# Strukture podataka i algoritmi

## Projektni zadatak 008\_CARS

Vaš zadatak je napisati program (konzolnu aplikaciju u C++) koji će korisniku omogućiti rad s podacima o automobilima. Program mora biti smisleno strukturiran i intuitivan. U nastavku su opisane funkcionalnosti koje treba program imati. Za izradu projekta se koristi datoteka naziva **SPA\_PROJ\_008\_CARS\_data\_3.csv**. Opis stupaca:

* + id: jedinstveni identifikator verzije modela.
  + brand\_id: identifikator proizvođača.
  + model\_id: identifikator modela.
  + name: naziv verzije modela.
  + year: godina kad je verzija izdana.
  + price: cijena verzije u dolarima (ako je 0, cijena nije poznata).

1. (Bodova: **1**) Pokretanjem aplikacije prikažite korisniku sljedeći izbornik:

Pretraga po modelu

Prikaz po godini

Kopiranje u novi vektor

Prikaz po godini

1. (Bodova: **2**) Dizajnirajte i implementirajte tip podataka Car koji može čuvati sve podatke iz datoteke. Pripremite tip podataka tako da se može konstruirati na smislene načine te definirajte gettere, settere i potrebne metode za svakog člana. Omogućite korisniku da pozivanjem metode na objektu može ispisati osnovne podatke o automobilu na ekran.
2. (Bodova: **2**) Kad korisnik odabere opciju „Pretraga po modelu“, prvo prekopirajte sve automobile iz vektora u odgovarajući kontejner. Zatim pitajte korisnika da upiše model\_id kojeg želi tražiti pa mu nakon toga ispišite podatke o automobilima tog modela\_id-a ili napišite da ih nema. Osigurajte da se pretraživanje odvija u logaritamskoj složenosti koristeći najbolji od sljedećih kontejnera: map, multimap, set, multiset.
3. (Bodova: **2**) Promijenite prethodni zadatak tako da pretraživanje radite prvo po odabranom kontejneru, a zatim i po vektoru i po listi (kopirajte podatke i u listu). Ispišite top ljestvicu brzine kontejnera: prvo prikažite najbrži, onda srednji pa najsporiji.
4. (Bodova: **2**) Kad korisnik odabere opciju „Prikaz po godini“, koristeći prioritetni red ispišite automobile rastuće prema godini.
5. (Bodova: **3**) Kad korisnik odabere „Kopiranje u novi vektor“, kreirajte novi vektor i napravite sljedeće:

* Prekopirajte sve automobile u novi kontejner koji omogućuje da se pretraživanje po brand\_id odvija u logaritamskoj složenosti (birajte map, multimap, set ili multiset).
* Sve dok to korisnik želi, omogućite mu unos brand\_id.
* Kad korisnik unese vrijednost, pronađite sve automobile u vašem odabranom kontejneru i premjestite ih u novi vektor.
* Ponavljajte sve dok to korisnik želi.
* Na kraju ispišite sve automobile iz novog vektora.

1. (Bodova: **3**) Kad korisnik odabere „Prikaz po godini“, koristeći prioritetni red ispišite sve automobile rastuće prema godini. Sve automobile iz iste godine obavezno prikažite tako sortirane prema cijeni.
2. (Bodova: **1**) Dodajte u izbornik sljedeće opcije:

Selection vs Insertion

Pogodi trajanje

Lets Bogo!

The Merge Dance

1. (Bodova: **3**) Kad korisnik odabere „Selection vs Insertion“, napravite sljedeće:

* Kreirajte dva pomoćna vektora *v1* i *v2* i iskopirajte u svako od njih sve automobile.
* Uzmite s predavanja Selection i Insertion sortiranja te ih promijenite tako da sortiraju automobile, rastuće po nazivu.
* Sortirajte vektor *v1* koristeći Selection sort.
* Sortirajte vektor *v2* koristeći Insertion sort.
* Ispišite koliko je trajalo svako sortiranje.

1. (Bodova: **2**) Promijenite prethodni zadatak tako da svaki algoritam pri sortiranju zbraja broj zamjena mjesta (*swap*) koje napravi i vraća to kao izlaznu vrijednost iz funkcije. Sortirajte pa ispišite za oba sortiranja broj zamjena mjesta.
2. (Bodova: **3**) Kad korisnik odabere opciju „Pogodi trajanje“, generirajte jedan slučajni broj *n* između 1 i 100.000.000 te pitajte korisnika da upiše koliko milisekundi misli da će trajati razbacivanje, sortiranje i binarno pretraživanje tog polja u potrazi za brojem 7. Kad korisnik upiše vrijednost, generirajte to polje s vrijednostima od 1 do *n*, razbacajte ga, sortirajte ga te binarnim pretraživanjem i pronađite broj 7. Ispišite korisniku kolika je razlika između njegove pogođene vrijednosti trajanja i konkretnog trajanja.
3. (Bodova: **4**) Kad korisnik odabere „Lets Bogo!“, slučajnim odabirom prekopirajte pet automobila u novi vektor. Isprogramirajte svoju vlastitu determinističku verziju bogo sorta te je iskoristite za slaganje automobila po nazivu u rastućem redoslijedu. Ispišite na ekranu svaku isprobanu permutaciju.
4. (Bodova: **4**) Kad korisnik odabere opciju „The Merge Dance“, pitajte ga da upiše broj *n* koji predstavlja broj elemenata. Nakon što upiše broj, u vektor ubacite slučajnim redoslijedom *n* pokemona. Zatim vektor sortirajte Merge sortom (koristite i promijenite prema potrebi implementaciju s predavanja) te za vrijeme sortiranja radite sljedeće:

* Svaki put kad se napravi poziv rekurzivne funkcije, brojač povećajte za 1
* Svaki put kad se završi poziv rekurzivne funkcije, brojač smanjite za 1.
* Za svaku promjenu brojača iscrtajte u tekstualnu datoteku jedan redak koji sadrži onoliko znakova # kolika je trenutna vrijednost brojača.

1. (Bodova: **1**) Dodajte u izbornik sljedeće opcije:

Pretraga po volumenu

Vizualiziraj

Izrada stringa iz stringa

1. (Bodova: **2**) Kad korisnik odabere „Pretraga po volumenu“, prvo prekopirajte automobile iz vektora u odgovarajući kontejner. Zatim pitajte korisnika da upiše volumen (primjerice, „1.6“) koji želi tražiti pa mu nakon toga ispišite podatke o svim automobilima koji u nazivu sadrže taj volumen. Osigurajte da se pretraživanje odvija u najboljoj mogućnoj složenosti koristeći neki od sljedećih kontejnera: unordered\_map, unordered\_multimap, unordered\_set, unordered\_multiset.
2. (Bodova: **2**) Promijenite prethodni zadatak tako da pretraživanje radite prvo po odabranom kontejneru, a zatim i po vektoru i po listi (kopirajte podatke i u listu). Ispišite top ljestvicu brzine kontejnera: prvo prikažite najbrži, onda srednji pa najsporiji.
3. (Bodova: **5**) Kad korisnik odabere „Vizualiziraj“, natrpajte sve automobile u unordered\_multimap tako da ključ bude godina izdavanja. Nakon toga, vizualizirajte *hash* tablicu iz unordered\_multimap tako da iscrtate sve njene *buckete* te uz svaki *bucket* prikažete i automobile u njemu. Primjer početka ispisa (vrijednosti su izmišljene):

Bucket 0: Alfa Romeo Brera 2.2 JTS Sky Window -- Porsche Macan S

Bucket 1: Fiat 500C 1.2 Cult -- Hyundai Coupe 2.0 GLS -- Opel Calibra 2.5 V6

Bucket 2: [EMPTY]

1. (Bodova: **6**) Kad korisnik odabere „Izrada imena iz imena“, omogućite mu da odabere jedan automobil i da učita jedan string. Koristeći unordered\_map, unordered\_multimap, unordered\_set ili unordered\_multiset, ispišite može li se učitani string kreirati koristeći raspoloživa slova iz naziva automobila (pri tome mala i velika slova smatrajte jednakima). Primjerice:

* Ako je ime automobila „Subaru Outback 2.0 AWD Turbo Diesel Premium“, a učitani string „out“, onda se učitani string može konstruirati iz imena automobila.
* Ako je ime automobila „Ferrari F355 F1 Berlinetta“, a učitani string „rrrrr“, onda se učitani string ne može konstruirati iz imena autmobila prve jer nedostaje jedno slovo r.