

---

# PROGRAMSKI JEZICI 1 (24.4.2017.)

---

1. Napisati program koji omogućava rješavanje sljedećeg problema. Niz riječi učitava se sa standardnog ulaza dok se ne učitava riječ „x“. Transformirati taj ulaz na sljedeći način.
  - Napisati klasu **Reader** koja omogućava učitavanje niza riječi sa ulaznog toka i transformaciju tog niza u oblik **std::vector<std::string>**. Ulazni tok se proslijeđuje kroz konstruktor, a pozivom metode **getWords()** se podaci vraćaju u traženom obliku.
  - Napisati generičku klasu **Counter<T>**, privatno naslijeđenu iz klase **std::unordered\_set<T>**, koja omogućava prebrojavanje elemenata nekog objekta tipa **std::vector<T>** proslijeđenog kroz konstruktor. Prebrojavanje se vrši pozivom metode **count(const std::function<bool(T)>&)** koja treba da vrati uređenu trojku čiji elementi sadrže redom broj jedinstvenih elemenata u kolekciji, broj elemenata najdužeg rastućeg podniza i broj elemenata koji zadovoljavaju uslov definisan proslijeđenom lambda funkcijom.
    - Statički onemogućiti kompajliranje uz odgovarajuću poruku, ukoliko tip **T** ne zadovoljava uslove neophodne za izračunavanje pomenute uređene trojke.
  - Napisati klasu **Filter** koja omogućava filtriranje nekog stringa na osnovu lambda funkcije proslijeđene kroz konstruktor, pozivom funkcije **filter()**.
  - Napisati generičku funkciju **convertType<B>(const A&, const std::function<B(const A&)>&)** u prostoru imena **utils**, koja prihvata proizvoljan element tipa **A** i pretvara ga u **std::vector<B>**, prema definisanoj funkciji za svaki element proslijeđenog objekta, pretpostavljajući da će proslijeđeni objekat imati definisan iterator.
  - Iskoristiti instancu klase **Reader** za učitavanje traženog niza riječi, a potom konstruisati novi vektor stringova koji sadrži samo one učitane riječi čija je dužina parna.
  - Konvertovati taj vektor stringova u vektor vektora karaktera.
  - Konvertovati taj vektor u vektor uređenih trojki dobijenih korištenjem metoda iz klase **Counter** nad svakim vektorom karaktera, pri čemu posljednji element trojke treba da sadrži broj cifara u nizu karaktera.
  - Sortirati dobijeni vektor u rastućem poretaku prema prvom članu trojke, a potom elemente vektora upisati u fajl pod nazivom „result.txt“, pri čemu je za upis u fajl potrebno koristiti **for\_each** funkciju iz prostora imena **algorithm**.
2. Odgovoriti na sljedeće stavke.
  - 2.1. Napisati primjer funkcije koja vrši alokaciju niza od  $n$  elemenata, a u slučaju da alokacija baci izuzetak, ona ne dovodi do propagiranja greške kroz stek, već kao rezultat alokacije vraća prazan pokazivač.
  - 2.2. Neka je data klasa **A** sa metodom **a()** i klasa **B** izvedena iz **A**. Kakva mora da bude metoda **a()**, da bi se mogla pozvati nad objektom klase **B** iako se poziv vrši nad pokazivačem na klasu **A** koji pokazuje na dati objekat?
    - 2.2.1. Napisati primjer kopiranja neke kolekcije koja predstavlja niz elemenata klase **B** u kolekciju koja predstavlja elemente klase **A**, tako da se omogući polimorfizam i ne odsijeca sadržaj kopiranog elementa.
  - 2.3. Napisati proizvoljan četvorobajtni binarni podatak djeljiv sa 128 u obliku binarnog literala sa separacijom na svakoj četvrtoj cifri.
  - 2.4. Koje metode (naziv i semantika povratnog tipa) mora da ima klasa koje je instancirani objekat **xx** da bi se mogla koristiti **for** petlja sa sljedećom sintaksom: **for (auto& x : xx)?**
  - 2.5. Napisati primjer nasljeđivanja sa metodom sa kovarijantnim povratnim tipom.

## NAPOMENE:

- Minimizirati broj linija koda i izbjeći dupliranje koda.
- Ispravno označiti metode koje sigurno bacaju izuzetke i koje sigurno ne bacaju izuzetke, kao i konstante metode.
- Poštovati konvencije, razdvojiti interfejs od implementacije.