

Zadaca 5 iz predmeta Strukture podataka

Zadatak 1

Implementirati strukturu podataka **Heap** pomoću niza kao što je to objašnjeno na predavanju. Ta struktura treba da implementira sljedeće:

- Pravilno ponašanje konstruktora i destruktoru,
- ubacivanje novog elementa i njegovo uređenje kao max heap,
- metod max koji vraća referencu na maksimalni element,
- metod removeMax koji uklanja i vraća maksimalni element iz heapu,
- `operator<<` koji ispisuje heap koristeći preorder obilazak.

Napisati kratki main program u kojem se testiraju sve gore navedene funkcionalnosti.

Zadatak 2

Implementirati kontejner **Map** koji predstavlja asocijativni kontejner implementiran pristupom binarnog stabla pretrage.

Unutar jednog čvora ovog stabla, nalaze se dva polja, ključ i vrijednost. Interna organizacija stabla se radi po ključu. Indeksiranje i pretrage stabla se rade po ključu, ali dobiveni rezultat je polje vrijednost koje se nalazi u tom čvoru.

```
template <typename K, typename V>
class Map {
public:

    Map();
    Map(const Map&);
    Map(Map&&);
    Map& operator=(const Map&);
    Map& operator=(Map&&);
    ~Map();

    bool empty() const;
    size_t size() const;

    // Metod baca iznimku ako element pod kljucem k ne postoji
    V& at(const K&);
    const V& at(const K&) const;

    // Metod vraca nullptr ako element nije pronaden
    V* find(const K&);
    const V* find(const K&) const;
```

```

// Metod vraca referencu na element u stablu. Ako element ne
// postoji potrebno je ubaciti novi cvor pod kljucem k, sa default
// konstruisanom vrijednosti v (stoga ovaj metod ne moze biti const)
V& operator[] (const K&);

// Ubacuje novi cvor u stablo pod kljucem k. Vrijednost se konstruise
// koristeći forward referencu. Potrebno je baciti iznimku ako vrijednost
// pod tim kljucem vec postoji
template <F>
insert(const K&, F&&);

// Vraca true ako je vrijednost izbrisana iz stabla, false ako ta vrijednost
// ne postoji
bool erase(const K&);
void clear();

};

```

Zadatak 3

Potrebno je implementirati program koji će analizirati tekst na način da ispiše sve riječi iz teksta zajedno sa pozicijama na kojima se one pojavljuju. Ispis treba da bude sortiran abecedno. Prvo je potrebno ispisati riječ, a zatim u novom redu sve pozicije na kojima se ta riječ nalazi u tekstu.

U prilogu zadatke se nalazi datoteka story.txt. Pri pokretanju programa ispis bi trebao da bude sličan ovom:

```

Rijec je 'a' i nalazi se na sljedećim pozicijama:
231,
Rijec je 'adresa' i nalazi se na sljedećim pozicijama:
75,
Rijec je 'ako' i nalazi se na sljedećim pozicijama:
163,

// dio teksta je preskočen

Rijec je 'jelom' i nalazi se na sljedećim pozicijama:
272,
Rijec je 'jež' i nalazi se na sljedećim pozicijama:
141, 264,
Rijec je 'ježa' i nalazi se na sljedećim pozicijama:
29, 82,
Rijec je 'ježić' i nalazi se na sljedećim pozicijama:
5, 61, 155, 191,

// ostatak teksta je preskočen

```