Feladat

Egy szöveges állományban egy cég gépjárműveinek tankolásainál felvett üzemanyag mennyiségét tároljuk literben. Az állomány minden sorában egy-egy gépjármű adatait találjuk: a jármű rendszámát és mellette annyi darab 0 és 80 közé eső számot, ahányszor tankoltak a járműbe. A tankolások száma járművenként eltérő, sőt olyan jármű is lehet, amelybe egyáltalán nem tankoltak. Az adatok között szóköz(ök) és/vagy tabulátor jelek találhatók. Feltehetjük, hogy az állomány helyesen van kitöltve. Mondja meg, hány olyan jármű van, amelybe egyáltalán nem tankoltak, valamint listázza ki azon járművek rendszámát, amelybe átlagosan 20 liternél több került!

# Specifikáció

Definiáljuk a felsorolandó adatok típusát ekképpen:

Tehát olyan *rekord*ok, melyeknek egyik *mező*je az adott jármű rendszáma, a másik mezője pedig egy számokból álló lista.

ahol a egy olyan függvény lesz, ami megszámolja járművenként az össztankolást.

Ez alapján:

és

# Visszavezetés és algoritmus

egyedi felsoroló, amely típusú elemeket képes felsorolni

|  |  |  |
| --- | --- | --- |
|  | ~ |  |
|  | ~ |  |

Ezt a felsorolót egy eket tartalmazó fájlfelsorolóra építjük. Jelöljük ezt -szel.

Továbbá a felsoroló osztályunk tartalmazzon egy típusú nevű és egy típusú nevű változót is.

Műveletei:

|  |  |
| --- | --- |
|  | |
|  |  |
|  | |
|  | |

|  |  |
| --- | --- |
|  | |
|  |  |
|  | |

|  |  |
| --- | --- |
|  | |
|  |  |
|  | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | |  |  | |
|  |  | | |  |  |  | |
|  |  | | | |  |  | |
|  |  | | | |  |  | |
|  | |  | | | |  | |
|  | |  | | | |  |
|  | |  | | | |  |
|  | |  | | | |  |
|  | |  | | | |  |
|  | |  | | | |  |
|  | |  |  | | |  |
|  | |  | | | |  | |

## Tételek:

számlálás

|  |  |  |
| --- | --- | --- |
|  | ~ |  |
|  | ~ |  |

összegzés

|  |  |  |
| --- | --- | --- |
|  | ~ |  |
|  | ~ |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | |  |
|  |  | | | | | |  |
|  | | | | | |  |  |
|  | |  | | | |  |  |
|  | |  |  | |  |  |
|  | |  | |  | |  |
|  | |  | | | |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | | |  |  | |
|  |  | |  |  |  | |
|  |  | | |  |  | |
|  | | | | |  | |
|  | |  | | |  |
|  | |  | | |  | |

# Implementáció

A programot C++ nyelven valósítottuk meg.

## Adattípusok

A tankolások listája egy vector, amiben egész számok vannak (int).

Magát a felsorolót egy osztállyal valósítottuk meg, a négy művelet kódja a mellékelt struktogramoknak megfelel, mögöttes nevezetes fájlfelsorolót használnak.

A műveletnél található művelet egy getline művelet, bár ez nem string-sorozatot, hanem csak egy stringet ad vissza. A bejárását se indexeléssel oldottuk meg, hanem stringstreammé alakítottuk és szóközönként olvastunk ebből, akár egy fájlból.

### Adatok formátuma:

A mögöttes fájlt soronként dolgozzuk fel, a memóriában a követelményeknek megfelelően mindig egy sornyi adatot tárolunk csak.

A bemenet formátuma kötött, és helyességét NEM ellenőrizzük. A program ugyanakkor MŰKÖDIK üres fájlra is. Ezen kívül értelmetlenül kitöltött fájlokra se „fagy ki” (lásd tesztesetek).

A bemenet egy sora ezt a formátumot követi:

rendszám (3 betű, kötőjel, 3 szám) liter1 … litern (egész számok)

Példa:

CEG-001 15 23 45 7

CEG-003 10 34 42

CEG-101 9 10 10 8 19 47

CEG-100

## A kimenet

A listázás a feldolgozás során (annak ciklusában), a két összetett függvényes tételre adott válasz pedig a teljes feldolgozás után kerül kiírásra. A formátum így alakul:

listázás (rendszámok)

nem tankoltak száma

A bemenetnél megadott példára:

CEG-001

CEG-003

1

## A projekt felépítése

A következő modulokat használjuk:

* *main* – a main.cpp forrásfájl, a program belépési pontja (main függvény, benne a három külső tétellel), valamint a belső tétel implementációja található itt.
* *enor* – enor.h fejlécállomány valamint enor.cpp forrásállomány. Előbbi adja meg a felsoroló típusdefinícióját, illetve a felsorolandó rekord típusát, míg utóbbi a műveletek definícióit.
* *iostream* – külső könyvtár, mely a *konzolos kommunikáció* eszközeit teszi elérhetővé
* *fstream* – külső könyvtár, mely a *fájlkezelés*ben nélkülözhetetlen, a beolvasáshoz használjuk
* *sstream* – külső könyvtár, amelynek segítségével a bemenet egy sorát tudjuk könnyen kezelni és feldolgozni
* *vector* – külső könyvtár, az std::vector típushoz

## Tesztelés

Az alábbiakban megadok néhány érvényes és érvénytelen tesztesetet:

* nem létező fájl [**be0.txt**]
* üres fájl [**be1.txt**] 🡪 az eredmény 0 (a listázás üres string, az összegzés eredménye 0)
* tankolásszámok nélküli sor(ok) [**be2.txt**] – az eredmény az a szám, ahány tankolások nélküli sor van (a listázás üres string, az összegzés eredménye a sorok száma)
* hibás fájl negatív tankolásszámmal [**be3.txt**] 🡪 logikusan működik

Egy elem

* több tankolás tartozik hozzá [**be4.txt**] 🡪 az összegzés eredménye 0, a listázásra helyesen működik

Több elem

* van köztük 0 tankolásos [**be.txt**] 🡪 a dokumentáció példája