

Listák (dinamikus tömb)

1. Hatékonyság

Hasonlítsuk össze a statikus és dinamikus tömbök hatékonyságát! Deklaráljunk mindkét típussal egy-egy tömböt, majd töltsük fel egymillió véletlenszámmal! Mérjük a feltöltés és az alábbi feladatok végrehajtási idejét!

- Átlagszámítás
- Maximumkiválasztás
- Rendezés
- Elem beillesztése a végére
- Elem beillesztése az elejére

2. Autókereskedés

Írj programot, amely billentyűzetről beolvassa egy használt autókereskedés számára az autók típusát és árát! A típust a **típus** nevű, az árat az **árak** nevű listában tároljuk.

- A program kezdje el bekérni az autók adatait úgy, hogy minden esetben írja ki, hogy hányadik autó adatait kérjük be.
- Ha a felhasználó az autó típusánál „-” karaktert ad meg, akkor fejeződjön be az adatbekérés. (Ekkor már az árat sem kell bekérni.)
- Képernyőtörlés után a program írja ki, hogy hány autó adatait vittük fel.
- A program írja ki a legdrágább autó árát és típusát!
- A program írja ki az autók összértékét!
- A program írja ki az autók átlagárát!
- A program írja ki, hány autó olcsóbb félmillió forintnál!
- A program írja ki a legolcsóbb Opel árát! Ha nincs Opel, akkor azt írja ki!

3. Javítás

A helyesírás szabályai szerint szövegben írásjelek elé nem teszünk szóközt (vessző, pont, kérdőjel, felkiáltójel, pontosvessző, kettőspont), utána viszont teszünk. Ugyancsak nem teszünk szóközt a nyitó zárójelek után és a csukó zárójelek elé (háromféle van: kerek, szögletes és kapcsos). A nyitó zárójel elé kell tenni szóközt és a csukó után is kell, hacsak nem írásjel követi. Ha egyszerre két szabályt kellene alkalmazni (pl. írásjel után kell, csukó zárójel előtt nem kell szóköz), akkor a tiltó szabály az erősebb. Készíts programot `javit` néven, amely egy szöveget átalakít a helyesírás szabályai szerint!

A `javit.be` szöveges állományban egyetlen sor van (legfeljebb 10000 karakter), amely a hibásan beírt szöveget tartalmazza. A `javit.ki` szöveges állományba egyetlen sort kell írni, a javított szöveget.

Példa:

`javit.be`

Ez egy hibásan ,rosszul(rossz zárójelezéssel)írt szöveg .

`javit.ki`

Ez egy hibásan, rosszul (rossz zárójelezéssel) írt szöveg.

(Forrás: Nemes Tihamér OITV 2005. 2. forduló 2. korcsoport 2. feladat)

4. Kiszámolás

Egy kiszámolás játékban N gyerek körbe áll a lenti ábrának megfelelően. A kiszámolás az elsőnél kezdődik, majd minden K -adikat kell kihagyni úgy, hogy végül csak egyetlen egy gyerek maradjon. Először tehát a K . marad ki, majd a $2K$., ... Ha az utolsóhoz értünk, a kör tovább folytatódik. Készíts programot `kiszamol` néven, amely beolvassa a gyerekek számát ($1 < N \leq 100$) és hogy minden hányadikat kell kihagyni ($K \geq 1$), majd kiírja képernyőre a kiszámolás játékban kiesőket, majd pedig a végén megmaradt gyerek sorszámát!

Példa:

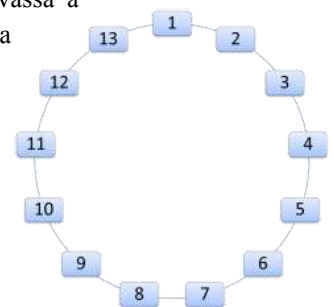
$N=13$

$K=6$

A kimaradók sorban: 6, 12, 5, 13, 8, 3, 1, 11, 2, 7, 4, 10

Végül megmaradt: 9

(Forrás: Nemes Tihamér OITV 1999. 3. forduló 1. korcsoport 1. feladat)



Összetett programozási tételek (másolás, kiválogatás, metszet, unió)

5. Felhasználók

Egy kereskedelmi webportálon több budapesti pizzériából lehet rendelni. Két 14. kerületi pizzéria megrendelői találhatóak a `users1.txt` és `users2.txt` állományokban: soronként ;-vel elválasztva a felhasználók nick-neve, születési dátuma, valamint a lakóhelyének kerülete található a fájlokban.

Olvassuk be a fájlból található adatokat, és oldjuk meg a következő feladatokat:

- Melyik pizzériának van több megrendelője?
- Számoljuk ki az 1. pizzéria a felhasználóinak életkorát (elég az évvel számolni) és határozzuk meg az átlagéletkorukat!
- Válogassuk ki a 14. kerületi felhasználókat és írjuk ki a nevüket és születési dátumukat a `users14.txt` fájlba.
- Gyűjtsük ki azokat a felhasználókat, akik mindkét pizzériából rendeltek! Rendezzük névsorba és írjuk a `users12.txt` fájlba a nevüket és életkorukat ;-vel elválasztva!
- Készítsük el a két pizzéria megrendelőinek egyesített listáját (unióját), és írjuk ki a neveket névsorba rendezetten a `users.txt` fájlba.

6. Meteorok

Egy központi csillagvizsgálóban összegyűjtötték egy adott nap adott órájára (2010.06.21. 23 óra) vonatkozóan az országszerte történt meteorészlelések időpontjait, amelyeket a `meteor.txt` fájlban rögzítettek, soronként egy észlelés időpontját, perc:másodperc formátumban. Írjon programot, amely beolvassa és feldolgozza a fájlban található adatokat a következők szerint:

- A beolvasott időpontokat írja a képernyőre, növekvően rendezetten, sorszámozva.
- Határozza meg az összes olyan észlelési időpontot, amelyet 3 másodpercen belül újabb két észlelés követett!
- Ha van ilyen észlelés, írassa ki az időpontjukat!
- Ha nincs ilyen észlelés, írjon ki ennek megfelelő üzenetet!
- Határozza meg azt a percet, amelyben a legtöbb meteor-észlelés történt! Ha több egyformán legnagyobb is van, akkor mindegyiket írassa ki!

7. Szétválogatás

Válogassuk szét a `diakok.txt` fájlban lévő adatokat külön a fiúk és külön a lányok listájába! A listákban csak a nevek szerepeljenek!

Akasztófa játék

Készítsünk „akasztófa” játékot a `filmek.txt` fájl felhasználásával! A fájl 100 darab filmcímét tartalmaz (mindegyik új sorban). A program véletlenszerűen válassza ki az egyik filmet a fájlból, és az legyen a feladvány!

Kezdetben a betűk helyén csillagok jelenjenek meg, valamint írja ki a program a hibalehetőségek számát!

```
*****
Adj meg egy betűt!
Hibalehetőség: 10
```

A program kérjen egy karaktert a billentyűzetről, ha szerepel a feladványban, akkor a megfelelő helyen jelenítse meg, ha nem szerepel benne, akkor csökkentse a hibalehetőségek számát 1-gyel, valamint írja ki a betűt a képernyő valamely részére!

Például: A felhasználó az „a” betűt adta meg, de az nem szerepel a feladványban:

```
*****
Adj meg egy betűt! a
Hibalehetőség: 9
```

Az alábbi képernyőképen pedig a felhasználó az „e” betűt adta meg és az szerepel a feladványban:

```
*****E*****
Adj meg egy betűt! a
Hibalehetőség: 9
```

A program érjen véget, ha a felhasználó kitalálta, vagy elfogyott a hibalehetőségek száma („felakasztották”).

```
KINCSES BOLYGO
Kitaláltad! a f p m
Hibalehetőség: 6
```

Ha nem sikerült kitalálnia a felhasználónak, akkor a program írja ki a helyes megoldást!

```
KÉT HÉT MŰLVA ÖRÖKKÉ
Felakasztottak! z e s o l p f j u n
A megoldás: KÉT HÉT MŰLVA ÖRÖKKÉ
Hibalehetőség: 0
```

A program megírása közben ügyelj a következőkre:

- ✓ A feladványban a szóközök helyén szóköz (és ne *) legyen!
- ✓ Ne tegyen különbséget kis és nagy betűk között!
- ✓ A feladvány csupa nagybetűvel jelenjen meg.
- ✓ Ha a felhasználó olyan karaktert ad meg, amellyel már korábban próbálkozott sikertelenül, akkor ne csökkenjen a hibalehetőségek száma, és nem kell még egyszer kiírni a megadott betűt!