=9 D=25

8. futar: C=16

5. nap

12. futar E=43 B=1

3. futar B=1 D=5
  - Szintaktika: `void kiir(havirendeles h)`
7.
  - Előfeltétel: h-ban kevesebb, mint 1001 rendelés van.
  - Utófeltétel: visszaadja az adott hónap leggyengébb forgalmú napját. Ha nem volt kiszállítás, akkor '0' a visszaadott érték. Több minimum esetén '-1' kerül visszaadásra.
  - Szintaktika: `int havi_min_nap(havirendeles h)`
8.
  - Előfeltétel: h-ban kevesebb, mint 1001 rendelés van.
  - Utófeltétel: visszaadja az adott hónapban legkelendőbb pizza fajtáját. Ha nem volt kiszállítás, akkor 'X' a visszaadott érték. Több maximum esetén 'Y' kerül visszaadásra.
  - Szintaktika: `char havi_max_pizza(havirendeles h)`
9.
  - Előfeltétel: h-ban kevesebb, mint 1000 rendelés van.
  - Utófeltétel: hozzáadja a rendelést a havirendeléshez.
  - Szintaktika: `void plusz_rendeles(havirendeles h, rendeles r)`
10.
  - Előfeltétel: a paraméterek megfelelnek a kikötéseknek.
  - Utófeltétel: hozzáadja a pizzát a rendeléshez. Ha már volt ilyen fajtájú pizza, akkor csak a darab számát emeli, amennyiben összesen nem haladja meg a határt (9). Ha meghaladja, nem adja hozzá.
  - Szintaktika: `void plusz_pizza(rendeles r, pizza p)`

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