

1 Liste i binarno pretraživačko stablo - zadaci za vežbanje

1. Biblioteku iz prvog zadatka proširiti rekurzivnim implementacijama narednih funkcija:

- (a) `void ispisi_listu_rekurzivno(Cvor * glava)` koja ispisuje vrednosti u čvorovima liste.
- (b) `int velicina_liste_rekurzivno(Cvor * glava)` koja vraća broj elemenata liste.
- (c) `int broj_neparnih_rekurzivno(Cvor * glava)` koja vraća broj neparnih elemenata liste.
- (d) `int proizvod_liste_rekurzivno(Cvor * glava)` koja računa proizvod svih elemenata liste.
- (e) `int zbir_liste_rekurzivno(Cvor * glava)` koja računa zbir svih elemenata liste.
- (f) `int minimum_rekurzivno(Cvor * glava)` koja vraća najmanju vrednost koja se javlja u listi.
- (g) `int maximum_rekurzivno(Cvor * glava)` koja vraća najveću vrednost koja se javlja u listi.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
ULAZ:
Unesite elemente liste (Ctrl+D za kraj unosa):
1 -2 3 8 1 9 7 -8 0
IZLAZ:
Lista:
1 -> -2 -> 3 -> 8 -> 1 -> 9 -> 7 -> -8 -> 0
Broj elemenata liste: 9
Broj neparnih elemenata liste: 5
Proizvod elemenata liste: 0
Zbir elemenata liste: 19
Minimum: -8
Maksimum: 9
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
ULAZ:
Unesite elemente liste (Ctrl+D za kraj unosa):
-4 3 2 1 -4
IZLAZ:
Lista:
-4 -> 3 -> 2 -> 1 -> -4
Broj elemenata liste: 0
Broj neparnih elemenata liste: 2
Proizvod elemenata liste: 96
Zbir elemenata liste: -2
Minimum: -4
Maksimum: 3
```

2. Napisati biblioteku za rad sa binarnim pretraživačkim stablima.

- (a) Definirati strukturu `Cvor` kojom se opisuje čvor stabla, a koja sadrži karakter `slovo` i pokazivače `levo` i `desno` redom na levo i desno podstablo.
- (b) Napisati funkciju `Cvor *napravi_cvor(char slovo)` koja alokira memoriju za novi čvor stabla i vrši njegovu inicijalizaciju zadatim karakterom `slovo`.
- (c) Napisati funkciju `int dodaj_u_stablo(Cvor ** adresa_korena, char slovo)` koja u stablo na koje pokazuje argument `adresa_korena` dodaje karakter `slovo`. Povratna vrednost funkcije je 0 ako je dodavanje uspešno, odnosno 1 ukoliko je došlo do greške.
- (d) Napisati funkciju `Cvor *pretrazi_stablo(Cvor * koren, char slovo)` koja proverava da li se karakter `slovo` nalazi u stablu sa korenom `koren`. Funkcija vraća pokazivač na čvor stabla koji sadrži traženu vrednost ili `NULL` ukoliko takav čvor ne postoji.
- (e) Napisati funkciju `void obrisi_element(Cvor ** adresa_korena, char slovo)` koja briše čvor koji sadrži karakter `slovo` iz stabla na koje pokazuje argument `adresa_korena`.
- (f) Napisati funkciju `void ispisi_stablo_infiksno(Cvor * koren)` koja infiksno ispisuje sadržaj stabla sa korenom `koren`. Infiksni ispis podrazumeva ispis levog podstabla, korena, a zatim i desnog podstabla.
- (g) Napisati funkciju `void ispisi_stablo_prefiksno(Cvor * koren)` koja prefiksno ispisuje sadržaj stabla sa korenom `koren`. Prefiksni ispis podrazumeva ispis korena, levog podstabla, a zatim i desnog podstabla.
- (h) Napisati funkciju `void ispisi_stablo_postfiksno(Cvor * koren)` koja postfiksno ispisuje sadržaj stabla sa korenom `koren`. Postfiksni ispis podrazumeva ispis levog podstabla, desnog podstabla, a zatim i korena.
- (i) Napisati funkciju `void oslobodi_stablo(Cvor ** adresa_korena)` koja oslobađa memoriju zauzetu stablom na koje pokazuje argument `adresa_korena`.

Korišćenjem kreirane biblioteke, napisati program koji sa standardnog ulaza učitava karaktere sve dok se ne unese `!`, dodaje u binarno pretraživačko stablo samo one karaktere koji predstavljaju malo slovo abecede (a-z) i ispisuje stablo u svakoj od navedenih notacija. Zatim omogućiti unos još dva karaktera i demonstrirati rad funkcije za pretragu nad prvim unetim karakterom i rad funkcije za brisanje elemenata nad drugim unetim karakterom.

Primer 1

```

INTERAKCIJA SA PROGRAMOM:
ULAZ:
Unesite elemente stabla (! za kraj unosa):
b z m k d h !
IZLAZ:
Infiksno: b d h k m z
Prefiksno: b z m k d h
Postfiksno: h d k m z b
Uneti karakter za pretragu:
m
Karakter se nalazi u stablu.
Uneti karakter za brisanje:
h
Stablo nakon brisanja: b d k m z

```

Primer 2

```

INTERAKCIJA SA PROGRAMOM:
ULAZ:
Unesite elemente stabla (! za kraj unosa):
p h u z a a m !
IZLAZ:
Infiksno: a a h m p u z
Prefiksno: p h a a m u z
Postfiksno: a a m h z u p
Uneti karakter za pretragu:
q
Karakter se ne nalazi u stablu.
Uneti karakter za brisanje:
l
Stablo nakon brisanja: a a m h z u p

```

3. Napisati sledeće funkcije za rad sa binarnim stablima koja sadrže cele brojeve:

- Funkcija koja izračunava broj čvorova stabla.
- Funkcija koja izračunava broj listova stabla.
- Funkcija koja izračunava proizvod čvorova stabla.
- Funkcija koja izračunava broj prostih brojeva sadržanih u stablu.
- Funkcija koja izračunava najmanji element u stablu.
- Funkcija koja izračunava prosečnu vrednost čvorova.
- Funkcija koja računa proizvod svih vrednosti u binarnom stablu pretrage koje su manje ili jednake od vrednosti sadržane u korenu.

Napisati zatim i *main* funkciju koja testira navedene funkcije. Cele brojeve unositi sa standardnog ulaza sve dok se ne unese kraj ulaza (EOF). Ostale neophodne funkcije za rad sa stablima, kao i definiciju čvora stabla, iskoristiti iz priloženih fajlova *stabla.h* i *stabla.c*.

Primer 1

```

INTERAKCIJA SA PROGRAMOM:
ULAZ:
Unesite elemente stabla (CTRL+D za kraj unosa):
6 3 12 1 4 10 5
IZLAZ:
Broj cvorova: 7
Broj listova: 3
Proizvod cvorova: 43200
Prostih brojeva: 3
Najmanji element: 1
Prosečna vrednost: 5.857
Proizvod svih elemenata cija je vrednost manja
od 6: 360

```

Primer 2

```

INTERAKCIJA SA PROGRAMOM:
ULAZ:
Unesite elemente stabla (CTRL+D za kraj unosa):
6 5 4 3 2 1
IZLAZ:
Broj cvorova: 6
Broj listova: 1
Proizvod cvorova: 720
Prostih brojeva: 4
Najmanji element: 1
Prosečna vrednost: 3.500
Proizvod svih elemenata cija je vrednost manja
od 6: 120

```

4. Napisati program koji izračunava i na standardni izlaz ispisuje broj pojavljivanja svakog od malih slova koja se javljaju u nizu niski. Niske se učitavaju iz datoteke čije se ime zadaje kao argument komandne linije. Program realizovati korišćenjem binarnog pretraživačkog stabla uređenog leksikografski po karakteristikama. Može se pretpostaviti da reči neće biti duže od 50 karaktera, kao i da neće biti više od 100 niski. Napraviti svoju biblioteku modifikovanjem biblioteke iz prvog zadatka.

Primer 1

```

NISKE.TXT
antananarive ananas ana nana
Poziv: ./a.out niske.txt
INTERAKCIJA SA PROGRAMOM:
'a': 11
'e': 1
'i': 1
'n': 8
'r': 1
's': 1
't': 1
'v': 1

```

Primer 2

```

KARAKTERI.TXT
abbb acdbb bABC acd zzz
Poziv: ./a.out karakteri.txt
INTERAKCIJA SA PROGRAMOM:
'a': 3
'b': 6
'c': 1
'd': 2
'z': 3

```

5. Napisati funkciju koja proverava da li za svako podstablo binarnog stabla celih brojeva važi da je vrednost koja se nalazi u korenu podstabla jednaka razlici vrednosti korena desnog podstabla i vrednosti korena levog podstabla.

Primer 1

INTERAKCIJA SA PROGRAMOM:

ULAZ:

Unesite elemente stabla (CTRL+D za kraj unosa):

6 3 12 1 4 10 5

IZLAZ:

Ne vazi osobina.

Primer 2

INTERAKCIJA SA PROGRAMOM:

ULAZ:

Unesite elemente stabla (CTRL+D za kraj unosa):

8 3 11 2 5 10 21 20 41

IZLAZ:

Vazi osobina.