# Strukture podataka i algoritmi

## Projektni zadatak 006\_GAME\_SALES

Vaš zadatak je napisati program (konzolnu aplikaciju u C++) koji će korisniku omogućiti rad s podacima o prodaji igara. Program mora biti smisleno strukturiran i intuitivan. U nastavku su opisane funkcionalnosti koje treba program imati. Za izradu projekta se koristi datoteka **SPA\_PROJ\_006\_GAME\_SALES\_data.csv**.

### Ishod 3 (bodova: 15)

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

Pretraga po žanru

Prikaz po godini

Kopiranje u novi vektor

Prikaz po globalnoj prodaji

1. (Bodova: **2**) Dizajnirajte i implementirajte tip podataka VideoGame koji može čuvati sve podatke o igri iz datoteke **SPA\_PROJ\_006\_GAME\_SALES\_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 igri na ekran (naziv, izdavač, godina izdanja, globalno zarađeni iznos).
2. (Bodova: **2**) Kad korisnik odabere opciju „Pretraga po žanru“, prvo prekopirajte sve igre iz vektora u odgovarajući kontejner. Zatim pitajte korisnika da upiše žanr koji želi tražiti pa mu nakon toga ispišite podatke o igrama 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 igre rastuće prema godini izdavanja.
5. (Bodova: **3**) Kad korisnik odabere „Kopiranje u novi vektor“, kreirajte novi vektor i napravite sljedeće:

* Prekopirajte sve igre u novi kontejner koji omogućuje da se pretraživanje po žanru odvija u logaritamskoj složenosti (birajte map, multimap, set ili multiset).
* Sve dok to korisnik želi, omogućite mu unos žanra.
* Kad korisnik unese neki žanr, pronađite sve igre iz te godine u vašem odabranom kontejneru i prekopirajte ih u novi vektor.
* Ponavljajte sve dok to korisnik želi.
* Na kraju ispišite sve igre iz novog vektora.

1. (Bodova: **3**) Kad korisnik odabere „Prikaz po globalnoj prodaji“, koristeći prioritetni red ispišite sve igre padajuće prema iznosu globalne prodaje (svojstvo „Global\_Sales“). Sve igre s istim iznosom prodaje obavezno prikažite abecedno rastuće prema nazivu.

### Ishod 4 (bodova: 17)

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

Bubble vs Insertion

Pogodi trajanje

Lets Bogo!

Quick quick

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

* Kreirajte dva pomoćna vektora *v1* i *v2* i iskopirajte u svako od njih sve igre.
* Uzmite s predavanja Bubble i Insertion sortiranja te ih promijenite tako da sortiraju igre, rastuće po nazivu.
* Sortirajte vektor *v1* koristeći Bubble 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 igara u novi vektor. Isprogramirajte svoju vlastitu slučajnu verziju bogo sorta te je iskoristite za slaganje igara po imenu u rastućem redoslijedu. Ispišite na ekranu svaku isprobanu kombinaciju.
4. (Bodova: **4**) Kad korisnik odabere opciju „Quick quick“, pitajte ga da upiše broj *n* koji predstavlja broj elemenata. Nakon što upiše broj, u vektor ubacite slučajnim redoslijedom *n* igara. Zatim vektor sortirajte Quick 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.

### Ishod 5 (bodova: 16)

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

Pretraga po platformi

Vizualiziraj

Izrada imena iz imena

1. (Bodova: **2**) Kad korisnik odabere „Pretraga po platformi“, prvo prekopirajte igre iz vektora u odgovarajući kontejner. Zatim pitajte korisnika da upiše platformu koju želi tražiti pa mu nakon toga ispišite podatke o svim igrama na toj platformi. 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 igre u unordered\_map tako da ključ bude njihov rang (svojstvo „Rank“). Nakon toga, vizualizirajte *hash* tablicu iz unordered\_map tako da iscrtate sve njene *buckete* te uz svaki *bucket* prikažete i igre u njemu. Primjer početka ispisa (vrijednosti su izmišljene):

Bucket 0: Wii Play -- Minecraft -- Pokemon Pinball

Bucket 1: Sonic the Hedgehog -- Kung Fu -- The Sims 4 -- Pure

Bucket 2: [EMPTY]

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

* Ako je ime prve igre „Frogger's Adventures: Temple of the Frog“, a druge „Ventures“, onda se ime druge može konstruirati iz imena prve.
* Ako je ime prve igre „Lady Bullseye“, a druge „Lady Bulls Eye“, onda se ime druge ne može konstruirati iz prve jer nedostaje jedan razmak.