

# Programiranje 2

## Binarna pretraživačka stabla

### 1 ZADACI SA ČASA

**Zadatak 1** • Napisati funkciju `Cvor* pretrazi_stablo(Cvor* koren, int broj)` koja proverava da li se ceo broj `broj` nalazi u stablu sa korenom `koren`. Funkcija vraća pokazivač na čvor stabla koji sadrži traženu vrednost ili `NULL` ukoliko takav čvor ne postoji.

- Napisati funkciju `Cvor* pronadji_najmanji(Cvor* koren)` koja pronalazi čvor koji sadrži najmanju vrednost u stablu sa korenom `koren`.
- Napisati funkciju `Cvor* pronadji_najveci(Cvor* koren)` koja pronalazi čvor koji sadrži najveću vrednost u stablu sa korenom `koren`.

Napisati program koji sa standardnog ulaza učitava cele brojeve sve do kraja ulaza, dodaje ih u binarno pretraživačko stablo i ispisuje uneto stablo na standardni izlaz. Zatim omogućiti unos još dva cela broja i demonstrirati rad funkcije za pretragu nad prvim unetim brojem i rad funkcije za brisanje elemenata nad drugim unetim brojem. **Napomena: Koristiti biblioteku za rad sa stablima.**

```
Primer 1:  
7 2 1 9 32 18  
  
Ispis: 1 2 7 9 18 32  
Trazi se broj: 11  
Broj se ne nalazi u stablu!  
Najmanji: 1  
Najveci: 32
```

**Zadatak 2** Dva binarna stabla su identična ako su ista po strukturi i sadržaju tj. ako oba korena imaju isti sadržaj i identična odgovarajuća podstabla. Napisati funkciju `int identitet(Cvor* koren1, Cvor* koren2)` koja proverava da li su binarna stabla `koren1` i `koren2` koja sadrže cele brojeve identična, a zatim i program koji testira njen rad. Elemente pojedinačnih stabla unositi sa standardnog ulaza sve do pojave broja 0.

U slučaju greške na standardni izlaz za greške ispisati -1.

**Napomena: Koristiti biblioteku za rad sa stablima.**

```
Primer 1:          Primer 2:  
1 4 2 6 3 1 4 2 1 0  1 4 2 6 3 1 4 2 1 0  
1 4 2 6 3 1 4 2 1 0  1 4 2 6 3 21 4 2 11 0  
  
Da                 Ne
```

**Zadatak 3** Dato je binarno pretraživačko stablo celih brojeva.

- Napisati funkciju koja izračunava broj čvorova stabla.

- Napisati funkciju koja izračunava broj listova stabla.
- Napisati funkciju koja štampa pozitivne vrednosti listova stabla.
- Napisati funkciju koja izračunava zbir čvorova stabla.
- Napisati funkciju koja izračunava najveći element stabla.
- Napisati funkciju koja izračunava dubinu stabla.
- Napisati funkciju koja izračunava broj čvorova na  $i$ -tom nivou stabla.
- Napisati funkciju koja ispisuje sve elemente na  $i$ -tom nivou stabla.
- Napisati funkciju koja izračunava maksimalnu vrednost na  $i$ -tom nivou stabla.
- Napisati funkciju koja izračunava zbir čvorova na  $i$ -tom nivou stabla.
- Napisati funkciju koja izračunava zbir svih vrednosti stabla koje su manje ili jednake od date vrednosti  $x$ .

Napisati program koji testira prethodne funkcije. Stablo formirati na osnovu vrednosti koje se unose sa standardnog ulaza, sve do kraja ulaza, a vrednosti parametara  $i$  i  $x$  pročitati kao argumente komandne linije.

U slučaju greške na standardni izlaz za greške ispisati -1.

**Napomena: Koristiti biblioteku za rad sa stablima.**

```
Primer 1:
./a.out 2 15

10 5 15 3 2 4 30 12 14 13

Broj cvorova: 10
Broj listova: 4
Pozitivni listovi: 2 4 13 30
Zbir cvorova: 108
Najveci element: 30
Dubina stabla: 5
Broj cvorova na 2. nivou: 3
Elementi na 2. nivou: 3 12 30
Maksimalni element na 2. nivou: 30
Zbir elemenata na 2. nivou: 45
Zbir elemenata manjih ili jednakih od 15:
78
```

**Zadatak 4** Napisati program koji ispisuje sadržaj binarnog pretraživačkog stabla po nivoima. Elementi stabla se učitavaju sa standardnog ulaza sve do kraja ulaza.

U slučaju greške na standardni izlaz za greške ispisati -1.

**Napomena: Koristiti biblioteku za rad sa stablima.**

```
Primer 1:          Primer 2:
10 5 15 3 2 4 30 12 14 13    6 11 8 3 -2

0.nivo: 10                0.nivo: 6
1.nivo: 5 15              1.nivo: 3 11
2.nivo: 3 12 30           2.nivo: -2 8
3.nivo: 2 4 14
4.nivo: 13
```

**Zadatak 5** Elementi binarnog pretraživačkog stabla su celi brojevi i učitavaju se sa standardnog ulaza sve do kraja ulaza. Odrediti broj čvorova stabla koji su jednaki zbiru svojih suseda. Smatramo da su susedi nekom čvoru njegovi direktni potomci i njegov prvi predak. Koren nema pretka, pa su njegovi susedi samo njegovi direktni potomci. Čvor koji je list nema potomke, pa je njegov sused samo njegov direktan predak. Rezultat ispisati na standardni izlaz.

U slučaju greške na standardni izlaz za greške ispisati -1.

**Napomena: Koristiti biblioteku za rad sa stablima.**

<b>Primer 1:</b> 19 -2 -20 -1 -8 21 21 3	<b>Primer 2:</b> 50 40 -10 20 -20 -30 -45 60 70 80 90 95 1	<b>Primer 3:</b> 7 0	<b>Primer 4:</b> 0
--	--	----------------------------	-----------------------

**Zadatak 6** U datoteci izraz.txt je zapisan aritmetički izraz u prefiksnoj notaciji. Izraz sadrži aritmetičke operatore +, -, \*, / i cele pozitivne brojeve. Napraviti aritmetičko stablo koje opisuje dati izraz:

- Napraviti strukturu koja opisuje aritmetičko stablo.
- Napsati funkciju Cvor\* ucitaj(FILE\* f) koja iz datoteke f učitava aritmetičko stablo i vraća adresu korena.
- Napisati funkciju void ispis(Cvor\* stablo) koja ispisuje stablo u infiksnom poretku.
- Napisati funkciju void oslobodi(Cvor\* stablo) koja oslobadja prostor koji zauzimaju elementi stabla.
- Napisati funkciju float vrednost(Cvor\* stablo) koja računa vrednost izraza zapisanog u datom aritmetičkom stablu.

Napisane funkcije testirati pozivom iz main-a.

U slučaju greške na standardni izlaz za greške ispisati -1.

<b>Primer 1:</b> izraz.txt: + 3 - * 2 5 / 4 2 11.67	<b>Primer 2:</b> izraz.txt: - * / 15 - 7 + 1 1 3 + 2 + 1 1 5.00
--	--

**Zadatak 7** Napisati funkciju koja iz datoteke karakteri.txt učitava karaktere i smešta ih u binarno pretraživačko stablo. Uz svaki karakter čuvati i broj pojavljivanja karaktera u datoteci. Ispisati na standardni izlaz karakter koji se pojavljuje najveći broj puta u datoteci i odgovarajući broj pojavljivanja.

U slučaju greške na standardni izlaz za greške ispisati -1.

<b>Primer 1:</b> karakter.txt: Danas je lep dan. 'a': 3	<b>Primer 2:</b> karakter.txt: U Datoteci je probni test za Zadatke sa ispita. ' ': 7
--	---

## 2 DOMAĆI ZADACI

**Zadatak 8** Napisati funkciju `void sortiraj(int* a, int n)` koja sortira niz celih brojeva  $a$  dimenzije  $n$  korišćenjem binarnog pretraživačkog stabla. Napisati i program koji sa standardnog ulaza učitava ceo broj  $n$  manji od 50 i niz  $a$  celih brojeva dužine  $n$ , poziva funkciju `sortiraj` i rezultat ispisuje na standardni izlaz.

U slučaju greške na standardni izlaz za greške ispisati -1.

**Napomena:** Koristiti biblioteku za rad sa stablima.

```
Primer 1:  
7  
1 11 8 6 37 25 30  
1 6 8 11 25 30 37
```

**Zadatak 9** AVL-stablo je binarno pretraživačko stablo kod kojeg apsolutna razlika visina levog i desnog podstabla svakog elementa nije veća od jedan. Napisati funkciju `int avl(Cvor * koren)` koja izračunava broj čvorova stabla sa korenom `koren` koji ispunjavaju uslov za AVL stablo. Napisati zatim i program koji ispisuje rezultat `avl` funkcije za stablo čiji se elementi unose sa standardnog ulaza sve do kraja ulaza. U slučaju greške na standardni izlaz za greške ispisati -1.

**Napomena:** Koristiti biblioteku za rad sa stablima.

```
Primer 1:      Primer 2:  
10 5 15 2 11 16 1 13      16 30 40 24 10 18 45 22  
7                      6
```

**Zadatak 10** Napisati funkciju koja za svaki čvor u stablu menja redosled njegovog levog i desnog direktnog potomka ukoliko levo podstablo ima veću dubinu od desnog podstabla. Ispisati dobijeno stablo na izlazu. Dubina predstavlja najduži put od korena do lista. Kreirati glavni program koji učitava stablo, poziva napisanu funkciju i ispisuje dobijeno stablo na izlaz.

U slučaju greške na standardni izlaz za greške ispisati -1.

**Napomena:** Koristiti biblioteku za rad sa stablima.

```
Primer 1:      Primer 2:      Primer 3:      Primer 4:  
20 10 30 5 12 7 0      10 5 15 3 1 0      20 10 30 9 15 25 12 17 0      10 5 3 8 7 0  
30 20 12 10 5 7      15 10 5 3 1      30 25 20 9 10 12 15 17      10 3 5 8 7
```

**Zadatak 11** Dva binarna stabla su slična kao u ogledalu ako su ili oba prazna ili ako oba nisu prazna i levo podstablo svakog stabla je slično kao u ogledalu desnom podstablu onog drugog (bitna je struktura stabala, ali ne i njihov sadržaj). Napisati funkciju koja proverava da li su dva binarna pretraživačka stabla slična kao u ogledalu, a potom i program koji testira rad funkcije nad stablima čiji se elementi unose sa standardnog ulaza sve do unosa broja 0 i to redom za prvo stablo, pa zatim i za drugo stablo.

U slučaju greške na standardni izlaz za greške ispisati -1.

**Napomena:** Koristiti biblioteku za rad sa stablima.

```
Primer 1:      Primer 2:  
11 20 5 3 0      11 20 5 3 0  
8 14 30 1 0      8 20 15 0  
Jesu.           Nisu.
```

**Zadatak 12** *Napisati program koji iz datoteke čije se ime zadaje kao prvi argument komandne linije čita reči i formira binarno uređeno stablo (uređeno leksikografski) koje u jednom čvoru sadržati reč koja se pojavljuje i broj pojavljivanja te reči u datoteci. Ispisati reč koja se pojavljuje najčešće u datoteci.*