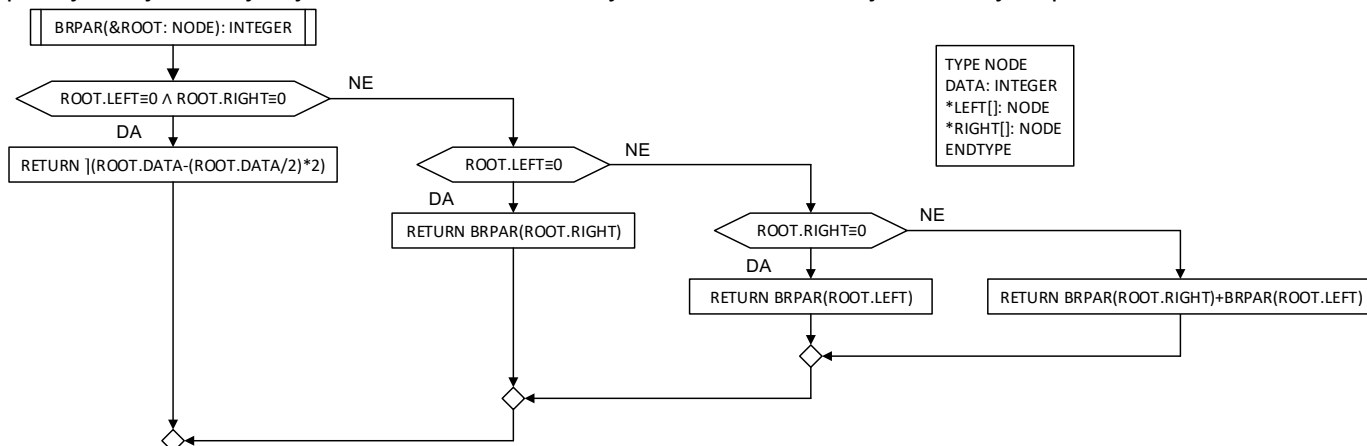
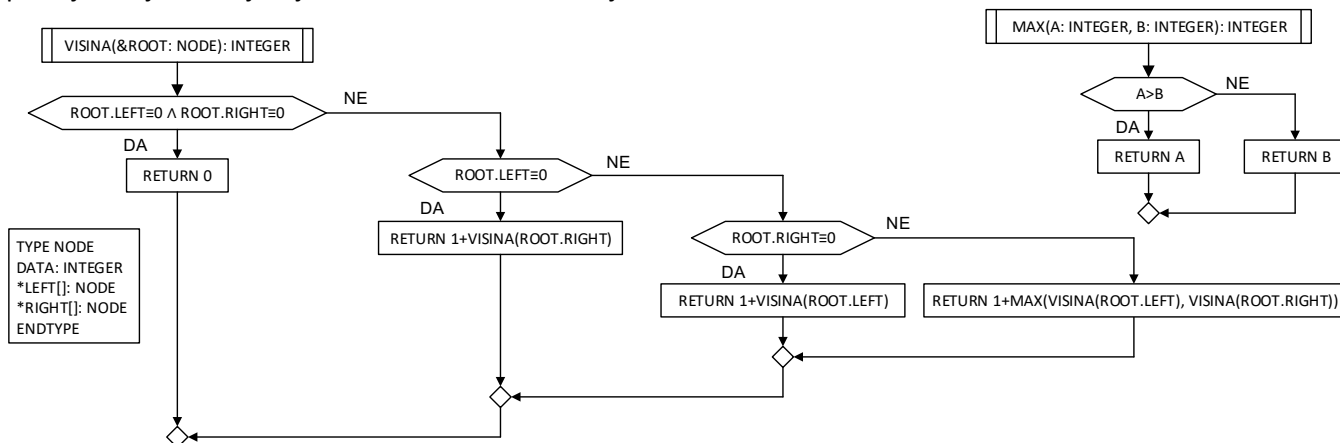


Računske vježbe 10

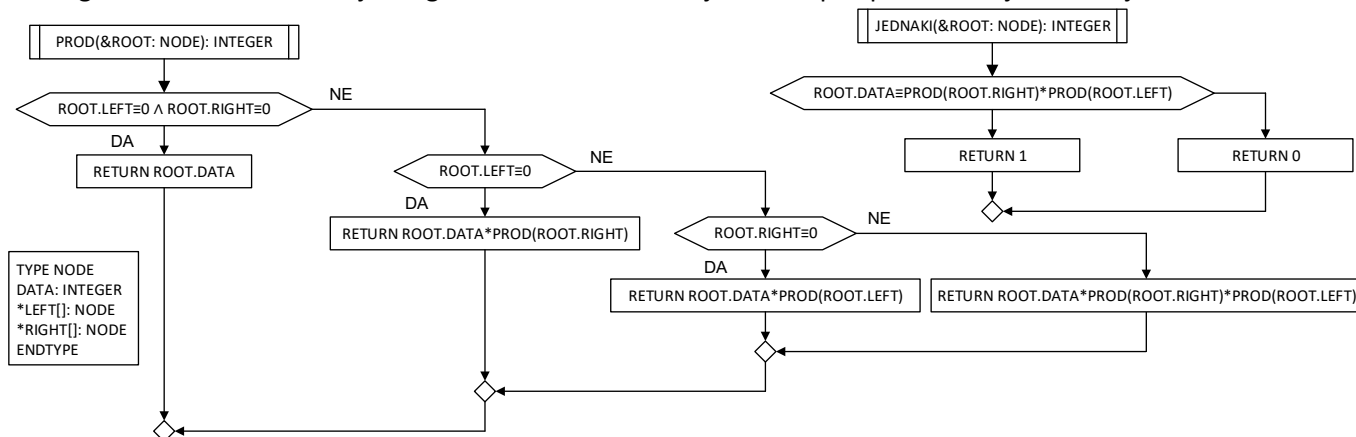
1. Pretpostaviti da je formirano binarno stablo, preko struktura sa po jednim cijelim brojem. Korijen stabla prosljeđen je funkciji koju treba realizovati. Funkcija treba da odredi broj listova koji su parni.



2. Pretpostaviti da je formirano binarno stablo, preko struktura sa po jednim cijelim brojem. Korijen stabla prosljeđen je funkciji koju treba realizovati. Funkcija treba da odredi visinu drveta.



3. Pretpostaviti da je formirano binarno stablo, preko struktura sa po jednim cijelim brojem. Korijen stabla prosljeđen je funkciji koju treba realizovati. Funkcija treba da odredi da li je vrijednost upisana u korijenu jednaka proizvodu vrijednosti upisanih u ostalim čvorovima. Pretpostaviti da korijen stabla ima i lijevog i desnog sina. Kako bi se funkcija mogla modifikovati u slučaju da ova pretpostavka nije zadovoljena?



4. Kreirati klasu **VEKTOR** koja ima podatke članove **N** (cio broj, predstavlja dužinu vektora) i **X** (niz realnih brojeva, maksimalne dužine 100). Implementirati neophodne konstruktore, mutatore i inspektore, kao i metode za sabiranje, oduzimanje i kvadriranje vektora elemenata vektora. U slučaju da se metode za sabiranje ili oduzimanje vektora pozovu za vektore različitih dužina, metoda treba da vrati vektor čija je dužina jednaka -1.

Realizacija klase:

```

CLASS VEKTOR
N: INTEGER
X[100]: FLOAT
VEKTOR()
VEKTOR(A: FLOAT)
KOPIRAJ(P: VEKTOR)
POSTAVI(I: INTEGER, A: FLOAT)
CONST STAMPAJ()
CONST SABERI(P: VEKTOR): VEKTOR
CONST ODUZMI(P: VEKTOR): VEKTOR
CONST KVADRIRAJ(): VEKTOR
ENDCLASS

```

Realizacija metoda:

VEKTOR() FROM VEKTOR I: INTEGER N=10 I=1 WHILE I≤N X[I]=0 I=I+1 ENDWHILE	VEKTOR(A:FLOAT) FROM VEKTOR I: INTEGER N=10 I=1 WHILE I≤N X[I]=A I=I+1 ENDWHILE	KOPIRAJ(P: VEKTOR) FROM VEKTOR I: INTEGER N=P.N I=1 WHILE I≤N X[I]=P.X[I] I=I+1 ENDWHILE
POSTAVI(I: INTEGER, A: FLOAT) FROM VEKTOR X[I]=A	STAMPAJ() FROM VEKTOR I: INTEGER I=1 WHILE I≤N OUTPUT X[I] I=I+1 ENDWHILE	
SABERI(P: VEKTOR): VEKTOR FROM VEKTOR I: INTEGER Q: VEKTOR IF P.N≠N Q.N=-1 RETURN Q ENDIF Q.N=N I=1 WHILE I≤N Q.X[I]=X[I]+P.X[I] I=I+1 ENDWHILE RETURN Q	KVADRIRAJ(): VEKTOR FROM VEKTOR I: INTEGER Q: VEKTOR Q.N=N I=1 WHILE I≤N Q.X[I]=X[I]*X[I] I=I+1 ENDWHILE RETURN Q	

5. Kreirati klasu **STUDENT** koja ima podatke članove: **IME** (string, ime studenta), **PREZIME** (string, prezime studenta), **GU** (cio broj, godina upisa), **GS** (cio broj, godina studija), **BPI** (cio broj, broj položenih ispita) i **PROSJEK** (realan broj, prosječna ocjena). Implementirati neophodne konstruktore, mutatore i inspektore. Realizovati i metodu članicu **PROVJERA** koja provjerava da li su podaci o studentu pravilno uneseni, tj. da ne postoje studenti upisani prije 1960. godine, a ni nakon 2023. godine, da ne postoje studenti koji studiraju na godini studija manjoj od 1 niti većoj od 5, da broj položenih ispita ne može biti manji od 0 niti veći od 40, i da prosjek nije van opsega od 6 do 10.

Realizacija klase:

```

CLASS STUDENT
IME[20], PREZIME[20]: CHAR
GU, GS, BPI: INTEGER
PROSJEK: FLOAT
STUDENT()
STUDENT(A[:CHAR, B[:CHAR, C: INTEGER, D: INTEGER, E: INTEGER, F:FLOAT)
CONST STAMP_STUD()
CONST DAJ_BPI(): INTEGER
CONST DAJ_PROS(): FLOAT
POV_GS()
AZURIRANJE(A: INTEGER)
PROVJERA(): INTEGER
ENDCLASS

```

Realizacija metoda:

STUDENT() FROM STUDENT GU=0 GS=0 BPI=0 PROS=0	STUDENT(A[: CHAR, B[: CHAR, C: INTEGER, D: INTEGER, E: INTEGER, F: FLOAT) FROM STUDENT I: INTEGER I=1 WHILE A[I]≠'\0' IME[I]=A[I] I=I+1 ENDWHILE IME[I]= '\0' I=1 WHILE B[I]≠'\0' PREZIME[I]=B[I] I=I+1 ENDWHILE PREZIME[I]='\0' GU=C GS=D BPI=F	
DAJ_BPI(): INTEGER FROM STUDENT RETURN BPI	DAJ_PROS(): FLOAT FROM STUDENT RETURN PROS	POV_GS() FROM STUDENT GS=GS+1
AZURIRANJE(A: INTEGER) FROM STUDENT PROSJEK=(PROSJEK*BPI+A) / (BPI+1) BPI=BPI+1	PROVJERA(): INTEGER FROM STUDENT IF (GU<1960 V GU>2003 V BPI<0 V BPI>40 V GS<1 V GS>5 V PROSJEK<6 V PROSJEK>10) RETURN 0 ELSE RETURN 1 ENDIF	