

4. Algoritmi, dijagrami toka, pseudoprogrami

4.1. Uvod

Algoritam je postupak ili pravilo koje se sastoji od opisa konačnog skupa koraka. Svaki od njih sadrži jednu ili više operacija. Glavne osobine algoritama su sljedeće:

- svaki algoritam ima početak i kraj,
- algoritmom se transformira i kreira određena struktura podataka,
- algoritam često koristi rekurzivno-iterativne postupke (operacije),
- algoritam mora obuhvatiti sva alternativna rješenja kako bi mogao prihvatiti razne vrijednosti ulaznih podataka.

Formiranje (pripremanje) algoritama je najkreativniji dio u procesu korištenja računala. Postoje dva načina predstavljanja algoritama. Najpopularniji je onaj preko dijagrama toka (blok sheme), gdje je svaki korak u algoritmu predstavljen blokom koji ima određeni geometrijski oblik, a veze između koraka predstavljeni su sa strelicama.

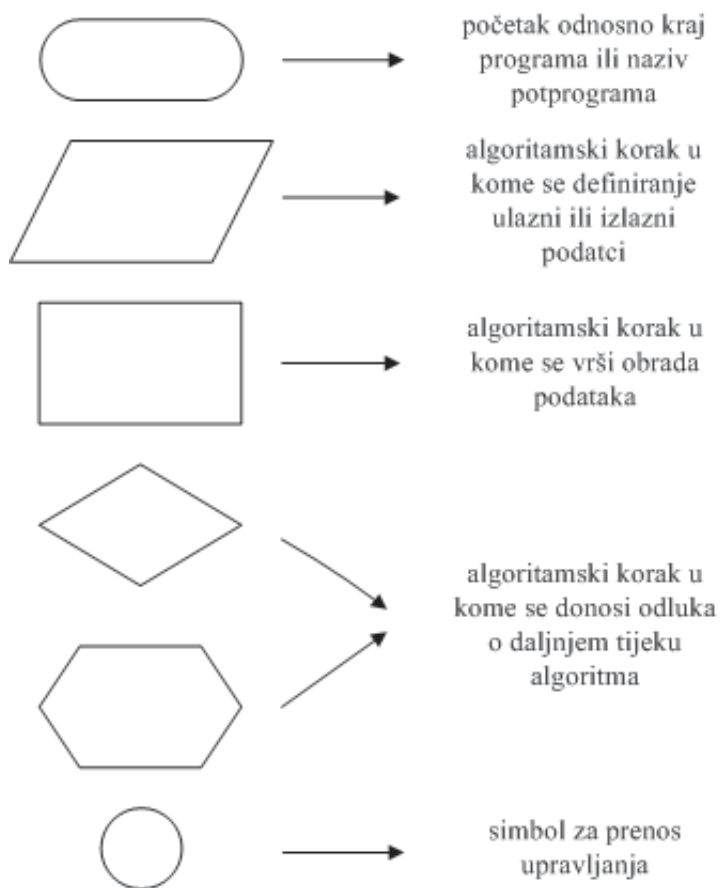
Drugi način predstavljanja algoritama je uz pomoć pseudo jezika (tekstualna notacija).

Radi implementacije algoritma na računalu, potrebno je opisati ga u nekom, u tu svrhu odabranom jeziku za programiranje.

4.2. Dijagrami toka

Pri predstavljanju algoritama uz pomoć dijagrama toka koriste se sljedeći grafički simboli (blokovi):

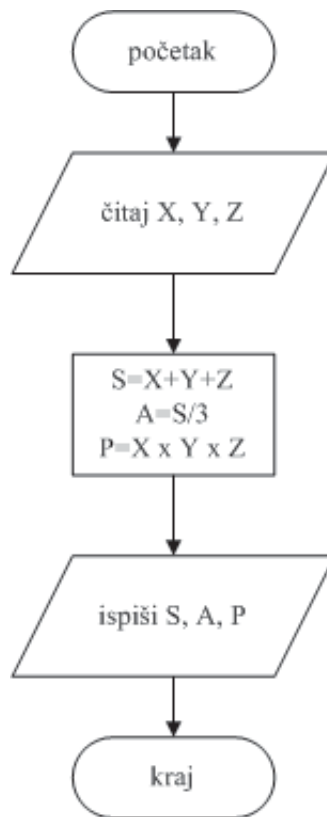
Postoje tri tipa algoritamske strukture: linijska (sekvenca), razgranata (selekcija) i ciklička (petlja). Kod linijskih algoritamskih struktura moguće je samo jedno izvršavanje nekog algoritamskog koraka, tj. nakon izvršavanja jednog algoritamskog koraka izvršavanje se može prenjeti samo na algoritamski korak koji još nije izvršen.



Slika 4.1.

Primjer 4.1. *Opišite dijagram toka za algoritam koji određuje sumu S , aritmetičku sredinu A i produkt P brojeva X, Y i Z .*

Rješenje.

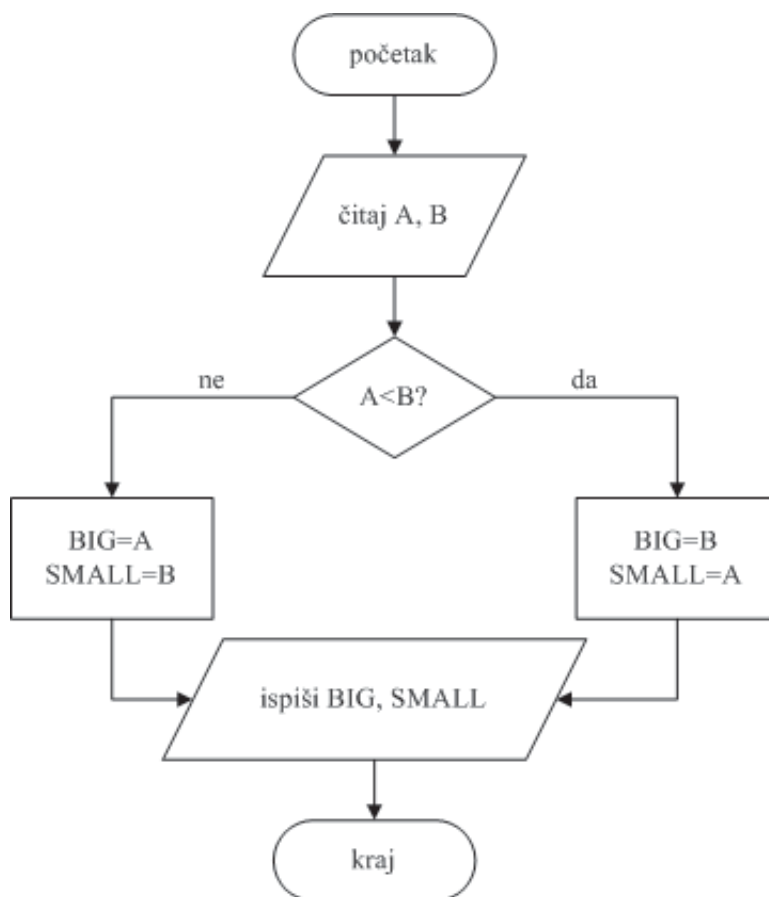


Slika 4.2.

Često je u algoritmima potrebno odlučiti koji od dva ili više algoritamskih koraka treba izvršiti s obzirom na postavljeni uvjet. Jednostavni oblik razgranate algoritamske strukture (selekcija) je izbor jednog od dva algoritamska koraka, ovisno o tome da li je postavljeni uvjet istinit, kad bi se izveo prvi algoritamski korak, ili neistinit, kad bi se izveo drugi. To je poznata IF-THEN-ELSE naredba u većini programskih jezika.

Primjer 4.2. *Opišite dijagram toka za algoritam koji dva zadana broja ispisuje u padajućem poretku.*

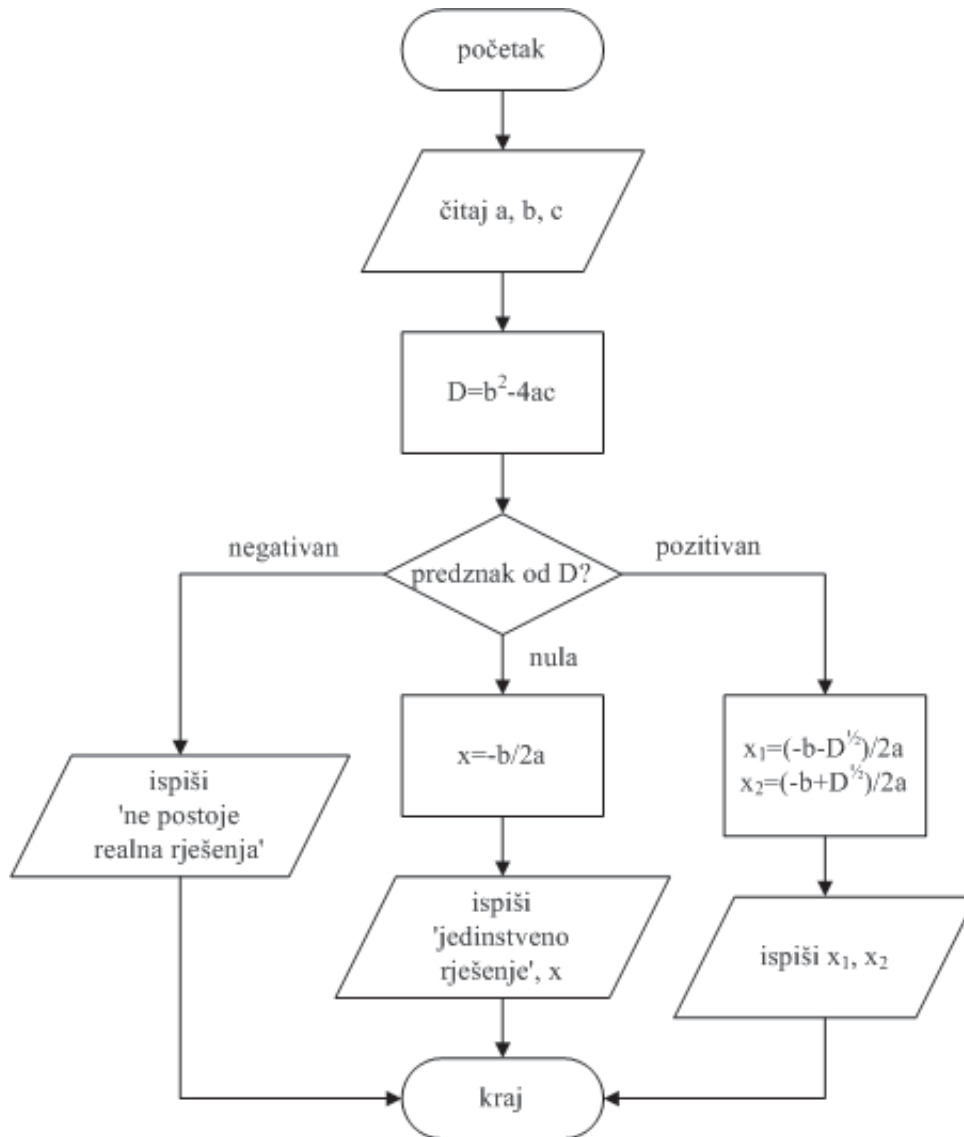
Rješenje.



Slika 4.3.

Primjer 4.3. *Opišite dijagram toka za algoritam koji određuje rješenja kvadratne jednadžbe $ax^2 + bx + c = 0$.*

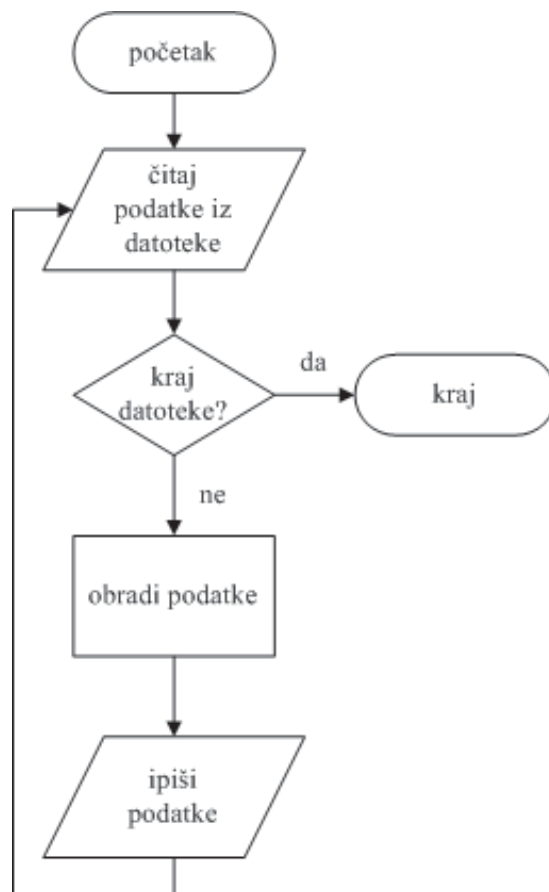
Rješenje.



Slika 4.4.

4.3. Petlje

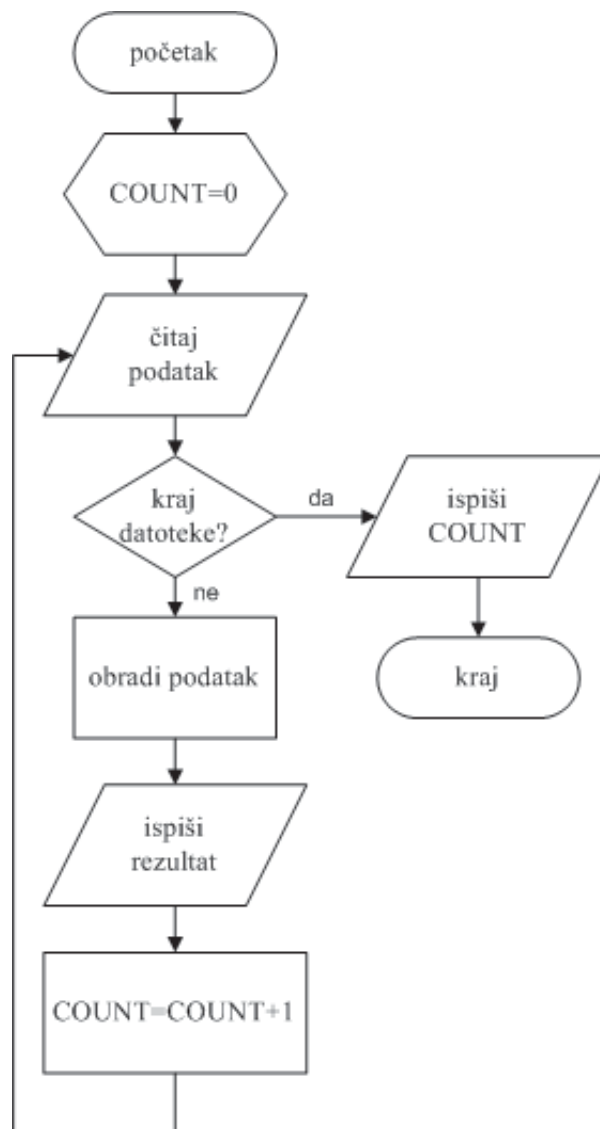
Dio algoritma koji zahtijeva višestruko izvršavanje jednog ili više algoritamskih koraka, pri čemu može doći i do promjene zadanog načina obrade podataka zovemo "petlja". To se ponavljanje vrši sve dok je neki postavljeni uvjet ispunjen (okončava kad uvjet više nije ispunjen - kad postane neistinit), ili sve dok je određeni uvjet neispunjen (okončava kada je uvjet ispunjen - postao je istinit).



Slika 4.5.

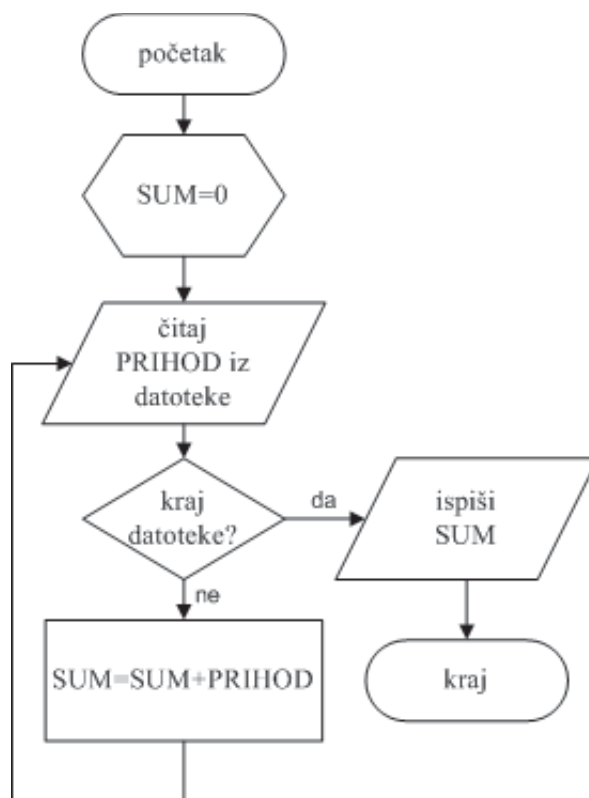
4.4. Brojači, DO petlja

Kada u nekoj petlji u zadanom algoritmu želimo izračunati broj podataka koji je obrađen, moramo postaviti jednu varijablu (brojač) čija je vrijednost početku jednaka nuli, a svaki put kada se neki podatak obradi njena se vrijednost povećava za 1.



Slika 4.6.

Ovakva se algoritamska struktura koristi i u problemima kada želimo izračunati sumu određenog broja numeričkih podataka. Ako recimo želimo izračunati ukupni prihod nekog poduzeća u tjedan dana dijagram toka je:

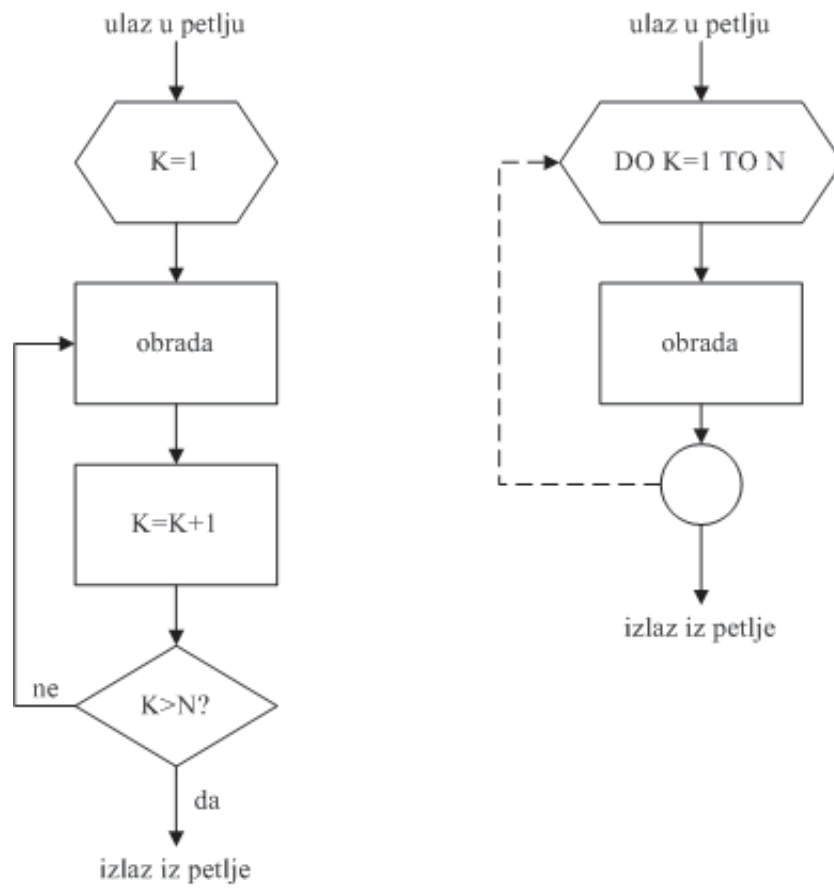


Slika 4.7.

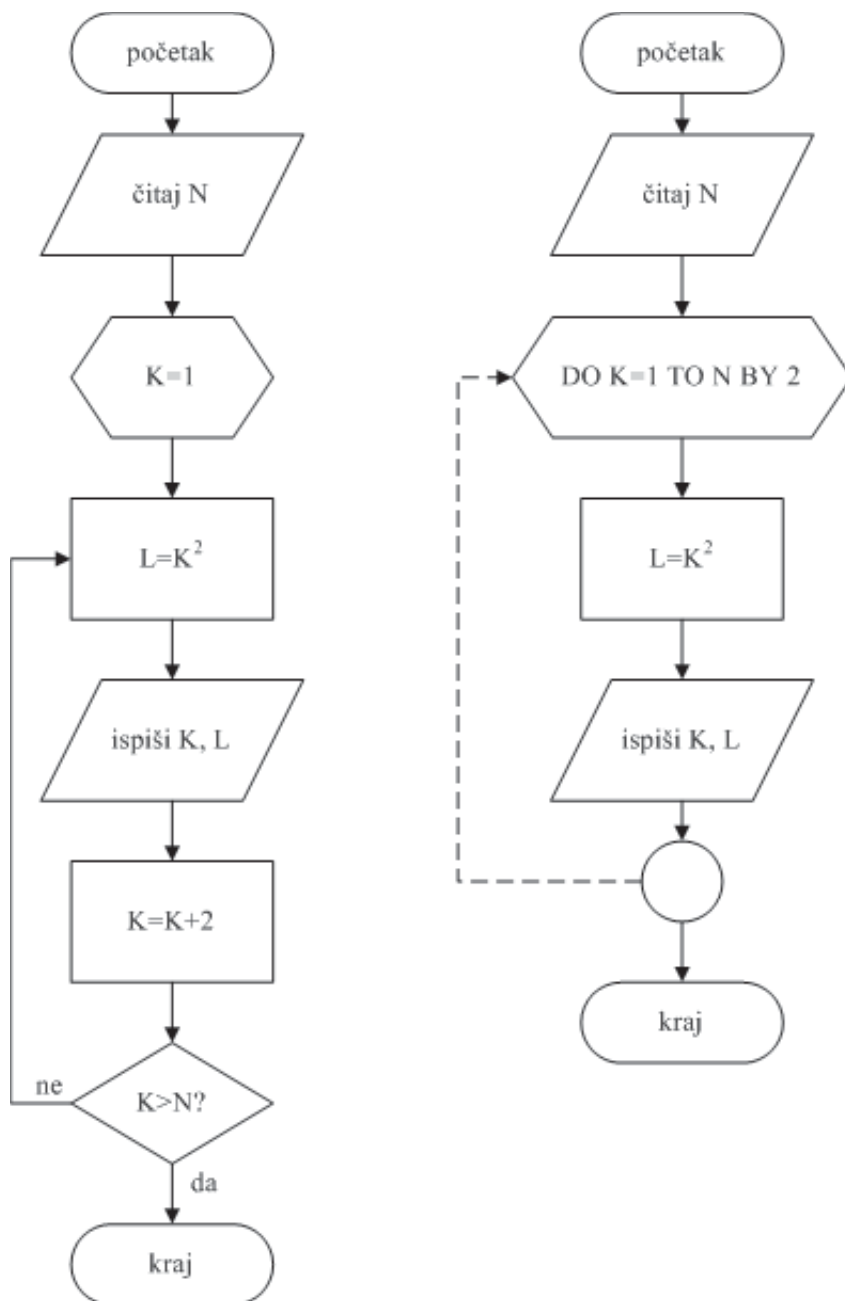
Ako neki algoritamski korak želimo izvršiti točno N puta možemo koristiti brojač kao u prethodnim primjerima ili tzv. DO petlju:

Primjer 4.4. *Odredite dijagram toka za algoritam koji nalazi sve pozitivne neparne cijele brojeve, zajedno s njihovim kvadratima, koji su manji ili jednaki od broja N , gdje je N cijeli broj veći ili jednak od 1.*

Rješenje.



Slika 4.8.



Slika 4.9.

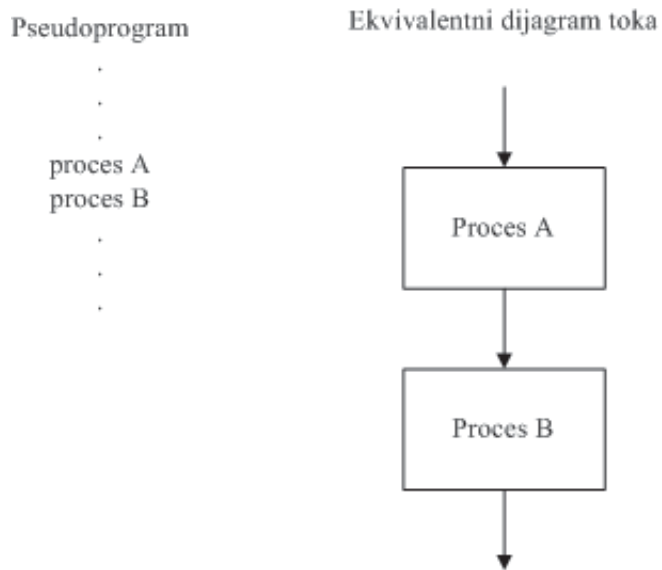
4.5. Pseudoprogrami

Za algoritam koji je sam po sebi dovoljno složen, predstavljanje uz pomoć dijagrama toka može biti teško za čitanje, modificiranje ili revidiranje. Zbog toga je razvijen tzv.

pseudoprogram uz pomoć kojeg možemo predstaviti algoritam.

Pseudoprogram je sastavljen od liste naredbi. Neke od tih naredbi su slične onima koje koristimo kod dijagrama toka, npr. čitaj, ispiši ili uvjetni izrazi.

Kod linijske algoritamske strukture naredbe u pseudoprogramu izvršavaju se redom počevši od vrha pa sve do dna.



Slika 4.10.

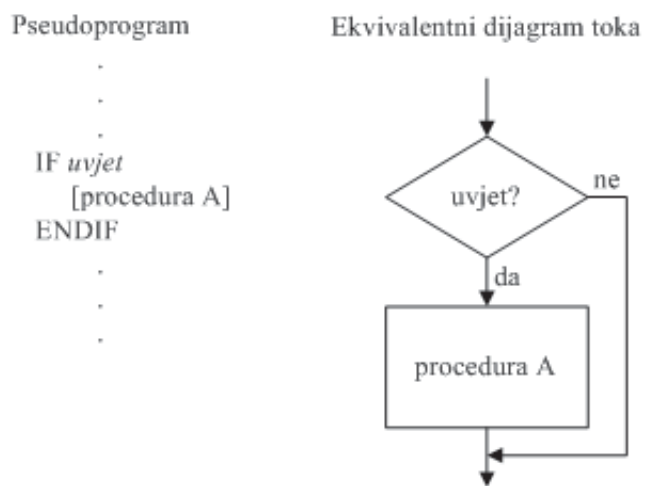
Kod selekcije se pojavljuje tzv. IF-struktura. Svaka takva struktura počinje s izrazom

IF *uvjet*

a završava s izrazom

ENDIF

Ona može biti jednostruka: IF-THEN struktura, dvostruka: IF-THEN-ELSE struktura i višestruka: ELSEIF struktura.



Slika 4.11. IF-THEN struktura

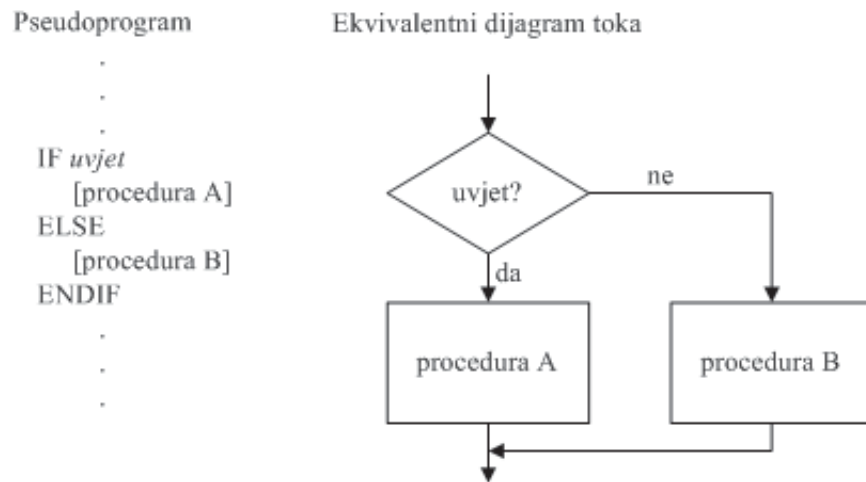
Primjer 4.5. *Napišite pseudoprogram za algoritam koji računa tjednu zaradu zaposlenika, ako je poznata njegova naknada za sat vremena rada, a prekovremeni se rad plaća 50% više.*

Rješenje.

```

čitaj IME, NAKNADA, ODRADJENO
ZARADA=ODRADJENO x NAKNADA
IF ODRADJENO>40
  ZARADA=ZARADA+(ODRADJENO-40) x 0.5 x NAKNADA
ENDIF
ispiši IME, ZARADA
END

```



Slika 4.12. IF-THEN-ELSE struktura

Primjer 4.6. *Napišite pseudoprogram za algoritam iz Primjera 4.2..*

Rješenje.

čitaj A,B

IF A<B

 BIG=B

 SMALL=A

ELSE

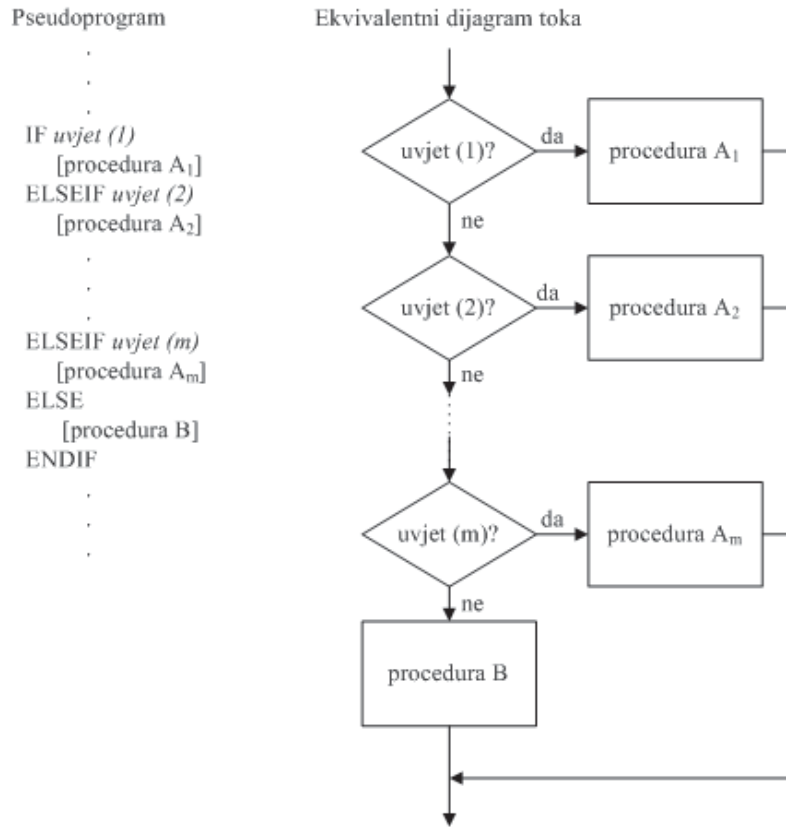
 BIG=A

 SMALL=B

ENDIF

ispiši BIG, SMALL

END



Slika 4.13.ELSEIF struktura

Primjer 4.7. Napišite pseudoprogram za algoritam iz Primjera 4.3..

Rješenje.

čitaj a, b, c

$D = b^2 - 4ac$

IF $D > 0$

$x_1 = (-b - \sqrt{D})/2a$

$x_2 = (-b + \sqrt{D})/2a$

ispiši x_1, x_2

ELSEIF $D = 0$

$x = -b/2a$

ispiši 'jedinstveno rješenje', x

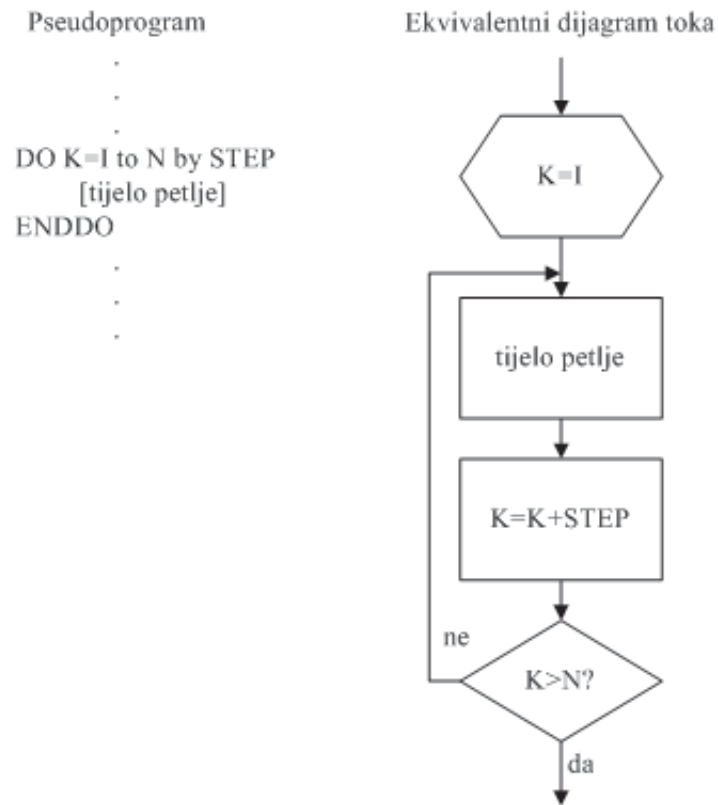
ELSE

ispiši 'nema realnih rješenja'

ENDIF

END

Kod pisanja petlji unutar pseudoprograma razlikujemo tri tipa strukture: DO, DOWHILE i DOUNTIL. Nareda ENDDO označava kraj petlje.



Slika 4.14.DO struktura

Primjer 4.8. *Napišite pseudoprogram za algoritam iz Primjera 4.4..*

Rješenje.

čitaj N

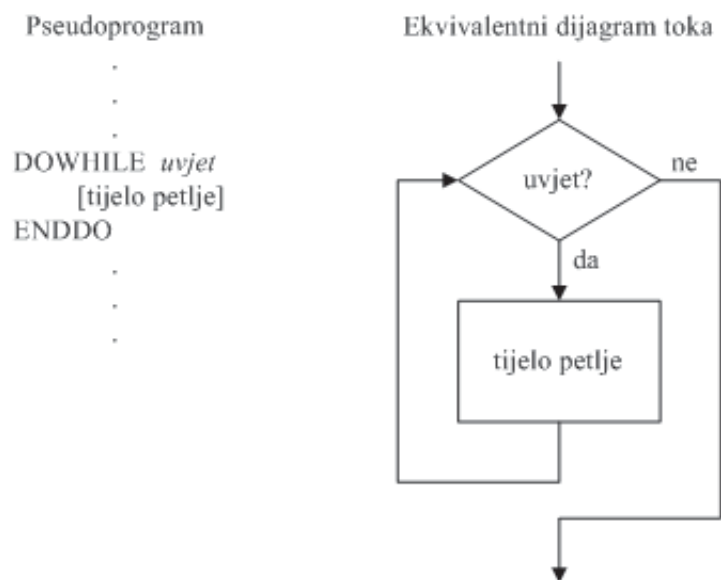
DO $K = 1$ to N by 2

$L = K^2$

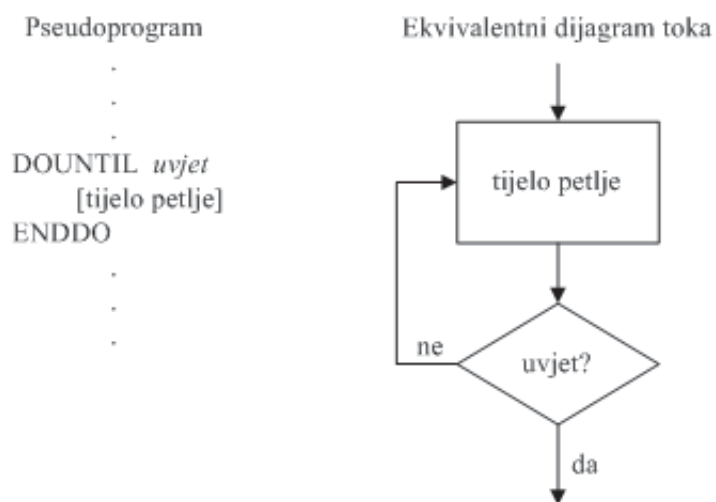
ispiši K, L

ENDDO

END



Slika 4.15.DOWHILE struktura



Slika 4.16.DOUNTIL struktura

Primjer 4.9. *Napišite pseudoprograme za algoritam iz Primjera 4.4., koristeći DOWHILE i DOUNTIL strukturu.*

Rješenje.

<pre> čitaj N K = 1 DOWHILE K ≤ N L = K² ispiši K, L K = K + 2 ENDDO END </pre>	<pre> čitaj N K = 1 DOUNTIL K > N L = K² ispiši K, L K = K + 2 ENDDO END </pre>
--	---

Primjer 4.10. *Pretpostavimo da neko poduzeće podatke o imenima i dobi svojih zaposlenika čuva u jednoj datoteci. Napišite pseudoprogram za algoritam koji treba ispisati imena zaposlenika koji su mlađi od 30 godina.*

Rješenje.

```

čitaj IME, DOB (prvi zapis iz datoteke)
DOUNTIL kraj datoteke
    IF DOB < 30
        ispiši IME
    ENDIF
    čitaj IME, DOB iz datoteke (sljedeći zapis iz datoteke)
ENDDO
END

```

Zadaci za vježbu

1. Opišite dijagram toka i napišite pseudoprogram za algoritam koji računa opseg i površinu trokuta sa stranicama duljine a, b, c .
2. Opišite dijagram toka i napišite pseudoprogram za algoritam koji za zadana tri broja a, b i c određuje da li mogu biti duljine stranica nekog trokuta.
3. Opišite dijagram toka i napišite pseudoprogram za algoritam koji iz datoteke u kojoj su podaci s razredbenog postupka ispisuje imena onih budućih studenata koji su osvojili 100 bodova.
4. Opišite dijagram toka i napišite pseudoprogram za algoritam koji iz datoteke s podacima studenata određuje srednju vrijednost njihovih težina.
5. Opišite dijagram toka i napišite pseudoprogram za algoritam koji iz datoteke s podacima studenata nalazi najvišeg studenta.
6. Opišite dijagram toka za algoritam koji za pozitivni cijeli broj N računa:

(a) $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{N}$

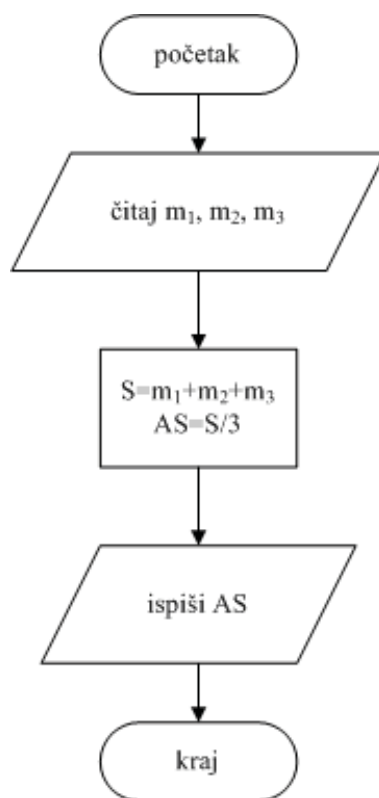
(b) $1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots \pm \frac{1}{N}$

7. Neka je $y = 2x^3 - 3x^2 - 27x + 28$. Napišite pseudoprogram za algoritam koji računa y za x između -5 i 5 s korakom 0.25 .
8. Koristeći DOWHILE strukturu, napišite pseudoprogram za algoritam koji pronalazi sve paraove cjelih pozitivnih brojeva m i n , t.d. je $m^2 + 2n^2 < 100$.
9. Opišite dijagram toka za algoritam koji za zadan pozitivni cijeli broj n računa $n!$.

4.6. Primjena u struci

Primjer 4.11. *Izračunati aritmetičku sredinu za ponovljena mjerenja mase uzorka.*

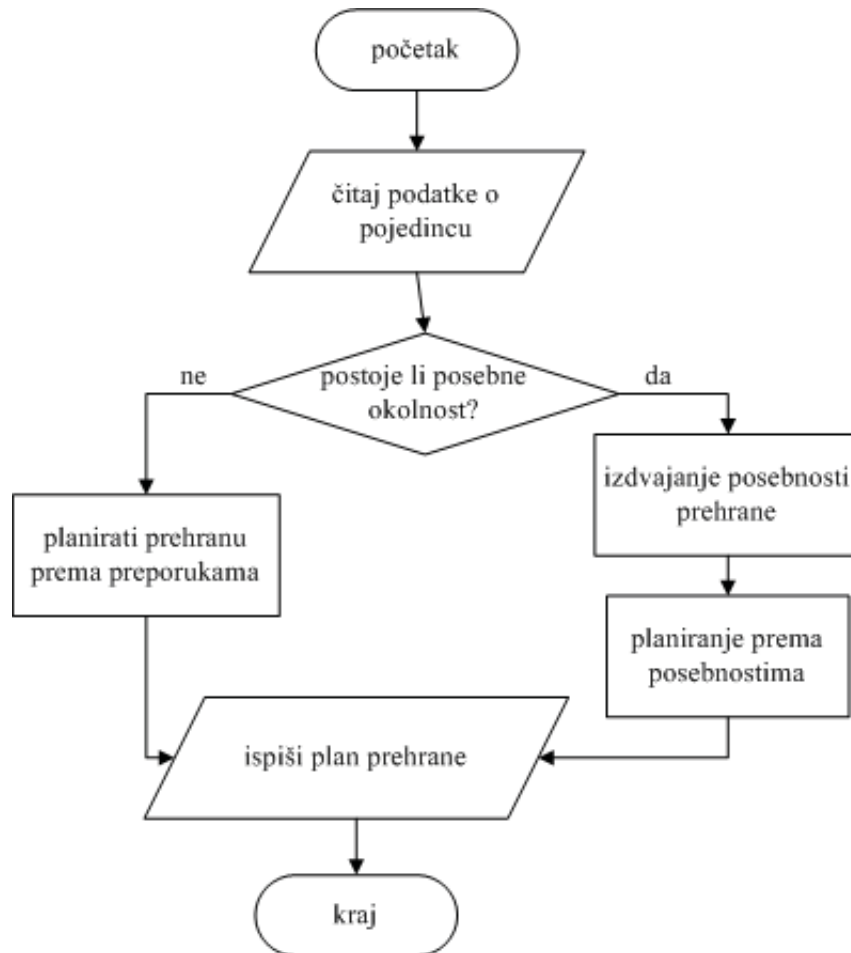
Rješenje.



Slika 4.17.

Primjer 4.12. *Planiranje prehrane za pojedince (prema DRI). Podaci o osobi su: spol, dob, tjelesna visina i tjelesna masa. Posebne okolnosti podrazumijevaju i posebne nutrijente koji će se tada promatrati. Npr. pušač (vitamin C), atletičar (Fe), vegetarijanac (Fe, Zn, folati), bolesnik (posebna prehrana) i sl.*

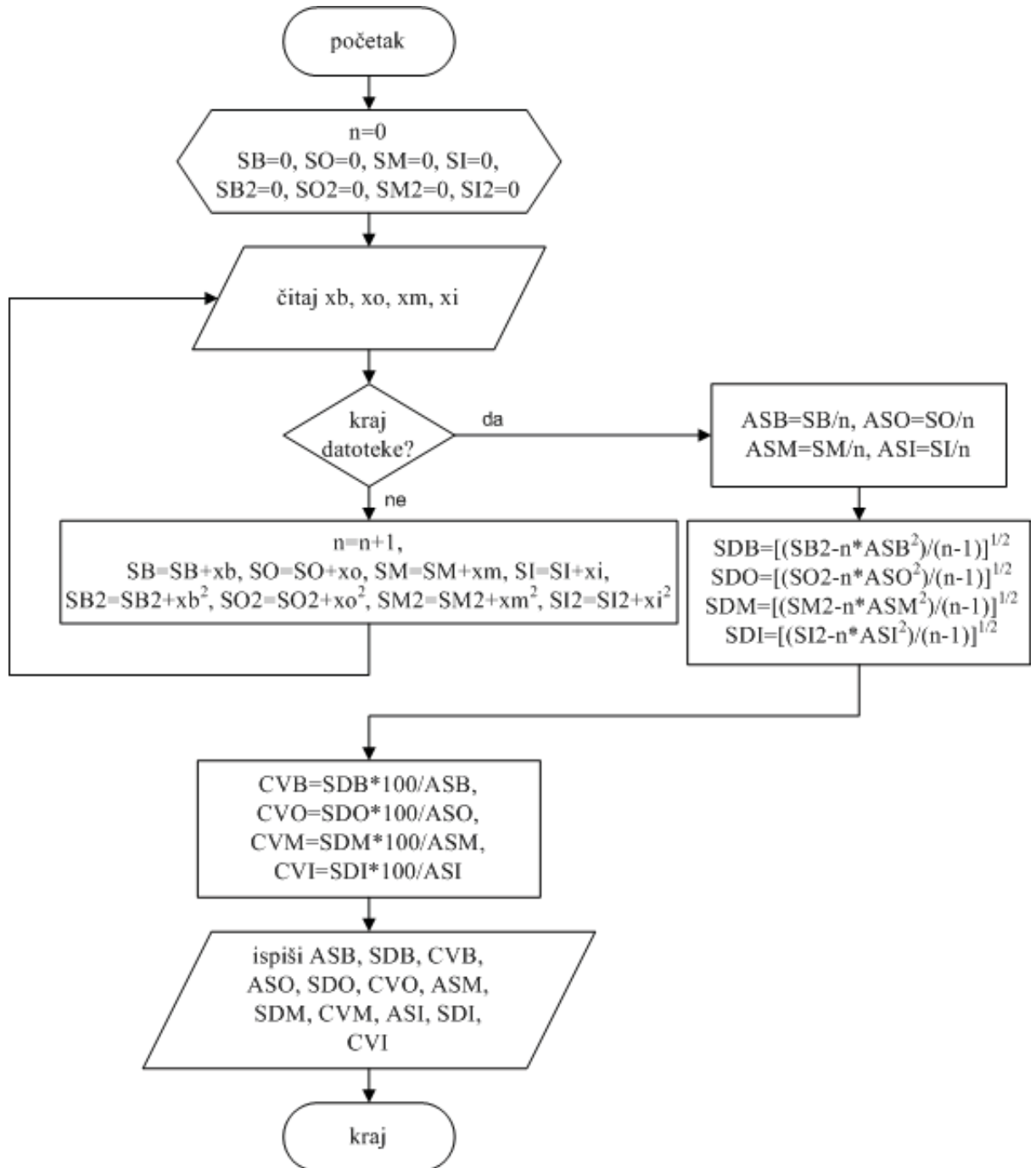
Rješenje.



Slika 4.18.

Primjer 4.13. n je ocjenivača ocjenjivalo 4 senzorske karakteristike: boju, okus, miris i izgled marmelade od mandarine. Zadatak je izračunati prosječne vrijednosti senzorske analize: SD -standardna devijacija i $CV = \frac{SD}{\bar{x}} \cdot 100\%$ -koeficijent varijacije, gdje je \bar{x} aritmetička sredina. Podaci su zadani u datoteci.

Rješenje.



Slika 4.19.