# Strukture podataka i algoritmi

## Projektni zadatak 005\_MUSIC

Vaš zadatak je napisati program (konzolnu aplikaciju u C++) koji će korisniku omogućiti rad sa glazbenicima i albumima. Program mora biti smisleno strukturiran i intuitivan. U nastavku su opisane funkcionalnosti koje treba program imati. Za izradu projekta se koristi datoteka **SPA\_PROJ\_005\_MUSIC\_data.csv**. Opis stupaca:

* title: Song's title
* artist: Song's artist
* top genre: the genre of the track
* year: Song's year in the Billboard
* bpm: Beats.Per.Minute - The tempo of the song.
* nrgy: Energy- The energy of a song - the higher the value, the more energtic. song
* dnce: Danceability - The higher the value, the easier it is to dance to this song.
* dB: Loudness..dB.. - The higher the value, the louder the song
* live: Liveness - The higher the value, the more likely the song is a live recording
* val: Valence - The higher the value, the more positive mood for the song.
* dur: Length - The duration of the song.
* acous: Acousticness.. - The higher the value the more acoustic the song is.
* spch: Speechiness - The higher the value the more spoken word the song contains.
* pop: Popularity - The higher the value the more popular the song is.

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

Pretraga po žanru

Prikaz po godini

Kopiranje u novu listu

Prikaz po prvom slovu

1. (Bodova: **2**) Dizajnirajte i implementirajte tip podataka Song koji može čuvati sve podatke o pjesmi iz datoteke **SPA\_PROJ\_005\_MUSIC\_data.csv**. 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 pjesmi na ekran (naziv, autor, godina izdanja).
2. (Bodova: **2**) Kad korisnik odabere opciju „Pretraga po žanru“, prvo prekopirajte sve pjesme iz vektora u odgovarajući kontejner. Zatim pitajte korisnika da upiše žanr koji želi tražiti pa mu nakon toga ispišite podatke o pjesmama u tom žanru ili napišite da ne postoje. 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 pjesme padajuće prema godini izdavanja.
5. (Bodova: **3**) Kad korisnik odabere „Kopiranje u novu listu“, kreirajte novu listu i napravite sljedeće:

* Prekopirajte sve pjesme u novi kontejner koji omogućuje da se pretraživanje po godini izdavanja odvija u logaritamskoj složenosti (birajte map, multimap, set ili multiset).
* Sve dok to korisnik želi, omogućite mu unos godine izdavanja.
* Kad korisnik unese neku godinu izdavanja, pronađite sve pjesme iz te godine u vašem odabranom kontejneru i premjestite ih u novu listu.
* Ponavljajte sve dok to korisnik želi.
* Na kraju ispišite sve pjesme iz nove liste.

1. (Bodova: **3**) Kad korisnik odabere „Prikaz po prvom slovu“, koristeći prioritetni red ispišite sve pjesme abecedno prema prvom slovu naziva. Sve pjesme s istim prvim slovom obavezno prikažite padajuće prema godini.
2. (Bodova: **1**) Dodajte u izbornik sljedeće opcije:

Selection vs Bubble

What will the number be

Randomness

Sortiranje brojanjem

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

* Kreirajte dva pomoćna vektora *v1* i *v2* i iskopirajte u svako od njih sve pjesme.
* Uzmite s predavanja Selection i Bubble sortiranja i promijenite ih tako da sortiraju pjesme, padajuće po naslovu.
* Sortirajte vektor *v1* koristeći Selection sort.
* Sortirajte vektor *v2* koristeći Bubble 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 „What will the number be“, generirajte jedan slučajni broj *n* između 1 i 100.000.000 te pitajte korisnika da upiše koliko mikrosekundi misli da će trajati razbacivanje, sortiranje i binarno pretraživanje tog vektora u potrazi za brojem 13. Kad korisnik upiše vrijednost, generirajte taj vektor s vrijednostima od 1 do *n*, razbacajte ga, sortirajte ga te binarnim pretraživanjem i pronađite broj 13. Ispišite korisniku kolika je razlika između njegove pogođene vrijednosti trajanja i konkretnog trajanja.
3. (Bodova: **4**) Kad korisnik odabere „Randomness“, slučajnim odabirom prekopirajte šest pjesama u novi vektor. Isprogramirajte svoju vlastitu determinističku verziju bogo sorta te je iskoristite za slaganje pjesama po nazivu u padajućem redoslijedu. Ispišite koliko je pokušaja bilo potrebno da se elementi sortiraju.
4. (Bodova: **4**) Kad korisnik odabere opciju „Sortiranje brojanjem“, pitajte ga da upiše broj *n* koji predstavlja maksimalni element. Nakon što upiše broj, u vektor ubacite 100.000 slučajnih brojeva između 1 i *n*. Zatim vektor sortirajte Counting sortom (koristite i promijenite prema potrebi implementaciju s predavanja) te za vrijeme sortiranja radite sljedeće:

* Prilikom prebrajanja koliko ima kojih elemenata, za svaki obrađeni element ispišite u tekstualnu datoteku sadržaj pomoćnog polja koje sadržava broj pronađenih elemenata. Primjerice, ako se radi o polju [ 2, 2, 1, 3, 2 ], u datoteci treba pisati:

0 0 0 0

0 0 1 0

0 0 2 0

0 1 2 0

0 1 2 1

0 1 3 1

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

Pretraga po godini

Vizualiziraj

Izrada imena iz imena

1. (Bodova: **2**) Kad korisnik odabere „Pretraga po godini“, prvo prekopirajte pjesme iz vektora u odgovarajući kontejner. Zatim pitajte korisnika da upiše godinu koju želi pretraživati pa mu nakon toga ispišite podatke o svim pjesmama iz te godine ili ispišite da ne postoji niti jedna pjesma. 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 pjesme u unordered\_multimap tako da ključ bude izvođač (svojstvo „artist“). Nakon toga, vizualizirajte *hash* tablicu iz unordered\_multimap tako da u tekstualnu datoteku iscrtate sve njene *buckete* te uz svaki *bucket* prikažete i pjesme u njemu. Primjer početka ispisa (vrijednosti su izmišljene):

Bucket 0: Dog Days Are Over -- Turning Page -- Everybody Talks

Bucket 1: People Like Us -- Blame -- Sparks -- Invitation

Bucket 2: [EMPTY]

1. (Bodova: **6**) Kad korisnik odabere „Izrada imena iz imena“, omogućite mu da odabere jednu pjesmu te mu zatim, koristeći unordered\_map, unordered\_multimap, unordered\_set ili unordered\_multiset, ispišite može li se ime izvođača kreirati koristeći raspoloživa slova iz naslova pjesme (pri tome mala i velika slova smatrajte jednakima). Primjerice:

* Ako je naslov pjesme „Keeping Your Head Up“, a izvođača „Pingu“, onda se ime izvođača može konstruirati iz naslova pjesme.
* Ako je naslov pjesme „Lady Bullseye“, a izvođača „Lady Bulls Eye“, onda se ime izvođača ne može konstruirati iz naslova pjesme jer nedostaje jedan razmak.