

Zadatak:

Napišite sučelje i implementaciju za strukturu `parking` koja opisuje parkiralište s beskonačno parkirnih mjesta (označenih sa 1,2,3,...). Struktura `parking` sadrži mapu koja svakom zauzetom parkirnom mjestu pridružuje registarsku oznaku parkiranog automobila i broj preostalih plaćenih sati parkiranja. Svako je vozilo jedinstveno određeno svojom registarskom oznakom (string grada duljine 2 + broj).

Sučelje za strukturu spremite u datoteku `parking.h`, a implementaciju u datoteku `parking.cpp`.

Struktura Parking

```
struct parking
{
    map<int, pair<string, int> > mapaParking;

    void dodaj(string, int);
    int nadi(string);
    int izbaci(string);
    void istekaoSat(void);
    list<string> preostaloVrijeme(void);
    set<pair<int, string> > izGrada(string);
    pair<int, int> maxBlok(void);
};
```

Funkcije članice

- **void** dodaj(**string** registracija, **int** h)

Dodaje novi automobil registarskih oznaka `registracija` koji ima uplaćeno parkiranje u trajanju od `h` sati na slobodno mjesto s najmanjom oznakom.

- **int** nadi(**string** registracija)

Vraća broj parkirnog mjesta na kojem je parkiran automobil registarske oznake `registracija`.

- **int** izbaci(**string** registracija)

Izbaci automobil s registarskom oznakom `registracija` i vraća 0 ukoliko mu nije isteklo uplaćeno vrijeme za parkiranje, u suprotnom 1.

- **void** istekaoSat(**void**)

Svim parkiranim vozilima smanjuje prestalo vrijeme parkiranja za 1sat pri čemu 0 označava da je vrijeme uplaćenog parkiranja isteklo.

- **list<string>** preostaloVrijeme(**void**)

Vraća listu registarskih oznaka sortiranu silazno po preostalom vremenu parkiranja. Ako postoje dva automobila s istim preostalim vremenom, uspoređuju se registarske oznake (silazno leksikografski).

- **set<pair<int, string>>** izGrada(**string** grad)

Vraća skup parova parkirno mjesto i registarska oznaka svih automobila iz grada `grad` (string duljine 2 kao na registraciji).

- **pair<int, int>** maxBlok(**void**)

Vraća početno i završno parkirno mjesto najvećeg konačnog bloka praznih parkirnih mjesta. Ukoliko je takvih više, bira blok s najmanjim početnim mjestom. Ukoliko ne postoji konačan prazan blok, vraća (0,0).

Opće napomene

- Struktura, funkcije i datoteke koje šalžete moraju se zvati točno onako kako je zadano u zadatku. Pazite na mala i velika slova!
- Trebate poslati samo sučelje i implementaciju. U datotekama koje šalžete ne smije se nalaziti funkcija `main()`!
- nijedna funkcija ne smije ništa učitavati s tipkovnice ili neke datoteke, niti išta ispisivati na ekran ili u neku datoteku.
- Svaki od main-ova pomoću kojih testiramo ispravnost vašeg programa neće pozivati sve gore navedene funkcije. Stoga, ako neku od funkcija ne znate napisati ipak možete dobiti koji bod (u tom slučaju tu funkciju nemojte navesti niti u `.h` niti u `.cpp` datoteci ili napravite neku trivijalnu implementaciju).

Ispravnost implementacija koje napišete bit će provjerena tako da ćemo mi napisati razne klijentske programe koji će deklarirati nekoliko varijabli zadane strukture, i na njima pozivati funkcije koje ste trebali napisati. Ako se poslani programi ne budu uspješno povezivali (linkali) s našim klijentskim programima, smatrat će se neispravnima.

Neki klijentski programi provjeravat će samo neke jednostavnije funkcije, dok će neki provjeravati sve funkcije koje trebate napisati. Provjera je potpuno automatska, tako da je od presudne važnosti da se pridržavate specifikacije. Nepridržavanje lako može uzrokovati osvojenih 0 bodova iz zadaće!

Naravno, za provjeru radi li implementacija prije nego što je pošaljete, preporučuje se da je testirate pomoću nekog klijentskog programa. No taj klijentski program ne šalžete!

Za sva pitanja vezana uz ovu zadaću javite se asistentici Wagner na vanja.wagner+rp1@gmail.com.