# Strukture podataka i algoritmi

## Projektni zadatak 003\_HEROES

Vaš zadatak je napisati program (konzolnu aplikaciju u C++) koji će korisniku omogućiti rad sa super junacima. Program mora biti smisleno strukturiran i intuitivan. U nastavku su opisane funkcionalnosti koje treba program imati. Za izradu projekta se koristi datoteka **SPA\_PROJ\_003\_HEROES\_data\_1.csv**.

### Ishod 3 (bodova: 15)

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

Pretraga po imenu

Prikaz po brzini

Kopiranje u novi vektor

Prikaz po strani

1. (Bodova: **2**) Dizajnirajte i implementirajte tip podataka Superhero koji može čuvati sve podatke o super herojima iz datoteke **SPA\_PROJ\_003\_HEROES\_data\_1.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 super heroju na ekran (ime, je li dobar ili zao te ukupni broj bodova iz stupca „Total“).
2. (Bodova: **2**) Kad korisnik odabere opciju „Pretraga po imenu“, prvo prekopirajte sve super junake iz vektora u odgovarajući kontejner (imena su jedinstvena u datoteci). Zatim pitajte korisnika da upiše ime koje želi tražiti pa mu nakon toga ispišite podatke o super junaku s tim imenom ili napišite da ne postoji super junak s tim imenom. 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 „Prikaz po brzini“, pitajte korisnika da odabere prikaz od slabijih prema jačim ili od jačih prema slabijim super junacima (svojstvo „Speed“). Koristeći prioritetni red, ispišite korisniku sve super junake, poštujući odabrani redoslijed.
5. (Bodova: **3**) Kad korisnik odabere „Kopiranje u novi vektor“, kreirajte novi vektor *v* i napravite sljedeće:

* Prekopirajte sve super junake u novi kontejner koji omogućuje da se pretraživanje po svojstvu „Total“ odvija u logaritamskoj složenosti (optimalno birajte map, multimap, set ili multiset).
* Sve dok to korisnik želi, omogućite mu unos min i max vrijednosti za „Total“.
* Kad korisnik unese vrijednosti, pronađite sve super junake koji zadovoljavaju uvjet da im je „Total“ između min i max u vašem odabranom kontejneru pa ih prekopirajte ih u vektor *v*.
* Na kraju ispišite sadržaj vektora *v*.

1. (Bodova: **3**) Kad korisnik odabere „Prikaz po strani“, koristeći prioritetni red ispišite sve super junake prema njihovoj strani, prvo dobre, a zatim zle. Super junake iste strane obavezno prikažite abecedno padajuće prema imenu.

### Ishod 4 (bodova: 17)

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

Insertion vs Shell

Lets play guess

Monkey Play

The Quick Dance

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

* Kreirajte dva pomoćna vektora *v1* i *v2* i iskopirajte u svako od njih sve super junake.
* Uzmite s predavanja Insertion i Shell sortiranja i promijenite ih tako da sortiraju super junake, padajuće po imenu.
* Sortirajte vektor *v1* koristeći Insertion sort.
* Sortirajte vektor *v2* koristeći Shell 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 „Lets play guess“, 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 „Monkey Play“, slučajnim odabirom prekopirajte šest super junaka u novi vektor. Isprogramirajte svoju vlastitu determinističku verziju bogo sorta te je iskoristite za slaganje super junaka po imenu u padajućem redoslijedu. Ispišite u tekstualnu datoteku svaku isprobanu permutaciju.
4. (Bodova: **4**) Kad korisnik odabere opciju „The Quick Dance“, pitajte ga da upiše broj *n* koji predstavlja broj elemenata. Nakon što upiše broj, u vektor ubacite slučajnim odabirom *n* super junaka. 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 imenu

Vizualiziraj

Izrada imena iz imena

1. (Bodova: **2**) Kad korisnik odabere „Pretraga po imenu“, prvo prekopirajte super junake iz vektora u odgovarajući kontejner. Zatim pitajte korisnika da upiše ime koje želi tražiti pa mu nakon toga ispišite podatke o super junaku s tim imenom ili ispišite da ne postoji. 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 super junake u unordered\_multimap tako da ključ bude njihov naziv. Nakon toga, vizualizirajte *hash* tablicu iz unordered\_multimap tako da iscrtate sve njene *buckete* te uz svaki *bucket* prikažete i super junake u njemu. Primjer početka ispisa (vrijednosti su izmišljene):

Bucket 0: Abraxas -- Blackwing -- Catwoman

Bucket 1: Deathlok -- Guardian -- Kraven the Hunter -- Nina Theroux

Bucket 2: [EMPTY]

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

* Ako je ime prvog super junaka „Killer Owl“, a drugog „Loki“, onda se ime drugog može konstruirati iz prvog.
* Ako je ime prvog super junaka „Lady Bullseye“, a drugog „Lady Bulls Eye“, onda se ime drugog ne može konstruirati iz prvog jer nedostaje jedan razmak.