Programiranje 2
Stabla – vežbanja razno

Zadatak 1 *Napisati biblioteku (*`stablo.h` *i* `stablo.c`*) za rad sa binarnim sortiranim pretraživačkim stablom koje sadrži cele brojeve:*

a) Napraviti strukturu podataka `cvor` *koja opisuje čvor binarnog stabla koja sadrži ceo broj (*`vrednost`*) i pokazivače na levo (*`levo`*) i desno (*`desno`*) podstablo.*

b) Napisati funkciju `cvor* novi(int broj)` *koja pravi novi čvor stabla, inicijalizuje njegove vrednosti i vraća adresu tog čvora.*

c) Napisati funkciju `cvor* ubaci_u_stablo(cvor* stablo, int broj)` *koja dodaje novi čvor u stablo tako da ono bude sortirano stablo.*

d) Napisati funkciju `cvor* unos(FILE* ulaz)` *koja učitava brojeve do kraja datoteke* `ulaz` *i sve brojeve smesta u binarno sortirano stablo i vraća adresu njegovog korena.*

e) Napisati funkciju `void ispis(cvor* stablo, FILE* izlaz)` *koja ispisuje elemente binarnog stabla u infiksnom poretkom u datoteku* `izlaz`*.*

f) Napisati funkciju `void ispis1(cvor* stablo, FILE* izlaz)` *koja ispisuje elemente binarnog stabla u prefiksnom poretkom u datoteku* `izlaz`*.*

g) Napisati funkciju `void ispis2(cvor* stablo, FILE* izlaz)` *koja ispisuje elemente binarnog stabla u postfiksnom poretkom u datoteku* `izlaz`*.*

h) Napisati funkciju `void oslobodi(cvor* stablo)` *koja oslobađa memoriju koju zauzimaju ělementi stabla.*

```
Primer 1:
    6
   / \
  3   12
 /
-1
unos: 6 3 -1 12

izlaz: -1 3 6 12
izlaz1: 6 3 -1 12
izlaz2: -1 3 12 6
```

```
Primer 2:
     10
    /  \
   5    23
  / \  /
 4  9  12
/
1
unos: 10 5 4 9 1 5 23 12

izlaz: 1 4 5 9 10 23 12
izlaz1: 10 5 4 1 9 23 12
izlaz2: 1 4 9 5 12 23 10
```

Zadatak 2 *Napisati funkciju* `int suma(cvor* stablo)` *koja izračunava sumu svih elemenata stabala.*

```
Primer 1:
    6
   / \
  3   12
 /
-1
unos: 6 3 -1 12

izlaz: 20
```

```
Primer 2:
       10
      /  \
     5    23
    / \  /
   4  9  12
  /
 1
 unos: 10 5 4 9 1 5 23 12

 izlaz: 64
```

Zadatak 3 *Napisati funkciju* `void ispis_listova(cvor* stablo)` *koja na standarni izlaz ispisuje listove datog stabala.*

```
Primer 1:
    6
   / \
  3   12
 /
-1
unos: 6 3 -112

izlaz: -1 12
```

```
Primer 2:
       10
      /  \
     5    23
    / \  /
   4  9  12
  /
 1
 unos: 10 5 4 9 1 5 23 12

 izlaz: 1 9 12
```

Zadatak 4 *Napisati funkciju* `int suma_listova(cvor* stablo)` *koja računa sumu listova datog pretraživačkog stabla.*

```
Primer 1:
    6
   / \
  3   12
 /
-1
unos: 6 3 -1 12

izlaz: 11
```

```
    Primer 2:
       10
      /  \
     5    23
    / \  /
   4  9  12
  /
 1
 unos: 10 5 4 9 1 5 23 12

 izlaz: 22
```

Zadatak 5 *Napisati funkciju* `int maks1(cvor* stablo)` *koja računa maksimalni element binarnog sortiranog pretraživakog stabla.*

```
Primer 1:
    6
   / \
  3   12
 /
-1
```

```
    Primer 2:
       10
      /  \
     5    23
    / \  /
   4  9  12
```

```
unos: 6 3 -1 12

izlaz: 12
```

```
 /
1
unos: 10 5 4 9 1 5 23 12

izlaz: 23
```

Zadatak 6 *Napisati funkcju* `int maks2(cvor* stablo)` *koja računa maksimalni element (proizvoljnog) binarnog stabla.*

```
Primer 1:
    6
   / \
  12   3
 /
-1
unos: 6 3 -1 12

izlaz: 12
```

```
Primer 2:
     10
    /  \
   5    23
  / \  /
 4  9  12
/
1
unos: 10 5 4 9 1 5 23 12

izlaz: 23
```

Zadatak 7 *Napisati funkciju* `int dubina(cvor* stablo)` *koja računa dubinu (visnu) datog stabla.*

```
Primer 1:
    6
   / \
  3   12
 /
-1
unos: 6 3 -1 12

izlaz: 3
```

```
Primer 2:
     10
    /  \
   5    23
  / \  /
 4  9  12
/
1
unos: 10 5 4 9 1 5 23 12

izlaz: 4
```

Zadatak 8 *Napisati funkcju* `int f8(cvor* drvo, int nivo)` *koja vraća broj elemenata drveta koji se nalaze na nivou* `nivo`*. Testirati pozivom u main-u.*

```
Primer 1:
      6
     / \
    4   12
   /   /  \
  3   10  32

nivo: 3

rez: 3
```

```
Primer 2:
  7
 / \
2   10
     \
      23

nivo: 3

rez: 1
```

```
------------------------------------------------------------
\textbf{Primer 3:
      10
        \
         2
```

```
Primer 4:}
      7
     / \
    4   20
```

```
                          /   /  \
nivo: 3                   3   15   23
                         /
rez: 0                  2

                          nivo: 2

                          rez: 2
```

Zadatak 9 *Napisati funkciju* `int f5(cvor* drvo)` *koja vraća broj elemenata stabla* `drvo` *koji su jednaki razlici svojih sinova.*

```
Primer 1:                              Primer 2:
    7                                         4
   / \                                       / \
  5   12                                    3   7
 /\   /\                                   /   /
1  6 10 15                                1   6
unos: 7 5 1 6 12 10 15                 unos: 4 3 1 7 6

izlaz: 2  (to su cvorovi 7 i 5)        izlaz: 1 (to je cvor 4)
-----------------------------------------------------------------------

\textbf{Primer 3:                                  Primer 4:}
    6                                        10
   / \                                      /  \
  3   12                                   5    23
 /                                        / \  /
-1                                       4  9  12
unos: 6 3 -1 12                         /
                                       1
izlaz: 0                               unos: 10 5 4 9 1 5 23 12

                                       izlaz: 1
```

Zadatak 10 *Funkcija vrši rekurzivnu rotaciju drveta oko svih čvorova, dakle dobija se odraz prvobitnog drveta u ogledalu. (ne kreirati novo stablo ili nove čvorove)*

```
Primer 1:
     7
   /  \
  2    10
 / \   / \
-4  5 9  17
       /
      12
       \
        13

izlaz:
     7
    / \
  10   2
```

```
 / \  / \
17  9 5  -4
 \
  12
 /
13
```

Zadatak 11 *Napisati funkciju* `int br_cvorova(cvor *koren)`*, koja računa broj čvorova u stablu koji su veći od svih svojih potomaka. Pretpostaviti da stablo nije sortirano.*

Zadatak 12 *Dva binarna stabla su identična ako su ista po strukturi i sadržaju, odnosno oba korena imaju isti sadržaj i njihova odgovarajuća podstabla su identična. Napistati funkciju koja proverava da li su dva binarna stabla identična.*

Zadatak 13 *Napisati funkciju* `int ravnomerno_izbalansirano(Cvor *stablo)` *koja proverava da li je stablo ravnomerno izbalansirano. Stablo je ravnomerno izbalansirano ako za svaki čvor važi da je pozitivna razlika između dubine levog i dubine desnog podstabla najviše* `1`*. Testirati funkciju pozivom u main-u, stablo se učitava sve dok se ne stigne do kraja datoteke. Ukoliko jeste izbalansirano ispisati* `1`*, a u suprotnom* `0`*.*

```
Primer 1:
     10
    /  \
   7    20
  / \   / \
 5  8  15  23

ulaz: 10 7 20 5 8 15 23
izlaz: 1
```

```
Primer 2:
     10
    /  \
   7    11
  /
 5
/
4

ulaz: 10 7 11 5 4
izlaz: 0
```

```
Primer 3:
ulaz:
izlaz: 1
```

```
Primer 4:
     10
    /  \
   7    20
  / \
 5   8

ulaz: 10 7 5 8 20
izlaz: 1
```

Zadatak 14 *Napisati funkciju* `int prebroj(cvor* drvo)` *koja vraća broj elemenata stabla drvo koji su iste parnosti kao oba svoja sina (sva tri parna ili neparna). Ukoliko je čvor list ili ima samo jednog sina ne ulazi u zbir.*

Zadatak 15 *Napisati funkciju* `int f5(cvor* stablo)` *koja u datom stablu određuje broj onih elemenata kod kojih je zbir cifara svih elemenata levog podstabla strogo veći od zbira cifara svih elemenata desnog podstabla. Testirati funkciju pozivom u main-u. Elementi stabla su celi pozitivni brojevi.*

Primer 1:	**Primer 2:**	**Primer 3:**	**Primer 4:**
52 38 64 21 40 55 103 88 0	304 0	104 88 110 78 99 105 120 55 0	1111 -100 0
1	0	4	1

Zadatak 16 *Napisati funkciju* `int f5(cvor* s, int k)` *koja računa zbir svih parnih elemenata stabla* `s` *na nivou* `k`*, umanjen za zbir svih neparnih elemenata stabla* `s` *na nivou* `k`*. Glavni program učitava stablo i ceo broj* `k`*. Potrebno je ispisati rezultat funkcije* `f5` *na standardni izlaz. U slučaju greške ispisati -1 na standardni izlaz.*

Ulaz	20 10 30 5 12 7 3	10 5 15 3 1 2	20 10 30 9 15 25 12 17 3	10 5 3 8 7 10
Izlaz	7	-20	-49	0

Zadatak 17 *Napisati funkciju* `cvor* f5(cvor* s)` *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 funkciju* `f5` *i ispisuje dobijeno stablo na izlaz. U slučaju greške ispisati* -1 *na standardni izlaz.*

Ulaz	20 10 30 5 12 7	10 5 15 3 1	20 10 30 9 15 25 12 17	10 5 3 8 7
Izlaz	30 20 12 10 5 7	15 10 5 3 1	30 25 20 9 10 12 15 17	10 3 5 8 7