

---

# Objektno programiranje (C++)

Druga domaća zadaća (ak. god. 2023./2024.)

---

Datum objave na Merlinu: četvrtak, 28. ožujka 2024. u 20:00h	Rok za predaju preko Merlina: nedjelja, 7. travnja 2024. do 20:00h
Ukupan broj zadataka: <b>2 zadatak</b>	Ukupno moguće ostvariti bodova: <b>8 bodova</b>

**Napomena.** Na Merlinu je uz svaki zadatak bilo potrebno priložiti jednu .cpp datoteku s rješenjem tog zadatka, te u polje za unos teksta uz taj zadatak kratko opisati na koji način je taj zadatak bio riješen.

**Zadatak 1.** (4 boda) Napišite program koji radi rangiranje filmova prema ocjenama publike. Naziv filma je ujedno i naziv .txt datoteke. Primjerice, za film s nazivom "Gospodar podataka - Povratak prioritnog reda" imamo datoteku *Gospodar podataka - Povratak prioritnog reda.txt*. Svaka datoteka sadrži ocjene publike (barem jednu!) za film s tim nazivom, a svaka ocjena je prirodan broj manji ili jednak 5. Potrebno je napraviti strukturu `film` koja sadrži sve navedene podatke o pojedinom filmu (naziv filma i sve ocjene za taj film, uz ostale podatke za koje smatrate da ih treba pamtit). Zatim treba korištenjem prikladnog spremnika (ili adaptera) spremiti podatke o svim filmovima čije nazive odgovarajućih datoteke (bez ekstenzije ".txt") program prima kao argumente komandne linije (ne treba provjeravati uspješnost otvaranja pojedine datoteke). Pritom mora vrijediti:

- Filmovi u tom spremniku (ili adapteru) sortirani su silazno prema prosječnoj ocjeni gledatelja, tako da korisnik može dobiti film s najvećom prosječnom ocjenom u vremenu  $O(1)$  te ubaciti podatke o novom filmu u vremenu  $O(\log n)$ . Također, nemojte da se pri ubacivanju filma mora puno puta računati prosječna ocjena gledatelja za filmove koji su već u spremniku.

Nakon učitavanja svih podataka, ispišite sve podatke o svim filmovima iz korištenog spremnika u silaznom poretku prema prosječnoj ocjeni filma (dakle tako da je film s najvećom prosječnom ocjenom ispisan prvi). U komentar u kodu obrazložite izbor spremnika (ili adaptera) koji ste koristili.

**Napomena** prije primjera: Naziv filma (pa onda i naziv datoteke) može sadržavati praznine. U slučaju da kao argument komandne linije navodimo string koji sadrži praznine, potrebno ga je navesti unutar dvostrukih navodnika (važno: to ne utječe na kod, primjerice, ako je u komandnoj liniji navedeno "Neki film" tada će kao argument komandne linije biti učitani string `Neki film`).

**Primjer.** Neka je sadržaj datoteke *Kung Fu Panda 4.txt* sljedeći:

4 1 5 2 4 3 5 1 3 2 4 3  
4 2 3 4 2 2 2 1 5 5 3 2

Neka je sadržaj datoteke *Argylle.txt* sljedeći:

4 1 5 2 4  
5 1 3 2 4  
4 2 3 4 2  
2 1 5 5 3

Neka je sadržaj datoteke *Godzilla x Kong - Novo carstvo.txt* sljedeći:

2 3 1 4 3  
2 2 2 1 5  
3 4 3 2 4  
4

Pod pretpostavkom da smo kod rješenja spremili kao datoteku *zad1.cpp* i kompilirali ovako:

```
g++ zad1.cpp -std=c++11 -o prog
```

pokretanje pomoću:

```
./prog "Kung Fu Panda 4" "Argylle" "Godzilla x Kong - Novo carstvo"
```

daje sljedeći ispis:

```
Naziv filma: Argylle  
Ocjene: 4 1 5 2 4 5 1 3 2 4 4 2 3 4 2 2 1 5 5 3  
Prosjek: 3.1  
Naziv filma: Kung Fu Panda 4  
Ocjene: 4 1 5 2 4 3 5 1 3 2 4 3 4 2 3 4 2 2 2 1 5 5 3 2  
Prosjek: 3  
Naziv filma: Godzilla x Kong - Novo carstvo  
Ocjene: 2 3 1 4 3 2 2 2 1 5 3 4 3 2 4 4  
Prosjek: 2.8125
```

□

**Zadatak 2.** (4 boda) Datoteka *prodaja.txt* sadrži podatke o prodanim kartama jedne autobusne tvrtke. Za svaku kartu upisani su u tu datoteku sljedeći podaci (svaki u posebnom retku!):

- naziv polazišta,
- naziv odredišta,
- cijena (broj koji predstavlja iznos u EUR),
- tip karte (jednosmjerna ili dvosmjerna),
- datum i vrijeme polaska (oblika mm:hh dd.MM.yyyy).

U datoteci nakon podataka za pojedinu kartu mogu postojati „prazni” retci, preciznije, retci koji sadrže samo bjeline (bjeline su ' ', '\f', '\n', '\r', '\t', '\v') - takve retke zane-marite.

Definirajte strukturu `relacija` koja pamti podatke o jednoj relaciji i to:

- naziv polazišta za tu relaciju,
- naziv odredišta za tu relaciju.

Korištenjem neuređenog višestrukog preslikavanja - preciznije, `unordered_multimap` spremnika uz minimalno 10 pretinaca, učitajte podatke o kartama iz datoteke *prodaja.txt* tako da za svaku relaciju imamo spremljene podatke o kartama za tu relaciju.

Naposlijetku, ispišite na ekran sve učitane podatke po pretincima, na način da se prvo ispiše broj pretinca te zatim svi podaci iz tog pretinca - svaki redak mora biti u formatu:

(polazište, odredište) -> x EUR, tip, datum i vrijeme

Nemojte ispisivati podatke o praznim pretincima. Pri definiranju vlastite *hash*-funkcije, kratko obrazložite u komentar u svom kodu odabir te *hash*-funkcije.

### **Primjer.**

Ako je sadržaj datoteke *prodaja.txt* sljedeći:

```
Varazdin
Zagreb
8.99
jednosmjerna
07:00 27.03.2024
```

```
Ladanje Gornje
Varazdin
3.40
jednosmjerna
```

16:20 12.03.2024

Varazdin  
Dubrovnik  
25.89  
jednosmjerna  
07:00 27.03.2024

Zagreb  
Varazdin  
14.45  
dvosmjerna  
14:30 20.03.2024

Varazdin  
Zagreb  
14.45  
dvosmjerna  
07:00 27.03.2024

Bednja  
Ivanecko Naselje  
3.10  
jednosmjerna  
08:40 27.03.2024

tada bi ispis mogao biti (naravno ovisi o, primjerice, *hash*-funkciji koja se koristi):

Elementi u pretincu 0:

(Varazdin,Dubrovnik) -> 25.89 EUR, jednosmjerna, 07:00 27.03.2024

Elementi u pretincu 2:

(Ladanje Gornje,Varazdin) -> 3.40 EUR, jednosmjerna, 16:20 12.03.2024

Elementi u pretincu 3:

(Zagreb,Varazdin) -> 14.45 EUR, dvosmjerna, 14:30 20.03.2024

(Varazdin,Zagreb) -> 14.45 EUR, dvosmjerna, 07:00 27.03.2024

(Varazdin,Zagreb) -> 8.99 EUR, jednosmjerna, 07:00 27.03.2024

Elementi u pretincu 6:

(Bednja,Ivanecko Naselje) -> 3.10 EUR, jednosmjerna, 08:40 27.03.2024

## **VAŽNO:**

Iako kod koji napišete kompiliramo na Linuxu, postoji mogućnost da je tekstualna datoteka napravljena na, primjerice, Windowsima (kao što je primjer datoteke *prodaja.txt* koja je objavljena uz ovu zadaću). Tada svaka linija u toj datoteci završava ne samo s '\n' nego s kombinacijom znakova '\r' i '\n'. To znači da će, primjerice, *getline* učitati i taj znak '\r'. To pak znači da će se nakon ispisa stringa koji završava s '\r', pisač vratiti

na početak tog retka u kojem je taj string ispisan i sljedeći ispis ide preko već ispisanog stringa! Zbog toga je potrebno da, ako primijetite da Vam ispis izgleda čudno, iz učitanih stringova uklonite znak '\r' ako on postoji.

□