

1. Napisati program koji prvo pročita koliko redaka ima datoteka, tj. koliko ima studenata zapisanih u datoteci. Nakon toga potrebno je dinamički alocirati prostor za niz struktura studenata (ime, prezime, bodovi) i učitati iz datoteke sve zapise. Na ekran ispisati ime, prezime, apsolutni i relativni broj bodova.

Napomena: Svaki redak datoteke sadrži ime i prezime studenta, te broj bodova na kolokviju.
$$\text{relativan_br_bodova} = \text{br_bodova} / \text{max_br_bodova} * 100$$

2. Definirati strukturu osoba (ime, prezime, godina rođenja) i napisati program koji:

- A. dinamički dodaje novi element na početak liste,
- B. ispisuje listu,
- C. dinamički dodaje novi element na kraj liste,
- D. pronalazi element u listi (po prezimenu),
- E. briše određeni element iz liste,

U zadatku se ne smiju koristiti globalne varijable.

3. Prethodnom zadatku dodati funkcije:

- A. dinamički dodaje novi element iza određenog elementa,
- B. dinamički dodaje novi element ispred određenog elementa,
- C. sortira listu po prezimenima osoba,
- D. upisuje listu u datoteku,
- E. čita listu iz datoteke.

4. Napisati program za zbrajanje i množenje polinoma. Koeficijenti i eksponenti se čitaju iz datoteke.

Napomena: Eksponenti u datoteci nisu nužno sortirani.

5. Napisati program koji iz datoteke čita postfiks izraz i zatim korištenjem stoga računa rezultat. Stog je potrebno realizirati preko vezane liste.

6. Napisati program koji čita datoteku *racuni.txt* u kojoj su zapisani nazivi svih datoteka koji predstavljaju pojedini račun. Na početku svake datoteke je zapisan datum u kojem vremenu je račun izdat u formatu *YYYY-MM-DD*. Svaki sljedeći red u datoteci predstavlja artikl u formatu *naziv, količina, cijena*. Potrebno je formirati vezanu listu računa sortiranu po datumu. Svaki čvor vezane liste sadržava vezanu listu artikala sortiranu po nazivu artikla. Nakon toga potrebno je omogućiti upit kojim će korisnik saznati koliko je novaca sveukupno potrošeno na specifični artikl u određenom vremenskom razdoblju i u kojoj je količini isti kupljen.

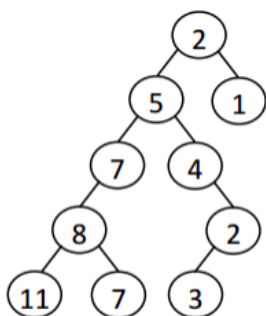
7. Napisati program koji pomoću vezanih listi (stabala) predstavlja strukturu direktorija.

Omogućiti unos novih direktorija i pod-direktorija, ispis sadržaja direktorija i povratak u prethodni direktorij. Točnije program treba preko menija simulirati korištenje DOS naredbi: 1- "md", 2 - "cd dir", 3 - "cd..", 4 - "dir" i 5 – izlaz.

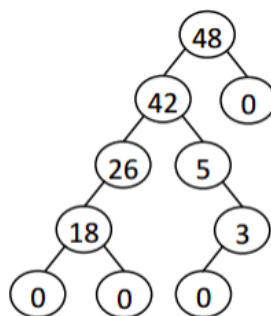
8. Napisati program koji omogućava rad s binarnim stablom pretraživanja. Treba omogućiti unošenje novog elementa u stablo, ispis elemenata (inorder, preorder, postorder i level order), brisanje i pronalaženje nekog elementa.

9. Zadan je niz brojeva 2, 5, 7, 8, 11, 1, 4, 2, 3, 7 koji su spremljeni u čvorove binarnog stabla.

- Napisati funkciju **insert** koja dodaje element u stablo tako da se pozivima te funkcije za sve elemente zadanog niza brojeva stvori stablo kao na slici Slika 1. Funkcije vraća pokazivač na korijen stabla.
- Napisati funkciju **replace** koja će svaki element stabla zamijeniti sumom elemenata u njegovom lijevom i desnom podstablu (tj. sumom svih potomaka **prije zamjene** vrijednosti u tim potomcima). Npr. stablo sa slike Slika 1 transformirat će se u stablo na slici Slika 2.
- Prepraviti program na način da umjesto predefiniranog cjelobrojnog polja korištenjem funkcije **rand()** generira slučajne brojeve u rasponu <10, 90>. Također, potrebno je upisati u datoteku sve brojeve u *inorder* prolasku nakon korištenja funkcije iz a), zatim b) dijela zadatka.



Slika 1



Slika 2

10. Napisati program koji čita datoteku *drzave.txt* u kojoj su zapisani nazivi pojedinih država. Uz ime države u datoteci se nalazi i ime dodatne datoteke u kojoj se nalaze gradovi pojedine države. Svaka datoteka koja predstavlja državu sadrži popis gradova u formatu *naziv_grada, broj_stanovnika*.

- Potrebno je formirati sortiranu vezanu listu država po nazivu. Svaki čvor vezane liste sadrži stablo gradova sortirano po broju stanovnika, zatim po nazivu grada.
- Potrebno je formirati stablo država sortirano po nazivu. Svaki čvor stabla sadrži vezanu listu gradova sortiranu po broju stanovnika, zatim po nazivu grada.

Nakon formiranja podataka potrebno je ispisati države i gradove te omogućiti korisniku putem tastature pretragu gradova određene države koji imaju broj stanovnika veći od unosa na tastaturi.

11. Prepraviti zadatak 10 na način da se formira hash tablica država. Tablica ima 11 mjesta, a funkcija za preslikavanje ključ računa da se zbraja ASCII vrijednost prvih pet slova države zatim računa ostatak cjelobrojnog dijeljenja te vrijednosti s veličinom tablice. Države s istim ključem se pohranjuju u vezanu listu sortiranu po nazivu države. Svaki čvor vezane liste sadrži stablo gradova sortirano po broju stanovnika, zatim po nazivu grada.