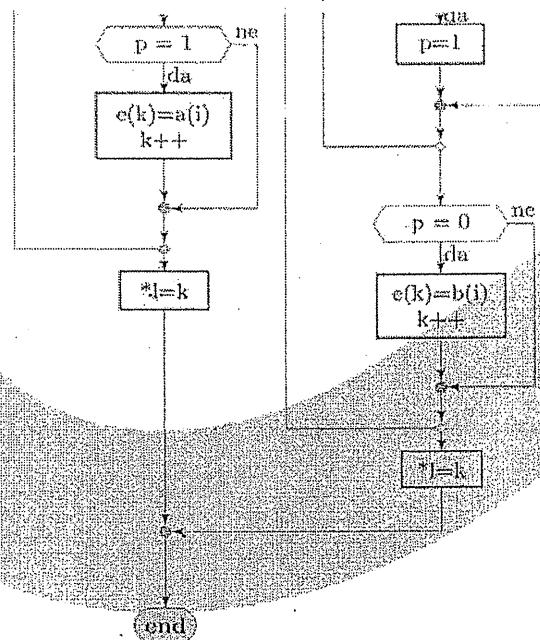
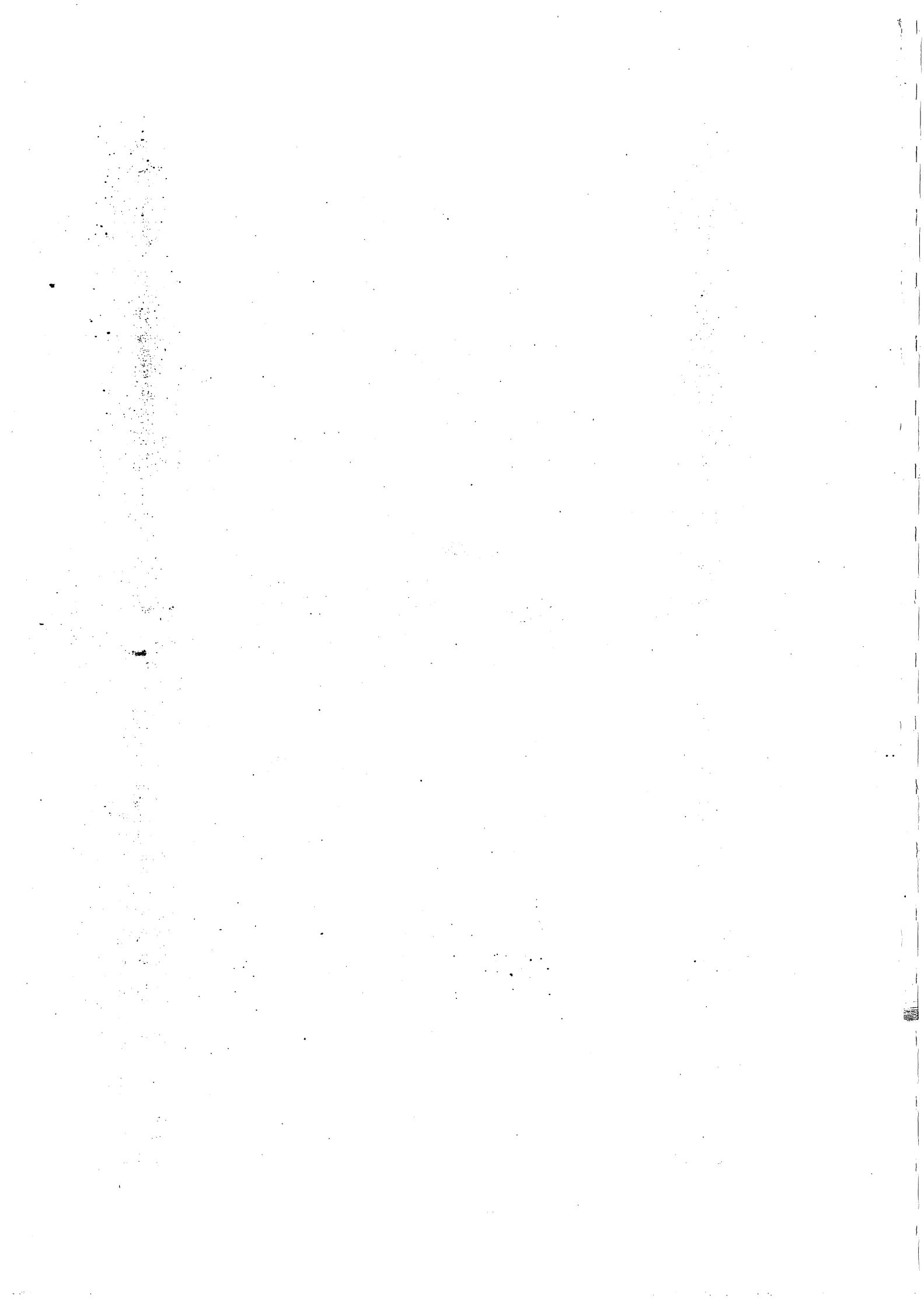


Algoritmi i programiranje zbirka rešenih zadataka na programskom jeziku C

Grupa autora



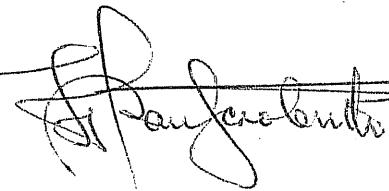




Univerzitet u Nišu
Elektronski fakultet

Leonid Stoimenov, Dragan Janković, Dejan Rančić,
Vladimir Ćirić, Natalija Stojanović, Oliver Vojinović,
Nikola Davidović, Nataša Veljković

Algoritmi i programiranje
zbirka rešenih zadataka na programskom jeziku C



Niš, 2012.

ALGORITMI I PROGRAMIRANJE
ZBIRKA REŠENIH ZADATAKA NA PROGRAMSKOM JEZIKU C

Autori Prof. dr Leonid Stoimenov, prof. dr Dragan Janković, prof. dr Dejan Rančić,
Doc. dr Vladimir Ćirić, doc. dr Natalija Stojanović, mr Oliver Vojinović,
Dipl. inž. Nikola Davidović, dipl. inž. Nataša Veljković.
Izdavač Elektronski fakultet u Nišu
P.fah 73, 18000 Niš
<http://www.elfak.ni.ac.rs/>
Recenzenti Prof. dr Emin Milovanović, Elektronski fakultet u Nišu
Doc. dr Suzana Stojković, Elektronski fakultet u Nišu
Teh. obrada Doc. dr Vladimir Ćirić

Glavni i odgovorni urednik: Prof. dr Zoran Perić

Odlukom Nastavno-naučnog veća Elektronskog fakulteta u Nišu, br. 07/05-008/12-003 od 24.05.2012. godine, rukopis je odobren za štampu kao pomoći udžbenik na Elektronskom fakultetu u Nišu.

ISBN 978-86-6125-069-9

CIP - Каталогизација у публикацији
Народна библиотека Србије, Београд

004.42(075.8)(076)
004.432.2C(075.8)(076)

ALGORITMI i programiranje : zbirka rešenih zadatka na programskom jeziku,C / Leonid Stoimenov ... [et al.]. - Niš : Elektronski fakultet, 2012 (Niš : Unigraf). - 153 str. : graf. prikazi ; 27 cm

Na vrhu nasl. str.: Univerzitet u Nišu. -
Tiraž 300. - Bibliografija: str. 153.

ISBN 978-86-6125-069-9
1. Стоименов, Леонид, 1967- [автор]
а) Програмирање - Вежбе б) Програмски језик "С" - Задаци
COBISS.SR-ID 191375372

Простављавање или умноžавање ове књиге nije dozvoljeno bez pismene dozvole izdavača.

Tiraž: 300 primeraka
Štampa: Unigraf, Niš

Predgovor

U ovoj zbirci zadataka metodički je prezentovana kolkcija rešenih problema vezanih za kreiranje strukturalnih dijagrama toka algoritma i implementaciju algoritama na programskom jeziku C. Kao takva, prvenstveno je namenjena studentima I godine Elektronskog fakulteta kao pomoćni udžbenik za predmet Algoritmi i programiranje, ali se može koristiti i kao priručnik za ovlađavanje osnovnim algoritamskim operacijama nad skalarnim podacima i linearnim indeksiranim strukturama podataka.

Zbirka se sastoji iz tri glave.

U prvoj glavi izloženi su osnovni koncepti upravljanja izvršenjem programa i osnovne upravljačke strukture. Obrađena je osnovna struktura programa u programskom jeziku C i kontrola toka izvršenja programa. U cilju ilustracije osnovnih struktura prikazani su tipični matematički problemi iz geometrije i dela numeričke matematike vezani za iterativne postupke.

Druga glava posvećena je linearnim indeksiranim strukturama podataka. U ovoj glavi obrađena su dva podtipa linearnih indeksiranih struktura: jednodimenzionalna i dvodimenzionalna polja, sa elementima numeričkih i znakovnog tipa.

U trećoj glavi predstavljeni su koncepti kreiranja korisničkih funkcija u programskom jeziku C. Obradene su funkcije koje za parametre imaju skalarne podatke i funkcije koje vrše obradu nad linearnim indeksiranim strukturama. Na kraju ove glave dat je skup problema kod kojih je u cilju čitanja i/ili upisa potrebno pristupiti fajlu. Obradeni su tekstualni fajlovi sa sekvensijalnim pristupom.

Poglavlja su koncipirana tako da su na početku poglavlja jednostavnii primeri koji ilustruju osnovne koncepte, a za njima slede zadaci koji kombinuju osnovne principe u cilju rešavanja složenijih problema. Svaki zadatak se sastoji od tekstualnog opisa problema, dijagrama toka algoritma, koda na programskom jeziku C, i primera ulaznih podataka i izlaza koji se dobija za navedene ulazne podatke. Svi izlazi su dobijeni izvršavanjem odgovarajućeg C koda na računaru za ulazne podatke date u primeru. Svi dijagrami toka algoritma generisani su automatski iz C koda. Za konverziju koda u algoritam korišćen je kompjuter autoru doc. dr Vladimira Čirića, koji vrši leksičku i sintaksnu analizu C programa i generiše dijagrami toka korišćenjem L^AT_EX primitiva.

Jedinstvena rešenja u programiranju su retkost, tako da velika većina zadataka iz ove zbirke ima i druga rešenja, različita od predloženih. Primarni cilj zbirke nije da da optimalna rešenja u bilo kom smislu, već da ponudi razumljiva rešenja. Sa

druge strane, i pored proveravanja, moguće je da i dalje postoje greske i nedostaci u ponuđenim rešenjima. Autori će biti zahvalni ukoliko im se ukaže na eventualne greske.

Autori

Zbirka zadataka

2π

Sadržaj

1 Osnovne upravljačke strukture	1
1.1 Struktura C programa i osnovne aritmetičke operacije	1
1.2 Kontrola toka izvršenja programa	8
1.2.1 Grananja	8
1.2.2 Petlje	11
1.3 Iterativni postupci	34
1.4 Geometrijski problemi	39
2 Linearne indeksirane strukture podataka	49
2.1 Nizovi	49
2.1.1 Brojevni nizovi	49
2.1.2 Znakovni nizovi - stringovi	85
2.2 Matrice	95
3 Funkcije i rad sa fajlovima	121
3.1 Funkcije sa skalarnim parametrima	121
3.2 Prenos nizova preko parametara funkcije	124
3.3 Matrice u funkciji	133
3.4 Fajlovi	142
3.5 Fajlovi u funkciji	145

Zbirka zadataka

1

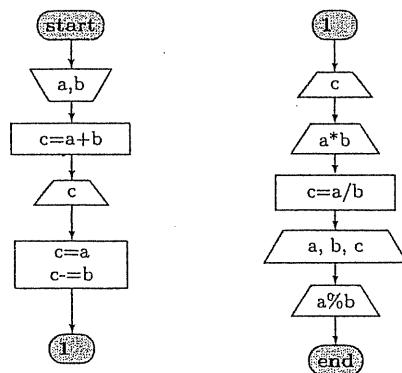
Osnovne upravljačke strukture

1.1 Struktura C programâ i osnovne aritmetičke operacije

Zadatak 1.1

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati program kojim se nad celobrojnim promenljivama a i b , čije vrednosti zadaje korisnik, izvršavaju osnovne aritmetičke operacije ($+, -, *, /, \%$), i prikazuju odgovarajući rezultati.

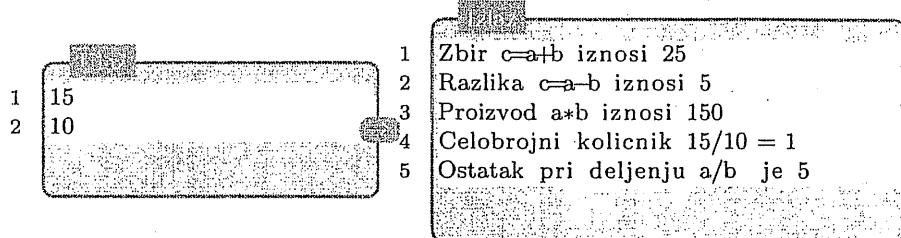
Rešenje



```

1 #include <stdio.h>
2 main()
3 {
4     int a,b,c;
5
6     printf ("uneti_prvi_broj\n");
7     scanf ("%d",&a);
8     printf ("uneti_drugi_broj\n");
9     scanf ("%d",&b);
10    c=a+b;
11    printf ("Zbir_c-a+b_iznosi.%d\n",c);
12    c=a;
13    c=b;
14    printf ("Razlika_c-a-b_iznosi.%d\n",c);
15    printf ("Proizvod_a*b_iznosi.%d\n",a*b);
16    c=a/b;
17    printf ("Celobrojni_kolicnik.%d/%d=%d\n", a,b,c);
18    printf("Ostatak_pri_deljenju_a/b_je.%d\n", a%b);
19 }

```



Zadatak 1.2

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati program kojim se nad realnim promenljivama a i b , čije vrednosti zadaje korisnik, izvršavaju osnovne aritmetičke operacije (+,-,*,/), i prikazuju odgovarajući rezultati.

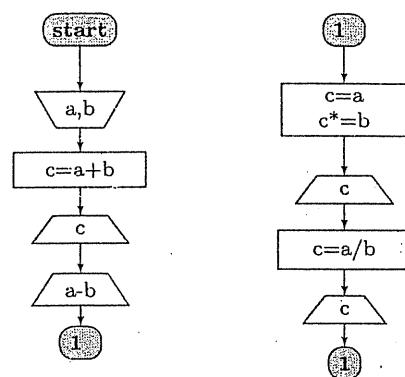
Rešenje

```

1 #include <stdio.h>
2 void main()
3 {
4     float a,b,c;
5     scanf ("%f",&a);
6     scanf ("%f",&b);
7     c=a+b;
8     printf ("Zbir_a+b_iznosi.%f\n",c);
9     printf ("Razlika_a-b_iznosi.%f\n",a-b);
10    c=a; c=b;

```

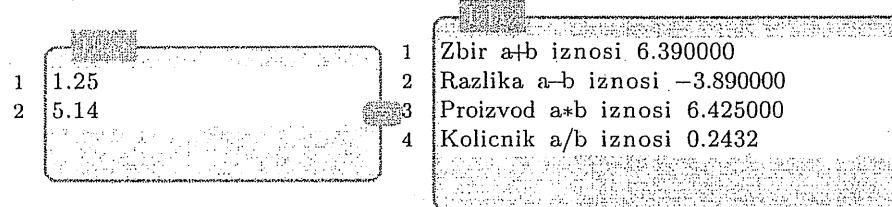
Zbirka zadataka



```

11     printf ("Proizvod a*b iznosi %.f\n",c);
12     c=a/b;
13     printf ("Kolicnik a/b iznosi %.4f\n",c);
14 }

```



Zadatak 1.3

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program za zamenu mesta celobrojnih vrednosti koje se nalaze u promenljivama a i b , tako da promenljiva b sadrži vrednost unetu u a , i obrnuto. Vrednosti promenljivih a i b zadaje korisnik. Prikazati vrednosti promenljivih nakon zamene mesta.

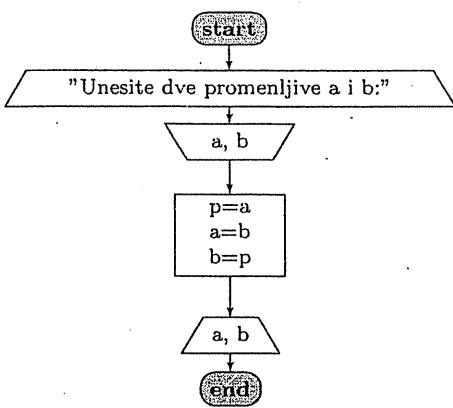
Rešenje

```

1 #include <stdio.h>
2 void main()
3 {
4     int a, b,p;
5     printf("Unesite dve promenljive a i b:\n");
6     scanf("%d%d",&a,&b);
7     p=a;
8     a=b;
9     b=p;

```

Zbirka zadataka



```

10     printf("Nakon zamene mesta promenljive imaju vrednosti a=%d i ..
11     b=%d\n", a,b);
  
```

1 3 5

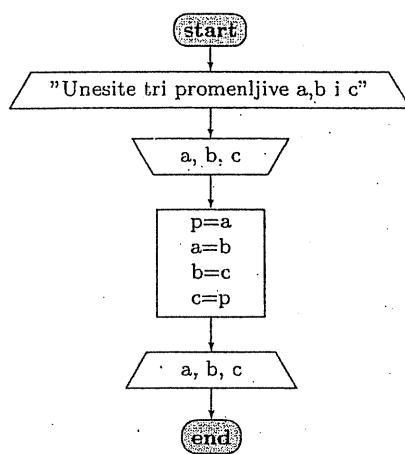
1 Unesite dve promenljive a i b:
2 Nakon zamene mesta promenljive imaju
vrednosti a=5 i b=3

Zadatak 1.4

Nacrtati strukturalni dijagram toka algoritma i na programskom jeziku C napisati strukturalni program za unos tri celobrojne vrednosti u promenljive a , b i c . Nakon unosa izvršiti rotaciju vrednosti promenljivih u levo, tako da se vrednost promenljive c nađe u promenljivoj b , vrednost promenljive b u promenljivoj a , i vrednost promenljive a u promenljivoj c . Prikazati vrednosti promenljivih nakon rotacije.

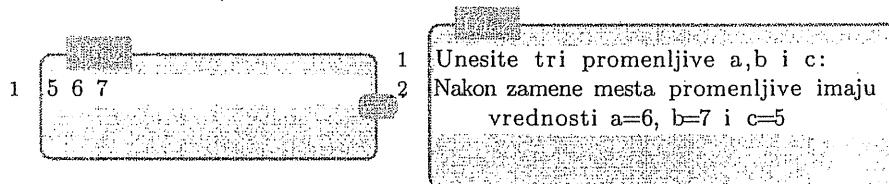
Rešenje

Zbirka zadataka



```

1 #include <stdio.h>
2
3 void main()
4 {
5     int a,b,c,p;
6     printf("Unesite tri promenljive a,b i c:\n");
7     scanf("%d%d%d",&a,&b,&c);
8     p=a;
9     a=b;
10    b=c;
11    c=p;
12    printf("Nakon zamene mesta promenljive imaju vrednosti a=%d, b=%d,
13 i c=%d\n", a,b,c);
  
```

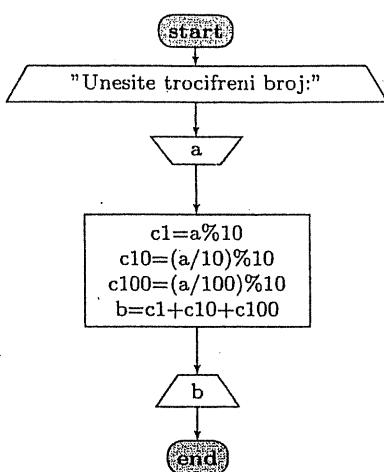


Zadatak 1.5

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje zbir cifara zadatog trocifrenog broja.

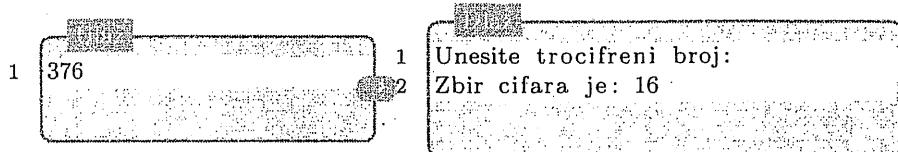
Rešenje

Zbirka zadataka



```

1 #include <stdio.h>
2 void main()
3 {
4     int a, b, c1, c10, c100;
5     printf("Unesite_trocifreni_broj:\n");
6     scanf("%d", &a);
7     c1 = a % 10;
8     c10 = (a / 10) % 10;
9     c100 = (a / 100) % 10;
10    b = c1 + c10 + c100;
11    printf("Zbir_cifara_trocifrenog_broja_je: %d\n", b);
12 }
  
```

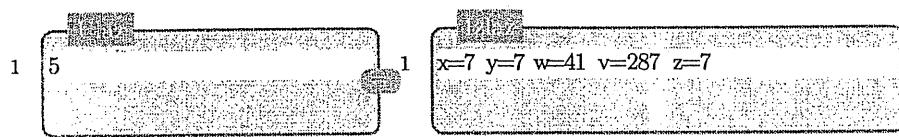
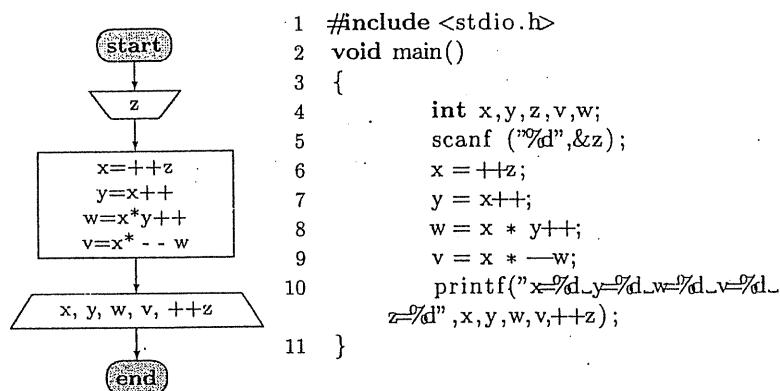


Zadatak 1.6

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji vrednost promenljive z , zadatu od strane korisnika, jednim iskazom jezika C inkrementira i dodeljuje promenljivoj x . Jednim iskazom vrednost promenljive x dodeliti promenljivoj y i uvećati za 1. Prikazati vrednosti promenljivih $x, y, w = x * y ++, v = x * --w + + z$.

Rešenje

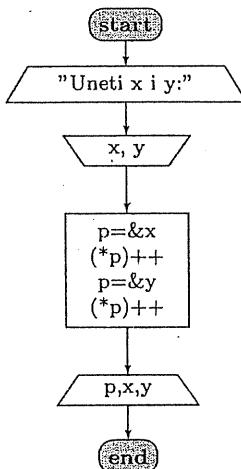
Zbirka zadataka



Zadatak 1.7

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji vrednosti promenljivih x i y uvećava za 1 korišćenjem jednog pokazivača p . Vrednosti x i y zadaje korisnik. Prikazati vrednosti promenljivih i vrednost pokazivača nakon modifikacije.

Rešenje

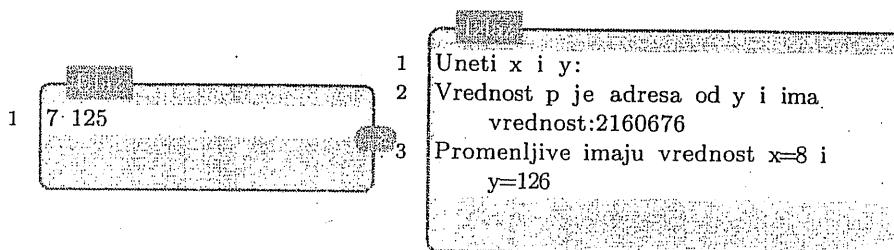


Zbirka zadataka

```

1 #include <stdio.h>
2 void main()
3 {
4     int x, y, *p;
5     printf("Uneti x i y:\n");
6     scanf("%d%d", &x, &y);
7     p = &x;
8     (*p)++;
9     p = &y;
10    (*p)++;
11    printf("Vrednost p je adresa od y i ima vrednost:%d\n", p);
12    printf("Promenljive imaju vrednost x=%d i y=%d\n", x, y);
13 }

```



Napomena: Vrednost pokazivača razlikuje se na različitim računarima i pri različitim pozivima programa, jer predstavlja adresu memorijske lokacije u kojoj je smještena promenljiva y na koju ukazuje pokazivač p .

1.2 Kontrola toka izvršenja programa

1.2.1 Grananja

Zadatak 1.8

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji za zadati broj x , prikazuje vrednost broja x i njegovu apsolutnu vrednost $|x|$.

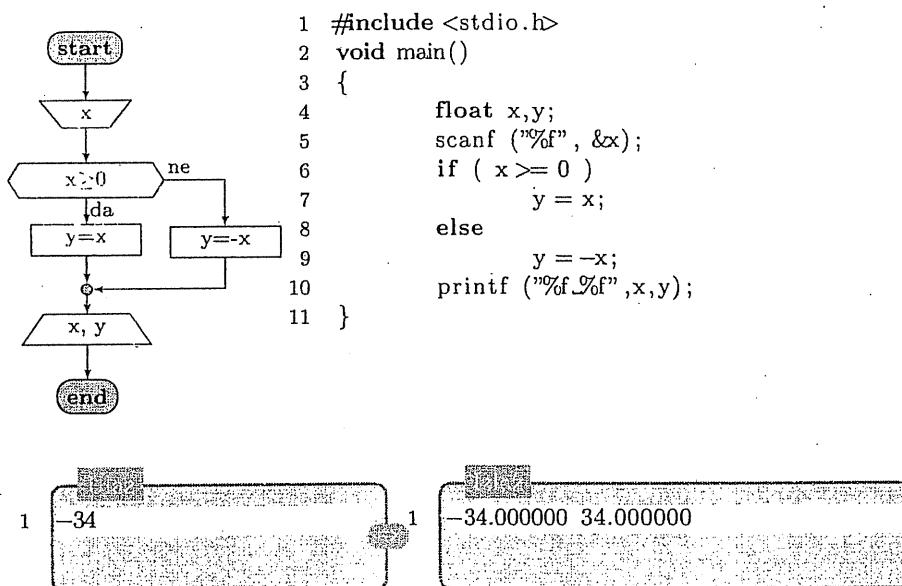
Rešenje

Apsolutna vrednost broja matematički se može izraziti kao

$$y = |x| = \begin{cases} x, & x \geq 0 \\ -x, & x < 0 \end{cases}$$

Direktnom interpretacijom matematičke formule dobijamo algoritam sa jednim uslovom u kome se ispituje da li je broj veći, ili manji od nule, što predstavlja rešenje zadatka.

Zbirka zadataka

**Zadatak 1.9**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji na osnovu karakterističnih podataka daje preporuke za životno opredeljenje. Program odlučuje, na osnovu unete visine i težine (celi broevi, u centimetrima, odnosno kilogramima) koja od tri buduće profesije najbolje odgovara korisniku. Idealnim predispozicijama za košarkaše se smatra visina veća od 184cm, idealnim predispozicijama za džokeje visina manja od 168cm i težina manja od 65kg, pri čemu težina igra značajniju ulogu. U školu za stjuarte u avionu se primaju samo kandidati visine između 174 i 189cm i maksimalne težine 84kg. Ukoliko nema idealnog poklapanja predispozicija, najbolje poklapanje se određuje na osnovu manje relativne razlike značajnije karakteristike. U slučaju idealnog poklapanja, program treba da pruži preporuku za profesiju, inače program daje preporuku i napomenu da je to "najbolji izbor".

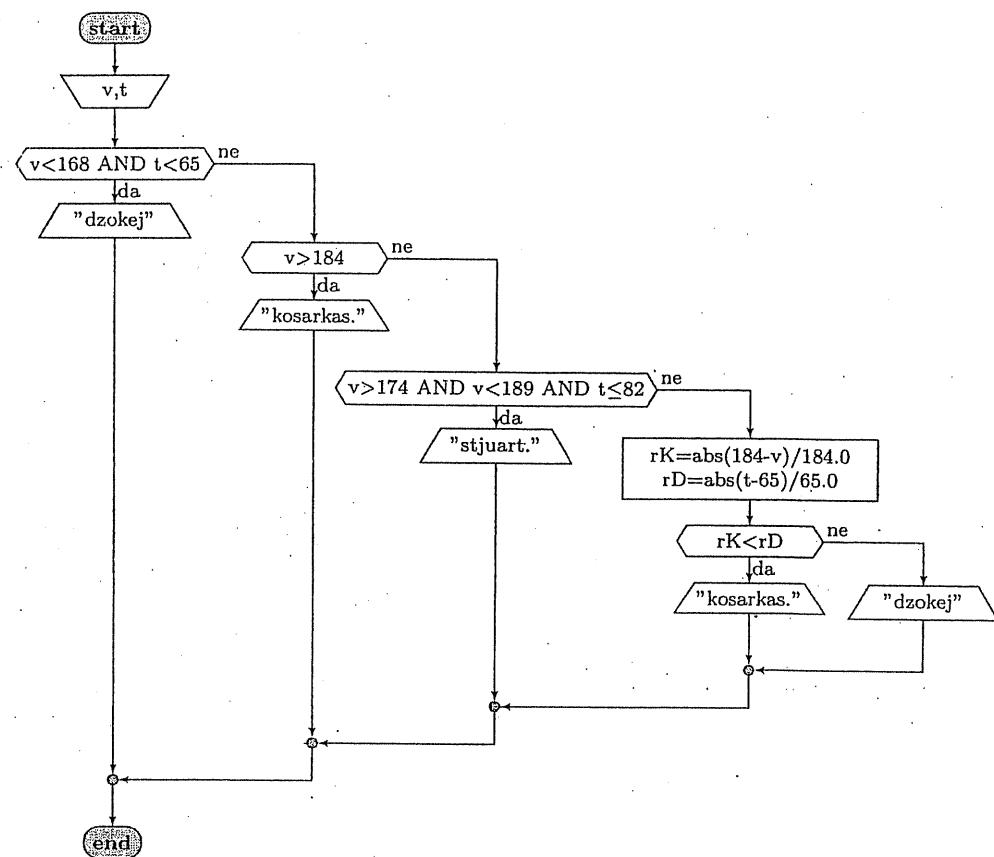
Rešenje

```

1 #include <stdio.h>
2 #include <math.h>
3 main()
4 {
5     int v,t;
6     float rK, rD;
7     printf("Unesite_visinu_u_centimetrima:\n");
8     scanf("%d",&v);
9     printf("Unesite_tezinu_u_kilogramima:\n");
10    scanf("%d",&t);
11    if ( v<168 && t<65)

```

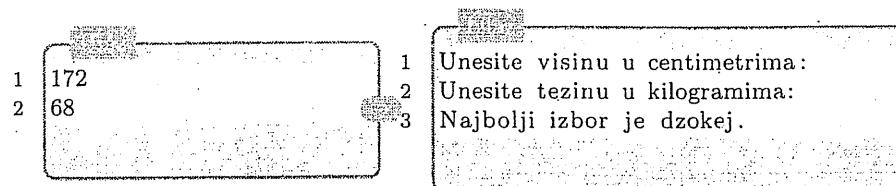
Zbirka zadataka



```

11     printf("Preporucujem_da_postanete_dzokej.\n");
12 else if (v>184)
13     printf("Preporucujem_da_postanete_kosarkas.\n");
14 else if (v>174 && v<189 && t<=82)
15     printf("Preporucujem_da_postanete_stjuart.\n");
16 else
17 {
18     rK=abs(184-v)/184.0;
19     rD=abs(t-65)/65.0;
20     if (rK<rD)
21         printf("Najbolji_izbor_je_kosarkas.\n");
22     else
23         printf("Najbolji_izbor_je_dzokej.\n");
24 }
25 }
  
```

Zbirka zadataka



1.2.2 Petlje

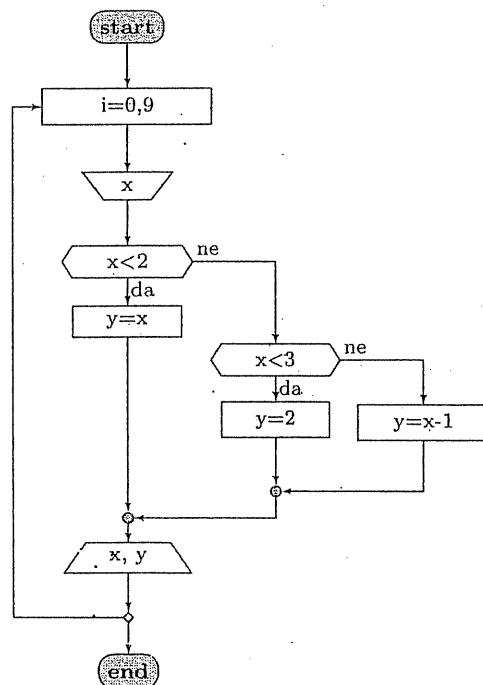
Zadatak 1.10

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji za 10 različitih vrednosti argumenta x određuje i prikazuje vrednost funkcije

$$y = \begin{cases} x, & x < 2 \\ 2, & 2 \leq x < 3 \\ x - 1 & x \geq 3 \end{cases}$$

Prikazati vrednost argumenta x i vrednost funkcije $y(x)$.

Rešenje

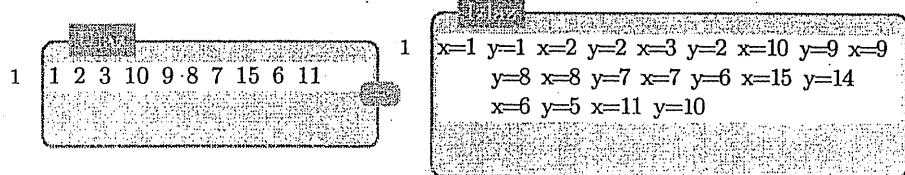


1 #include <stdio.h>

Zbirka zadataka

```

2 void main()
3 {
4     int i,x,y;
5     for( i = 0; i < 10; i++)
6     {
7         scanf ("%d",&x);
8         if ( x < 2 )
9             y = x;
10        else if ( x < 3 )
11            y = 2;
12        else
13            y = x-1;
14        printf ("%d,%d",x,y);
15    }
16 }
```

**Zadatak 1.11**

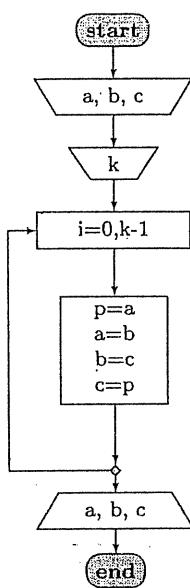
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji vrednosti tri promenljive a , b i c rotira za k mesta u levo. Vrednosti promenljivih i vrednost parametra k zadaje korisnik. Prikazati vrednosti promenljivih nakon rotacije.

Rešenje

```

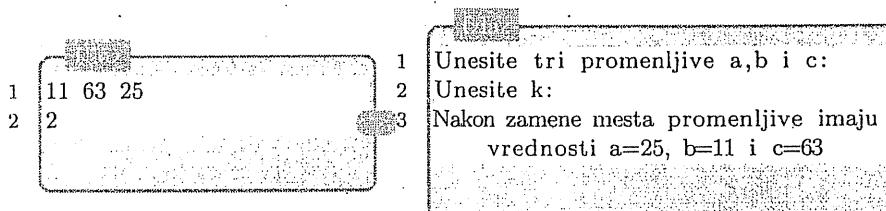
1 #include <stdio.h>
2 void main()
3 {
4     int a,b,c,k,p,i;
5     printf("Unesite tri promenljive a,b,i,c:\n");
6     scanf("%d%d%d",&a,&b,&c);
7     printf("Unesite k:\n");
8     scanf("%d",&k);
9     for (i=0;i<k; i++)
10    {
11        p=a;
12        a=b;
13        b=c;
14        c=p;
15    }
```

Zbirka zadataka



```

16     printf("Nakon zamene mesta promenljive imaju vrednosti a=%d, b=%d,
17         i=%d\n", a,b,c);
    }
  
```



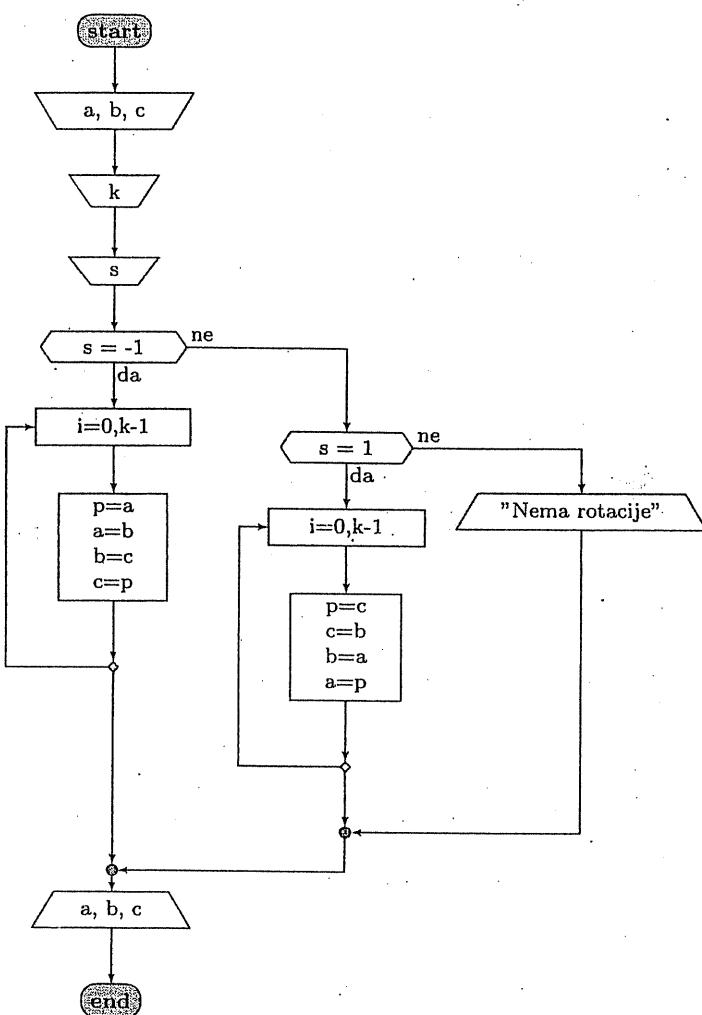
Zadatak 1.12

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji vrši rotaciju vrednosti tri promenljive za k mesta u levo ili desno u zavisnosti od unetog smera rotacije s . Ukoliko se za smer rotacije unese vrednost -1 , rotacija se vrši u levo, a ako se unese 1 rotacija se vrši u desno. Za bilo koju drugu vrednost ne vršiti rotaciju. Vrednosti promenljivih, kao i parametre k i s zadaje korisnik. Prikazati vrednosti promenljivih nakon rotacije.

Rešenje

```
1 #include <stdio.h>
```

Zbirka zadataka



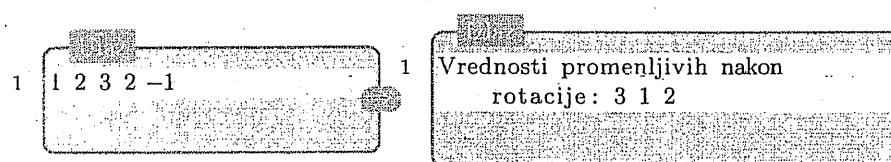
```

2 void main()
3 {
4     int a,b,c,k,s,p,i;
5     scanf ("%d%d%d",&a,&b,&c);
6     scanf ("%d",&k);
7     scanf ("%d",&s);
8     if ( s == -1)
9         for ( i = 0; i < k; i++)
10     {
11         p = a;
12         a = b;
  
```

Zbirka zadataka

```

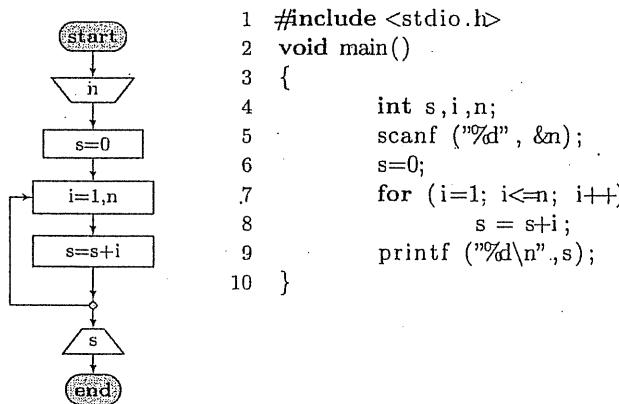
13             b = c;
14             c = p;
15         }
16     else if (s == 1)
17         for ( i = 0; i < k; i++ )
18         {
19             p = c;
20             c = b;
21             b = a;
22             a = p;
23         }
24     else
25         printf ("Nema rotacije");
26     printf ("Vrednosti promenljivih nakon rotacije: %d %d %d", a, b, c);
27 }
```



Zadatak 1.13

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje sumu prvih N prirodnih brojeva.

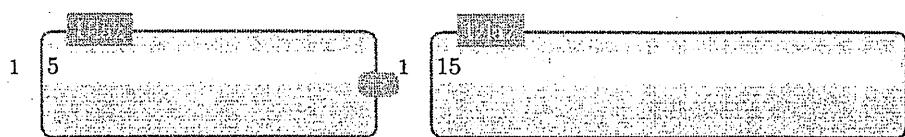
Rešenje



```

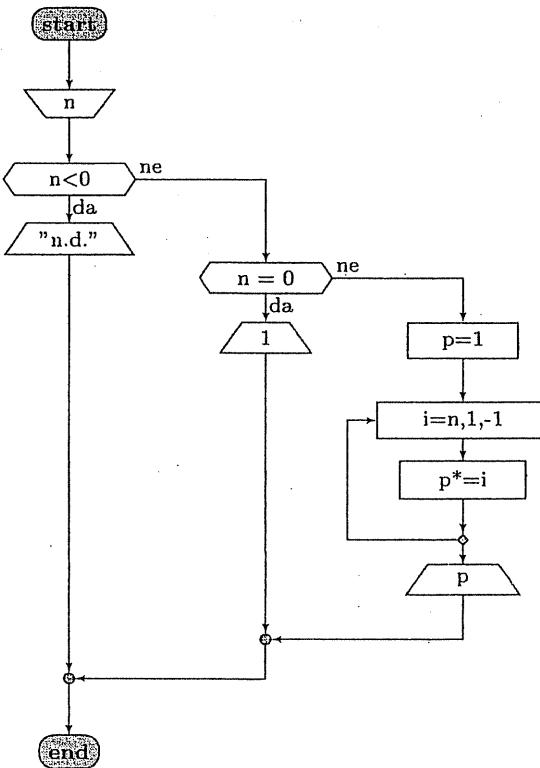
1 #include <stdio.h>
2 void main()
3 {
4     int s,i,n;
5     scanf ("%d", &n);
6     s=0;
7     for ( i=1; i<=n; i++)
8         s = s+i;
9     printf ("%d\n",s);
10 }
```

Zbirka zadataka

**Zadatak 1.14**

Nacrtati struktturni dijagram toka algoritma i napisati struktturni program na programskom jeziku C koji određuje i prikazuje faktorijal unetog celog broja. Faktorijal broja n definiše se kao

$$n! = \begin{cases} \text{nije def.,} & n < 0 \\ 1, & n = 0 \\ 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n, & n \geq 1 \end{cases}$$

Rešenje

```

1 #include <stdio.h>
2 void main()
3 {
4     int i, n, p;
    
```

Zbirka zadataka

```

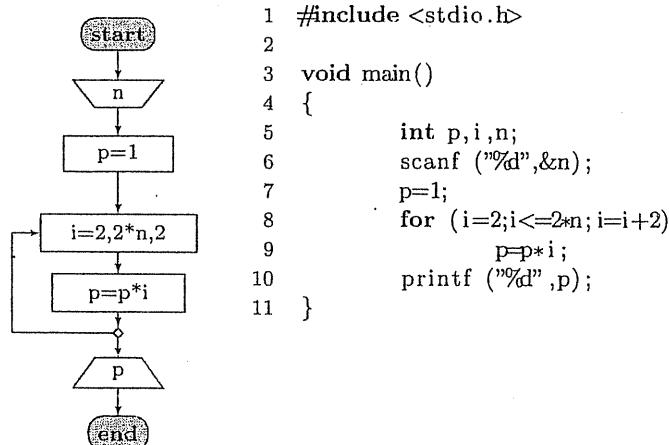
5     scanf ("%d", &n);
6     if ( n < 0 )
7         printf ("Faktorijal broja %d nije definisan", n);
8     else
9         if ( n == 0 )
10            printf ("Faktorijal broja %d je 1", n);
11     else
12     {
13         p=1;
14         for ( i = n; i > 1; i-- )
15             p *= i;
16         printf ("Faktorijal broja %d je %d", n, p);
17     }
18 }
```



Zadatak 1.15

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje proizvod prvih N parnih prirodnih brojeva.

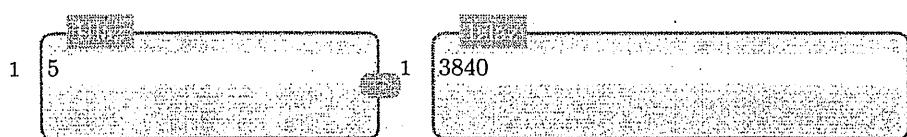
Rešenje



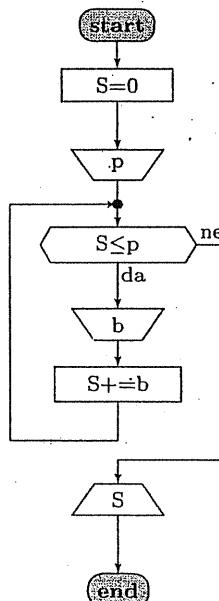
```

1 #include <stdio.h>
2
3 void main()
4 {
5     int p,i,n;
6     scanf ("%d",&n);
7     p=1;
8     for ( i=2;i<=2*n; i=i+2)
9         p=p*i;
10    printf ("%d",p);
11 }
```

Zbirka zadataka

**Zadatak 1.16**

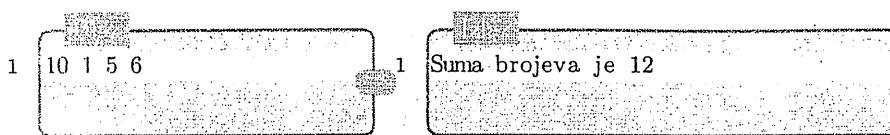
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program za sumiranje brojeva. Brojevi se unose sa tastature redom, a sumiranje se vrši sve dok suma ne postane veća od unapred zadate vrednosti.

Rešenje

```

1 #include <stdio.h>
2 void main()
3 {
4     int i,p,b,S;
5     S = 0;
6     scanf ("%d",&p);
7     while ( S <= p )
8     {
9         scanf ("%d",&b);
10        S += b;
11    }
12    printf ("Suma brojeva je %d",S);
13 }
  
```

Zbirka zadataka

**Zadatak 1.17**

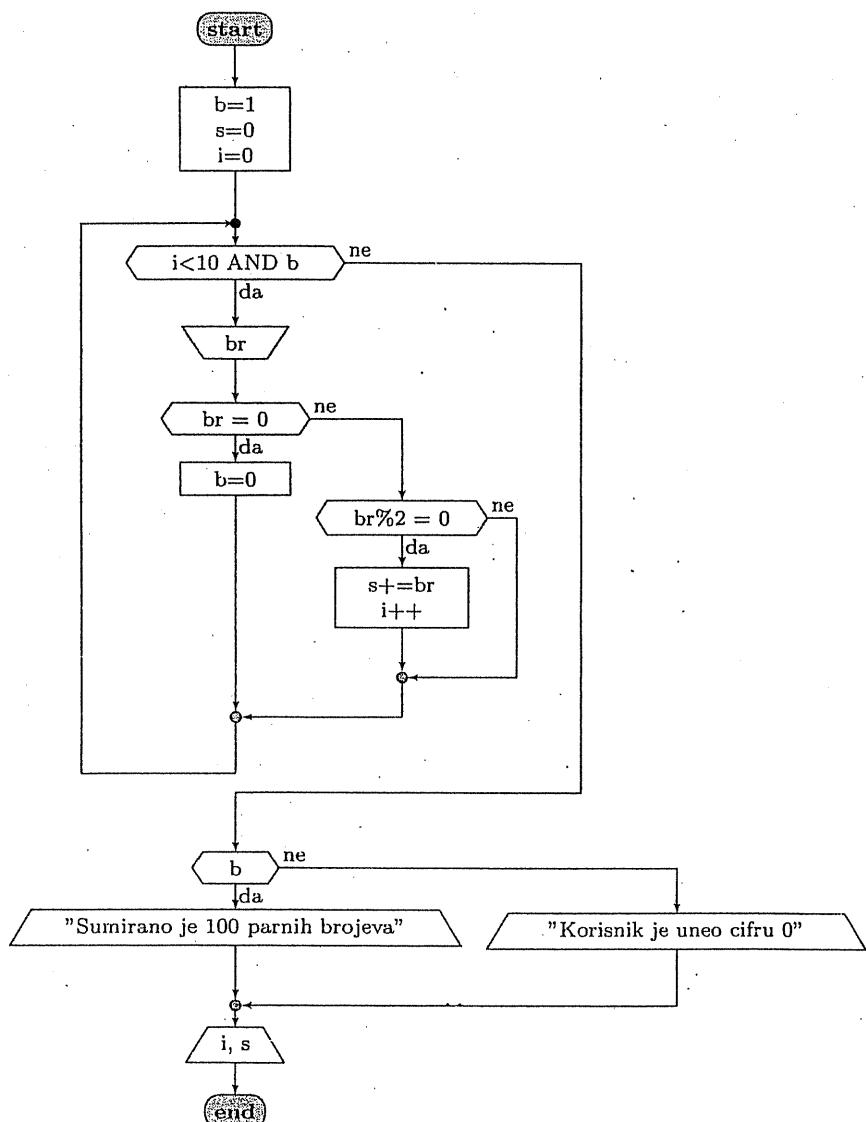
Nacrtati strukturni dijagram toka algoritma i na programsku jeziku C napisati program koji određuje sumu prvih 100 unetih parnih brojeva. Brojeve unosi korisnik redom, sve dok ne unese nulu, ili 100 parnih brojeva. Kada se unese nula prekinuti sumiranje brojeva bez obzira da li je izvršeno sumiranje tačno 100 parnih brojeva. Prikazati razlog izlaska iz programa, sumu brojeva i koliko je ukupno brojeva sumirano.

Rešenje

```

1 #include <stdio.h>
2
3 void main()
4 {
5     int b=1,s=0,br,i=0;
6     while (i<10 && b)
7     {
8         printf("Unesite broj:\n");
9         scanf("%d",&br);
10        if (br==0)
11            b=0;
12        else if (br%2==0)
13        {
14            s+=br;
15            i++;
16        }
17    }
18    if (b)
19        printf ("Sumirano je 100 parnih brojeva\n");
20    else
21        printf ("Korisnik je uneo cifru 0\n");
22    printf ("Suma prvih %d parnih brojeva je %d\n", i, s);
23 }
```

Zbirka zadataka



1 2 3 8 4
2 6 1 12 0

1 Unesite broj:
2 Unesite broj:
3 ...
4 Unesite broj:
5 Korisnik je uneo cifru 0
6 Suma prvih 5 parnih brojeva je 32

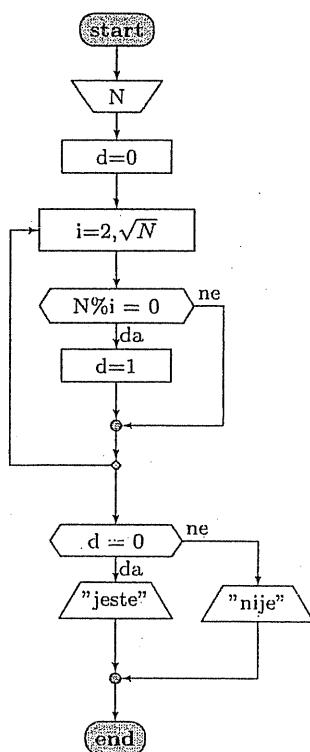
Zbirka zadataka

Zadatak 1.18

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje da li je uneti prirodni broj N prost i prikazuje odgovarajuću poruku. Nije potrebno optimizovati algoritam tako da se izvršenje prekine čim se utvrdi da je uneti broj deljiv nekim brojem.

Napomena: broj je prost ukoliko nije deljiv ni jednim drugim brojem osim jedinicom i samim sobom.

Rešenje



```

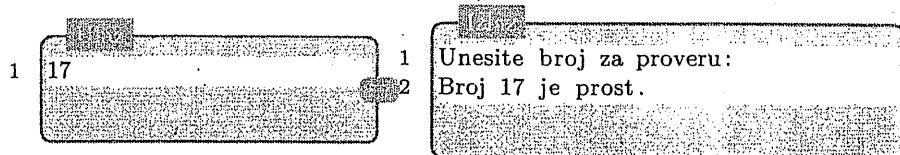
1 #include <stdio.h>
2 #include <math.h>
3 void main()
4 {
5     int N, i, d;
6     printf("Unesite broj za proveru:\n");
7     scanf("%d", &N);
8     d = 0;
9     for(i=2;i<=sqrt(N); i++)
  
```

Zbirka zadataka

```

10         if(N%i == 0)
11             d=1;
12         if (d==0)
13             printf("Broj %d je prost.\n",N);
14         else
15             printf("Broj %d nije prost.\n",N);
16     }

```

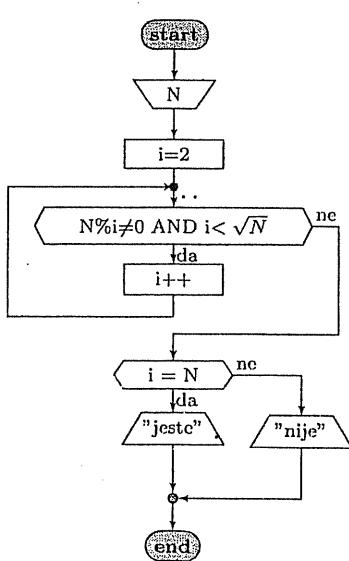


Zadatak 1.19

Nacrtati strukturalni dijagram toka algoritma i na programskom jeziku C napisati strukturalni program koji određuje da li je uneti prirodni broj N prost i prikazuje odgovarajuću poruku. Optimizovati algoritam tako da se izvršenje prekine čim se utvrdi da je uneti broj deljiv nekim brojem.

Napomena: broj je prost ukoliko nije deljiv ni jednim drugim brojem osim jedinicom i samim sobom.

Rešenje

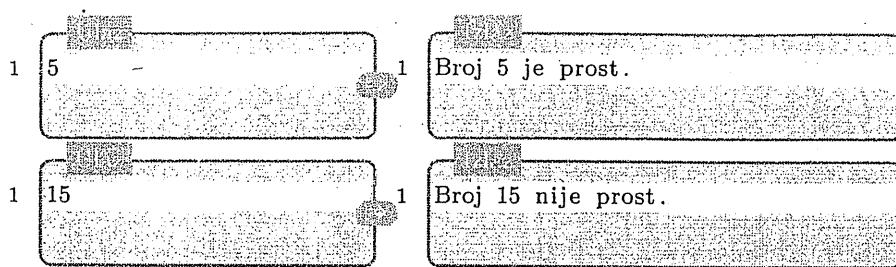


```

1 #include <stdio.h>
2 #include <math.h>
3 main()
4 {
5     int N, i;
6     scanf("%d",&N);
7     i=2;
8     while (N%i != 0 && i<sqrt(N))
9         ii==N)
11        printf("Broj %d je prost.\n",N);
12    else
13        printf("Broj %d nije prost.\n",N);
14 }

```

Zbirka zadataka

**Zadatak 1.20**

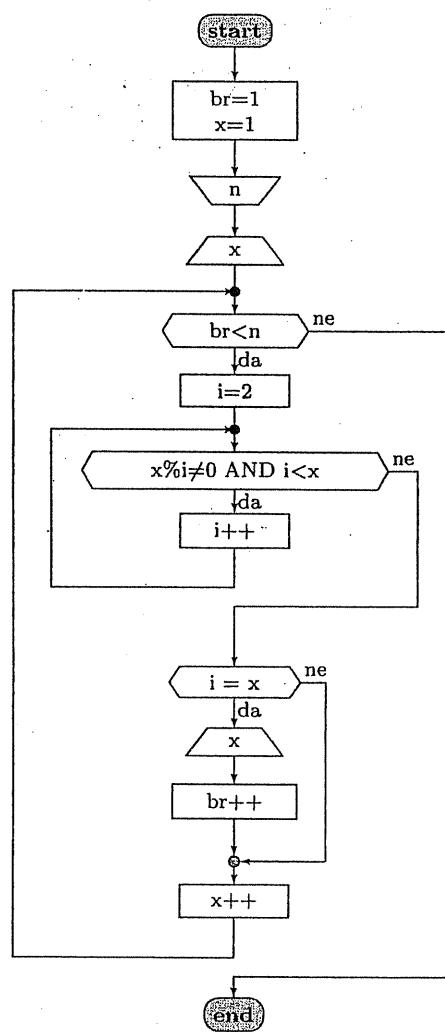
Nacrtati struktturni dijagram toka algoritma i na programskom jeziku C napisati struktturni program koji određuje i prikazuje prvih N prostih brojeva. Parametar N unosi korisnik.

Rešenje

```

1 #include <stdio.h>
2 void main()
3 {
4     int n,br=1,x=1,i;
5     scanf("%d",&n);
6     printf("Broj %d je prost\n",x);
7     while( br < n )
8     {
9         i=2;
10        while ( x%i != 0 && i < x )
11            i++;
12
13        if ( i == x )
14        {
15            printf("Broj %d je prost\n",x);
16            br++;
17        }
18        x++;
19    }
20 }

```



1 10

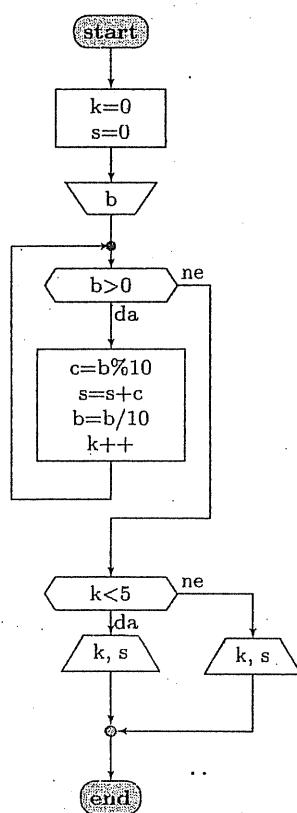
- | | |
|----|------------------|
| 1 | Broj 1 je prost |
| 2 | Broj 2 je prost |
| 3 | Broj 3 je prost |
| 4 | Broj 5 je prost |
| 5 | Broj 7 je prost |
| 6 | Broj 11 je prost |
| 7 | Broj 13 je prost |
| 8 | Broj 17 je prost |
| 9 | Broj 19 je prost |
| 10 | Broj 23 je prost |

Zbirka zadataka

Zadatak 1.21

Nacrtati strukturni dijagram toku algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje zbir cifara unetog N -tocifrenog pozitivnog celog broja. Prikazati i ukupan broj cifara zadatog broja.

Rešenje



```

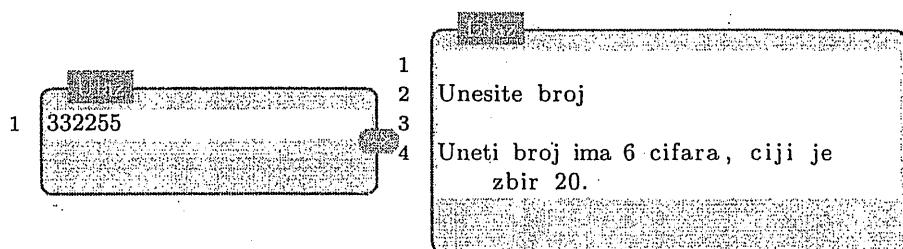
1 #include <stdio.h>
2 int main()
3 {
4     int b, c, k=0, s=0;
5     printf("\nUnesite broj\n");
6     scanf("%d",&b);
7     while (b>0)
8     {
9         c = b % 10;
10        s = s + c;
  
```

Zbirka zadataka

```

11         b = b / 10;
12         k++;
13     }
14     if (k < 5) //samo zbog pravopisa
15         printf("\nUneti broj ima %d cifre , ciji je zbir %d.\n",
16             k, s);
17     else
18         printf("\nUneti broj ima %d cifara , ciji je zbir %d.\n",
19             k, s);
20     }

```

**Zadatak 1.22**

Nacrtati struktturni dijagram toka algoritma i na programskom jeziku C napisati program koji prikazuje sve trocifrene brojeve čiji je zbir cifara deljiv zadatim brojem b . Broj b zadaje korisnik. Napomena: nije dozvoljeno korišćenje indeksiranih promenljivih.

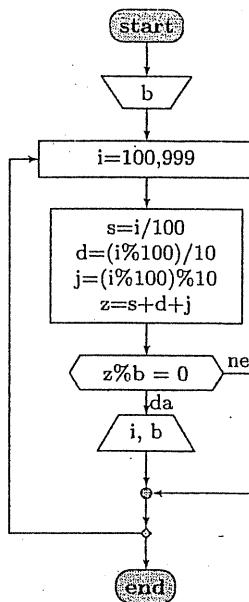
Rešenje

```

1 #include <stdio.h>
2 void main()
3 {
4     int s,d,j,i,z,b;
5     scanf("%d", &b);
6     for(i=100;i<1000;i++)
7     {
8         s=i/100;
9         d=(i%100)/10;
10        j=(i%100)%10;
11        z=s+d+j;
12        if(z%b==0)
13            printf("Zbir cifara broja %d deljiv je sa %d\n", i, b);
14    }
15 }

```

Zbirka zadataka



1

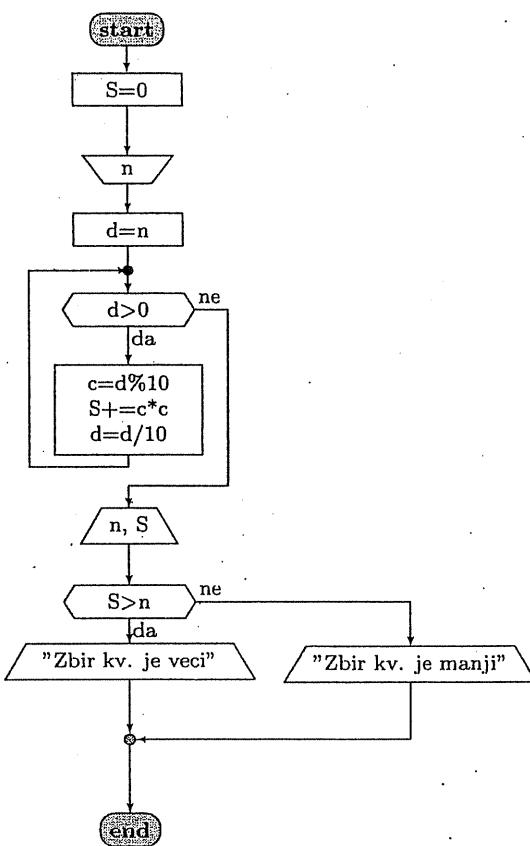
25

- | | |
|---|---------------------------------------|
| 1 | Zbir cifara broja 799 deljiv je sa 25 |
| 2 | Zbir cifara broja 889 deljiv je sa 25 |
| 3 | Zbir cifara broja 898 deljiv je sa 25 |
| 4 | Zbir cifara broja 979 deljiv je sa 25 |
| 5 | Zbir cifara broja 988 deljiv je sa 25 |
| 6 | Zbir cifara broja 997 deljiv je sa 25 |

Zadatak 1.23

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje zbir kvadrata cifara unetog prirodnog broja. Prikazati poruku da li je dobijeni zbir veći ili manji od unetog broja.
 Primer: za uneti broj 1234, zbir kvadrata cifara je $1+4+9+16=30$.

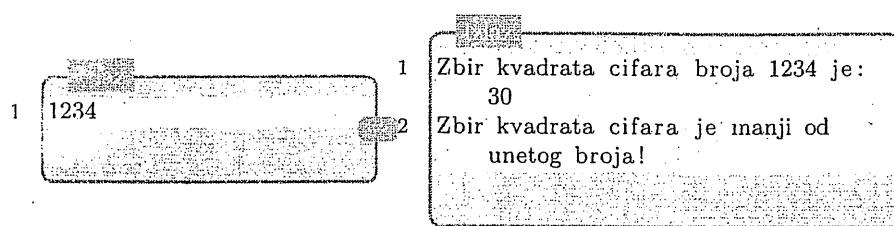
Rešenje**Zbirka zadataka**



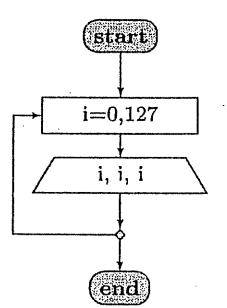
```

1 #include <stdio.h>
2 void main()
3 {
4     int n,S = 0,c,d;
5     scanf ("%d",&n);
6     d = n;
7     while ( d > 0 )
8     {
9         c = d % 10;
10        S += c * c;
11        d = d / 10;
12    }
13    printf ("Zbir_kvadrata_cifara_broja_je:%d\n",n,S);
14    if ( S > n )
15        printf("Zbir_kvadrata_cifara_je_veci_od_unetog_broja!\n");
16    else
17        printf("Zbir_kvadrata_cifara_je_manji_od_unetog_
broja!\n");
18 }
  
```

Zbirka zadataka

**Zadatak 1.24**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati program koji prikazuje ASCII tabelu u formatu: kod (dekadno), kod (heksadecimalno) i ASCII znak.

Rešenje

```

1 #include <stdio.h>
2 void main()
3 {
4     int i;
5     for (i=0; i <= 127; i++)
6         printf("%d%4x %c\n", i, i, i);
7 }
  
```

1 alg. nema ulaz

1	...	
2	45	2d
3	46	2e
4	47	2f
5	48	30
6	49	31
7	...	
8	65	41
9	66	42
10	67	43
11	...	

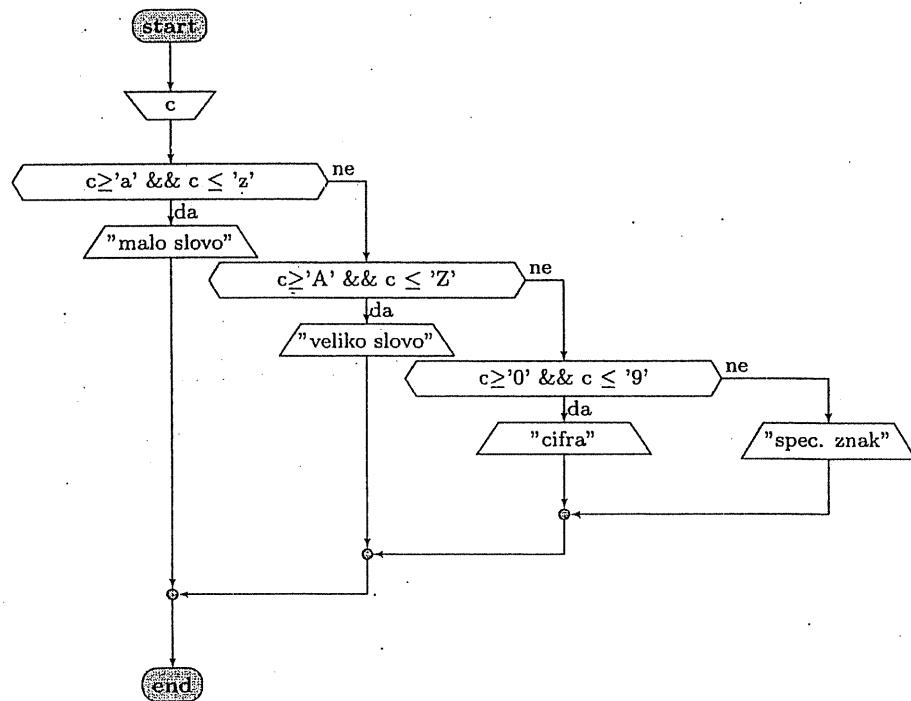
Zadatak 1.25

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati struk-

Zbirka zadataka

turni program koji određuje da li je uneti karakter malo ili veliko slovo, cifra ili specijalni znak i prikazuje odgovarajuću poruku.

Rešenje



```

1 #include <stdio.h>
2 void main()
3 {
4     char c;
5     scanf("%c",&c);
6     if (c >= 'a' && c <= 'z')
7         printf("Karakter %c je malo slovo.\n", c);
8     else if (c >= 'A' && c <= 'Z')
9         printf("Karakter %c je veliko slovo.\n", c);
10    else if (c >= '0' && c <= '9')
11        printf("Karakter %c je cifra.\n", c);
12    else
13        printf("Karakter %c je specijalni karakter.\n", c);
14 }
  
```

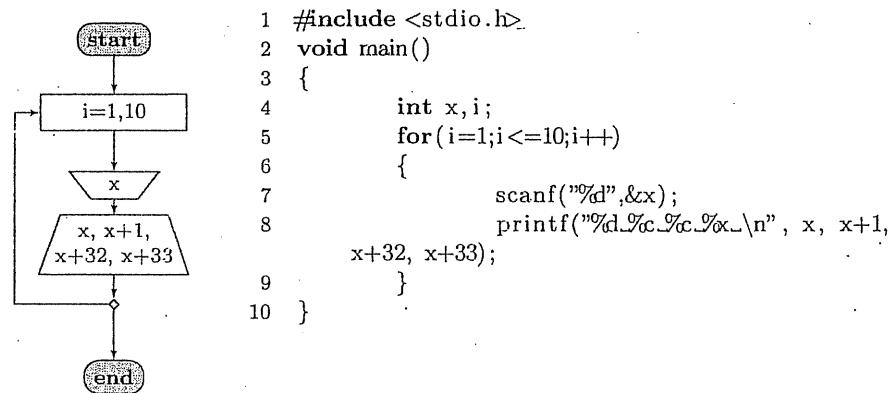
Zbirka zadataka



Zadatak 1.26

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati program koji za 10 unetih celobrojnih vrednosti prikazuje ASCII kod koji odgovara zadatoj vrednosti, sledeće veliko slovo u ASCII tabeli, malo slovo koje odgovara ASCII znaku unete vrednosti i heksadecimalni ekvivalent koji odgovara narednom malom slovu u ASCII tabeli. Smatrati da se unose samo vrednosti koje odgovaraju velikim slovima u ASCII tabeli (između 65 i 90).

Rešenje



1	65
2	66
3	67
4	68
5	69
6	70
7	71
8	72
9	73
10	90

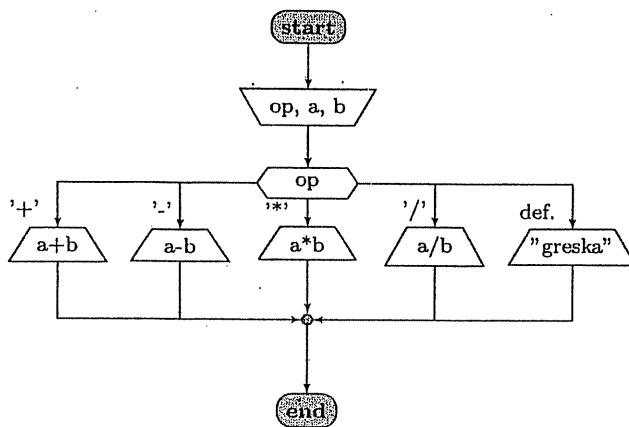
1	65 B a 62
2	66 C b 63
3	67 D c 64
4	68 E d 65
5	69 F e 66
6	70 G f 67
7	71 H g 68
8	72 I h 69
9	73 J i 6a
10	90 [z 7b

Zbirka zadataka

Zadatak 1.27

Nacrtati struktturni dijagram toka algoritma i na programskom jeziku C napisati struktturni program kojim se vrši unos jednog karaktera, koji predstavlja aritmetičku operaciju i dva cela broja. Prikazati unesene brojeve i rezultat izvršene aritmetičke operacije nad njima. Ukoliko se za aritmetičku operaciju unese neki karakter različit od +, -, * ili /, umesto rezultata prikazati poruku o tome da je operacija nepoznata.

Rešenje



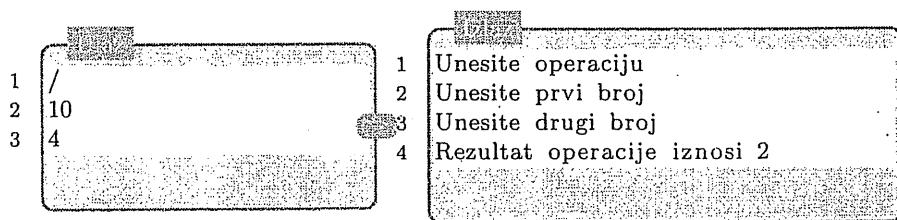
```

1 #include <stdio.h>
2 void main()
3 {
4     int a,b;
5     char op;
6     printf("Unesite_operaciju\n");
7     scanf("%c",&op);
8     printf("Unesite_prvi_broj\n");
9     scanf("%d",&a);
10    printf("Unesite_drugi_broj\n");
11    scanf("%d",&b);
12    printf("Rezultat_operacije");
13    switch(op)
14    {
15        case '+':
16            printf("iznosi %d\n",a+b);
17            break;
18        case '-':
19            printf("iznosi %d\n",a-b);
20            break;
21        case '*':
22            printf("iznosi %d\n",a*b);
23            break;
  
```

Zbirka zadataka

```

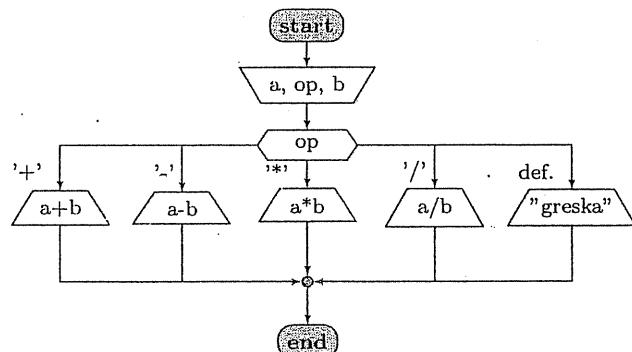
24     case '/':
25         printf("Iznosi %d\n", a/b);
26         break;
27     default:
28         printf("Nije poznat.\n");
29         break;
30     }
31 }
```

**Zadatak 1.28**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program kojim određuje i prikazuje rezultat binarne aritmetičke operacije nad celim brojevima. Korisnik zadaje izraz u formatu

operand_1 operacija operand_2

gde je *operacija* jedan od karaktera '+', '-', '*' ili '/' (primer: 4+3). Ukoliko je za aritmetičku operaciju unet neki karakter različit od +, -, * ili /, umesto rezultata prikazati poruku o tome da je operacija nepoznata.

Rešenje

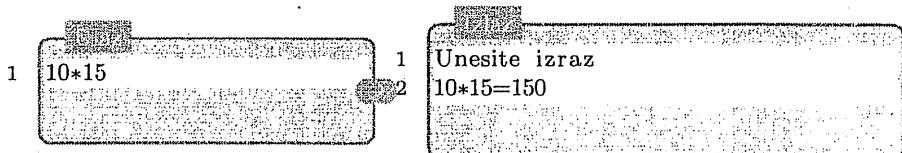
```

1 #include <stdio.h>
2 void main()
3 {
```

Zbirka zadataka

```

4     int a,b, rez;
5     char op;
6     printf("Unesite izraz\n");
7     scanf("%d%d",&a,&op,&b);
8     printf("%d%d",a,op,b);
9     printf("\n");
10    switch(op)
11    {
12        case '+':
13            printf("%d\n",a+b);
14            break;
15        case '-':
16            printf("%d\n",a-b);
17            break;
18        case '*':
19            printf("%d\n",a*b);
20            break;
21        case '/':
22            printf("%d\n",a/b);
23            break;
24        default:
25            printf("Greska.\n");
26            break;
27    }
28 }
```



1.3 Iterativni postupci

Zadatak 1.29

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje N -ti član Fibonačijevog niza. Parametar N zadaje korisnik. Elementi Fibonačijevog niza definisani su formulom

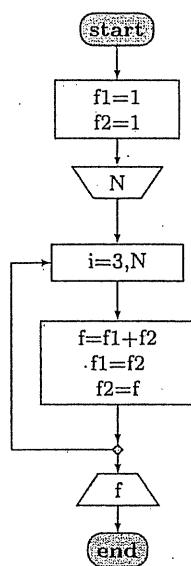
$$f_i = f_{i-1} + f_{i-2},$$

za $i \geq 3$, gde su $f_1 = 1$ i $f_2 = 1$.

Napomena: smatrati da će korisnik zadati $N \geq 3$.

Rešenje

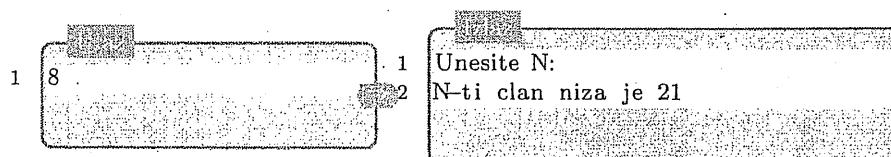
Zbirka zadataka



```

1 #include <stdio.h>
2 void main()
3 {
4     int f1=1, f2=1, f, N, i;
5     printf("Unesite N:\n");
6     scanf("%d", &N);
7     for (i = 3; i <= N; i++)
8     {
9         f = f1 + f2;
10        f1 = f2;
11        f2 = f;
12    }
13    printf("N-ti clan niza je %d\n";
14 }

```



Zadatak 1.30

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje vrednost $x = \sqrt[a]{a}$, gde je $a > 0$, primenom

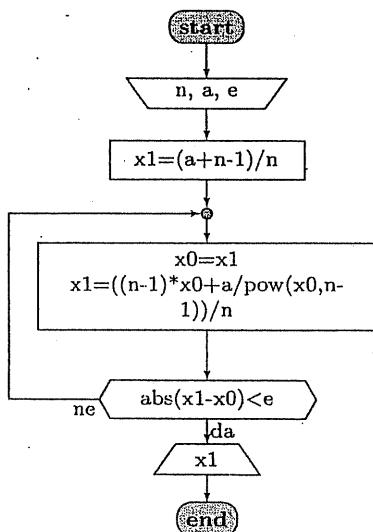
Zbirka zadataka

iterativnog postupka

$$\begin{aligned}x_0 &= \frac{(a+n-1)}{n} \\x_{i+1} &= \frac{\left((n-1) \cdot x_i + \frac{a}{x_i^{n-1}}\right)}{n},\end{aligned}$$

gde je $i = 0, 1, 2, \dots$. Izračunavanje prekinuti kada je $|x_{i+1} - x_i| \leq \epsilon$, gde je ϵ zadata tačnost.

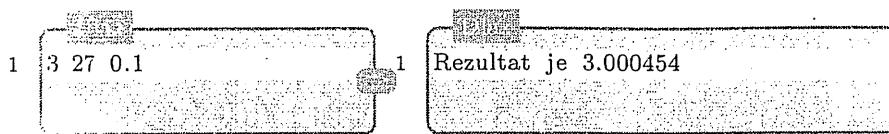
Rešenje



```

1 #include <stdio.h>
2 #include <math.h>
3 void main()
4 {
5     int n,a;
6     double e,x0,x1;
7     scanf ("%d%d%lf",&n,&a,&e);
8     x1 = (a + n - 1) / n;
9     do
10    {
11        x0 = x1;
12        x1 = ( (n - 1) * x0 + a / pow ( x0 , n - 1 ) ) / n;
13    } while ( abs ( x1 - x0 ) > e );
14    printf ("Rezultat_je.%lf",x1);
15 }
  
```

Zbirka zadataka

**Zadatak 1.31**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje vrednost funkcije $ch(x)$ primenom razvoja u red

$$ch(x) = \sum_{k=0}^{\infty} \frac{x^{2k}}{(2k)!}$$

Izračunavanje prekinuti kada relativna vrednost priraštaja sume postane manja od zadate tačnosti ϵ .

Napomena: Ovaj red važi za $|x| < 4$.

Rešenje

K-ti član sume a_k je

$$a_k = \frac{x^{2k}}{(2k)!}$$

Sumiranje treba prekinuti kada je razlika sukcesivnih vrednosti sume

$$\frac{|S_{i+1} - S_i|}{S_i} \approx \left| \frac{a_{i+1}}{S_{i+1}} \right|$$

Ukoliko imamo prethodno određeni član sume a_k , vrednost a_{k+1} možemo odrediti na osnovu

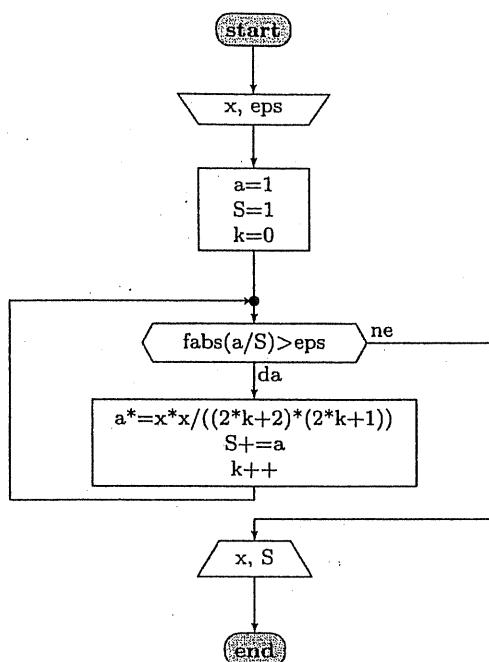
$$\frac{a_{k+1}}{a_k} = \frac{\frac{x^{2(k+1)}}{(2(k+1))!}}{\frac{x^{2k}}{(2k)!}} = \frac{x^2}{(2k+2)(2k+1)},$$

odnosno

$$a_{k+1} = \frac{x^2}{(2k+2)(2k+1)} a_k.$$

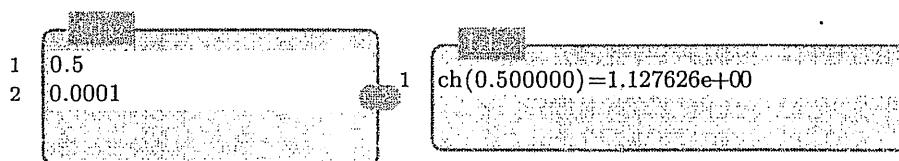
Početna vrednost dobija se zamenom $k = 0$ u izraz a_k , pa je $a_0 = 1$. Na osnovu ovoga vrednost sume se može odrediti na sledeći način:

Zbirka zadataka



```

1 #include <stdio.h>
2 void main()
3 {
4     int k;
5     float x,a,S,c,eps;
6     scanf("%f%f",&x,&eps);
7     a=1;
8     S=1;
9     k=0;
10    while(fabs(a/S)>eps)
11    {
12        a=x*x/((2*k+2)*(2*k+1));
13        S+=a;
14        k++;
15    }
16    printf("ch(%f)=%e\n",x,S);
17 }
  
```



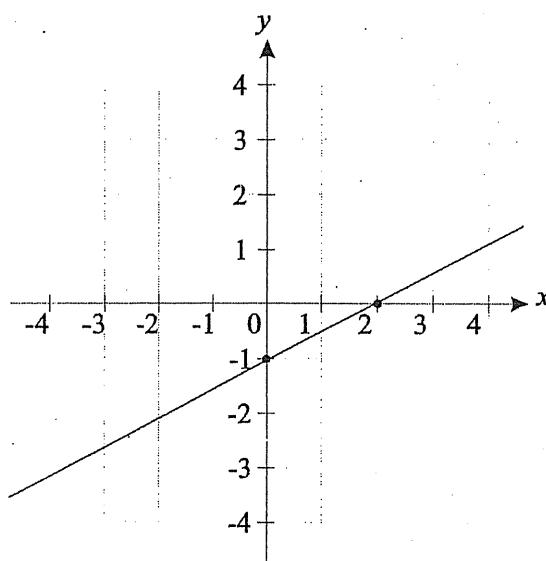
1.4 Geometrijski problemi

Zadatak 1.32

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje sa koje strane prave sa slike 1.1 se nalazi tačka p . Koordinate (x, y) tačke p zadaje korisnik. Jednačina prave kroz dve tačke data je sa

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1).$$

Napomena: tačke na pravoj proizvoljno uključiti u bilo koju oblast.



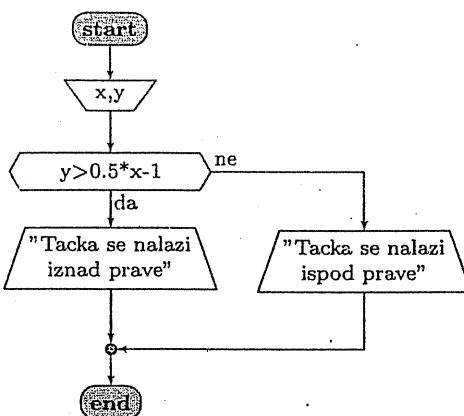
Slika 1.1

Rešenje

Na osnovu slike 1.1, za tačke kroz koje prolazi prava izaberimo $t_1 = (0, -1)$ i $t_2 = (2, 0)$. Zamenom ovih tačaka u jednačini pravc dobijamo jednačinu konkretnе prave datu na slici kao

$$y = \frac{1}{2}x - 1.$$

Za tačke koje nisu na pravoj važe nejednakosti $y < \frac{1}{2}x - 1$, za tačke ispod, i $y > \frac{1}{2}x - 1$ za tačke iznad prave, što se može proveriti zamenom konkretnе tačke, npr $t = (0, 0)$ u jednačini prave.



```

1 #include <stdio.h>
2
3 void main()
4 {
5     float x,y;
6     scanf("%f",&x);
7     scanf("%f",&y);
8     if(y>0.5*x - 1)
9         printf("Tacka se nalazi iznad prave\n");
10    else
11        printf("Tacka se nalazi ispod prave\n");
12 }

```

1 5
2 -0.25 1 Tacka se nalazi ispod prave

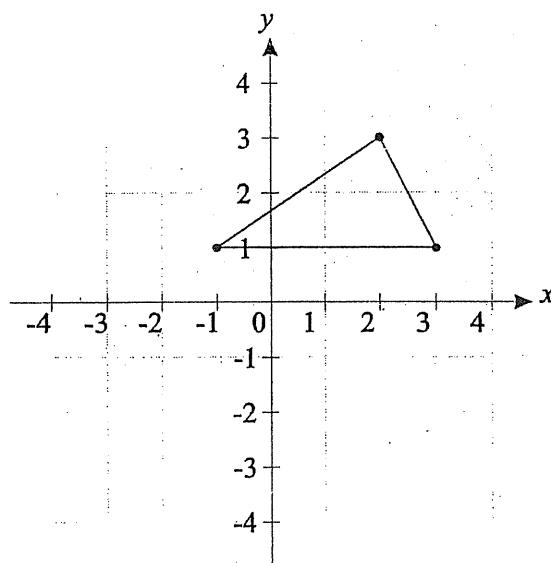
Zadatak 1.33

Nacrtati struktturni dijagram toka algoritma i na programskom jeziku C napisati struktturni program koji određuje i prikazuje koliko se tačaka, od ukupno N , nalazi unutar, a koliko van trougla zadatog na slici. Parametar N zadaće korisnik, a za njim redom i koordinate tačaka (x, y) . Jednačina prave kroz dve tačke data je sa

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1).$$

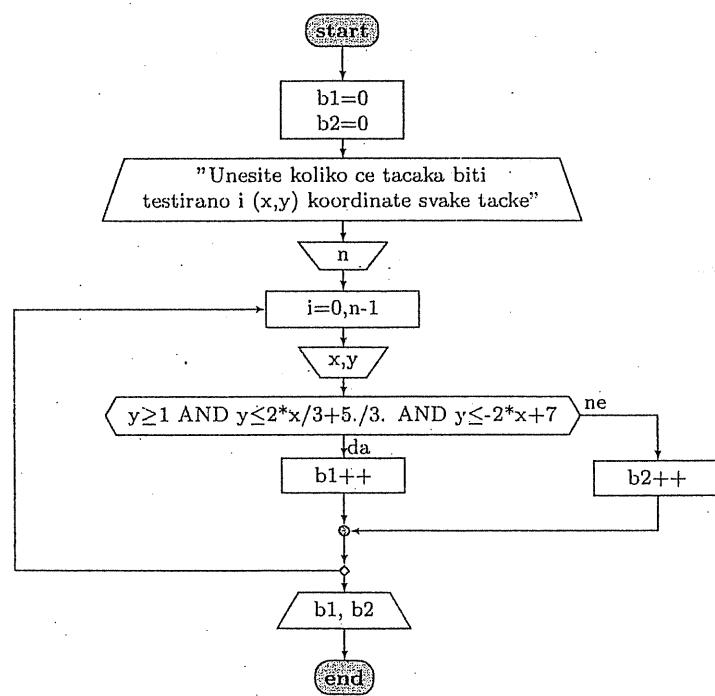
Napomena: smatrati da su stranice trougla deo unutrašnje oblasti.

Zbirka zadataka



Slika 1.2

Rešenje



Zbirka zadataka

```

1. #include <stdio.h>
2. void main()
3. {
4.     float x, y;
5.     int n, i, b1=0, b2=0;
6.     printf("Unesite koliko ce tacaka biti testirano i (x,y) koordinate svake tacke\n");
7.     scanf("%d", &n);
8.     for(i=0; i<n; i++)
9.     {
10.         scanf("%f", &x);
11.         scanf("%f", &y);
12.         if( y >= 1 && y <= 2*x/3 + 5./3. && y <= -2*x + 7 )
13.             b1++;
14.         else
15.             b2++;
16.     }
17.     printf("Unutar trougla je %d tacaka a van trougla je %d tacaka\n", b1, b2);
18. }
```

1. 6
 2. 2.0 2.0
 3. 1.0 2.0
 4. 3.0 1.0
 5. 1.0 1.0
 6. 4.0 3.0
 7. -2.0 1.0

1. Unesite koliko ce tacaka biti testirano i (x,y) koordinate svake tacke
 2. Unutar trougla je 4 tacaka a van trougla je 2 tacaka

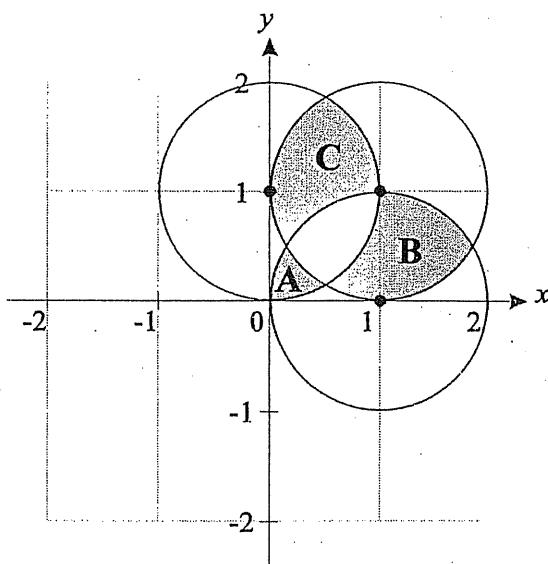
Zadatak 1.34

Dato je N tačaka u ravni koordinatama (x, y) . Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturu program koji za svaku tačku ispituje da li pripada nekoj od oblasti A, B ili C. Na izlazu prikazati redni broj tačke, njene koordinate i oznaku oblasti kojoj pripada (* ako ne pripada ni jednoj od zadatih oblasti). Oblasti su definisane kružnicama, prikazanim na slici 1.3, koje su opisane jednačinama na sledeći način:

$$\begin{aligned} K_1 : \quad (x - 1)^2 + y^2 &= 1 \\ K_2 : \quad (x - 1)^2 + (y - 1)^2 &= 1 \\ K_3 : \quad x^2 + (y - 1)^2 &= 1 \end{aligned}$$

Napomena: za tačke na kružnici smatrati da pripadaju unutrašnjosti kruga.

Zbirka zadataka



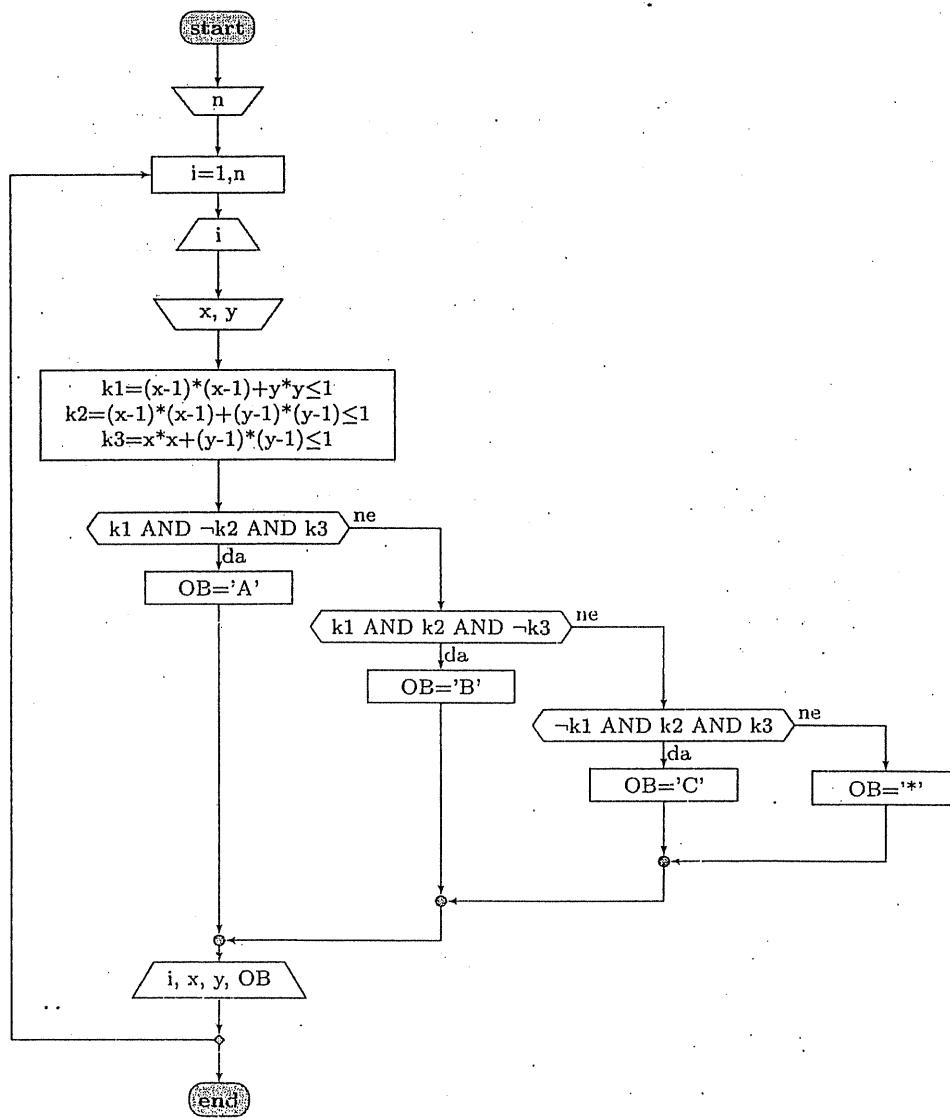
Slika 1.3

Rešenje

```

1 #include <stdio.h>
2 void main()
3 {
4     int n, i, k1, k2, k3;
5     char OB;
6     float x,y;
7     printf("Unesite koliko tacaka cete testirati:\n");
8     scanf("%d", &n);
9     for(i = 1; i <= n; i++)
10    {
11        printf("Unesite x,i y koordinate tacke:\n", i);
12        scanf("%f%f", &x,&y);
13        k1 = (x-1)*(x-1) + y*y <= 1;
14        k2 = (x-1)*(x-1) + (y-1)*(y-1) <= 1;
15        k3 = x*x + (y-1)*(y-1) <= 1;
16        if(k1 && !k2 && k3)
17            OB = 'A';
18        else if(k1 && k2 && !k3)
19            OB = 'B';
20        else if (!k1 && k2 && k3)
21            OB = 'C';
22        else
23            OB = '*';

```



```

24     printf("%d. stacka sa koordinatama %4.2f,%4.2f je u
25     oblasti %c\n", i,x,y,OB);
26   }
  
```

Zbirka zadataka

- 1 4
2 0.2 0.3
3 0.2 0.68
4 0.85 0.12
5 0.95 1.86

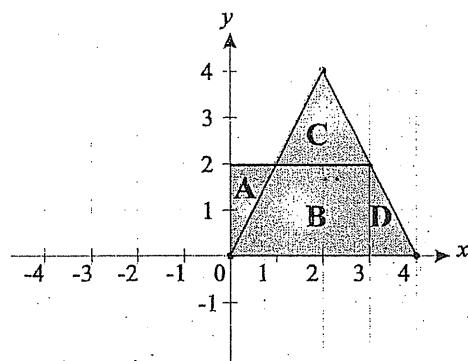
- 1 Unesite koliko tačaka cete testirati:
 2 Unesite x i y koordinate 1. tacke:
 3 1. tacka sa koordinatama 0.20,0.30
 je u oblasti A
 4 Unesite x i y koordinate 2. tacke:
 5 2. tacka sa koordinatama 0.20,0.68
 je u oblasti C
 6 Unesite x i y koordinate 3. tacke:
 7 3. tacka sa koordinatama 0.85,0.12
 je u oblasti B
 8 Unesite x i y koordinate 4. tacke:
 9 4. tacka sa koordinatama 0.95,1.86
 je u oblasti *

Zadatak 1.35

Dato je 10 tačaka u ravni koordinatama (x, y) . Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji za svaku tačku ispituje da li pripada nekoj od oblasti A, B, C ili D, označenih na slici 1.4. Na izlazu prikazati oznaku oblasti kojoj pripada. Ukoliko ne pripada ni jednoj oblasti ispisati poruku "Tacka ne pripada ni jednoj oblasti". Jednačina prave kroz dve tačke data je sa

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1).$$

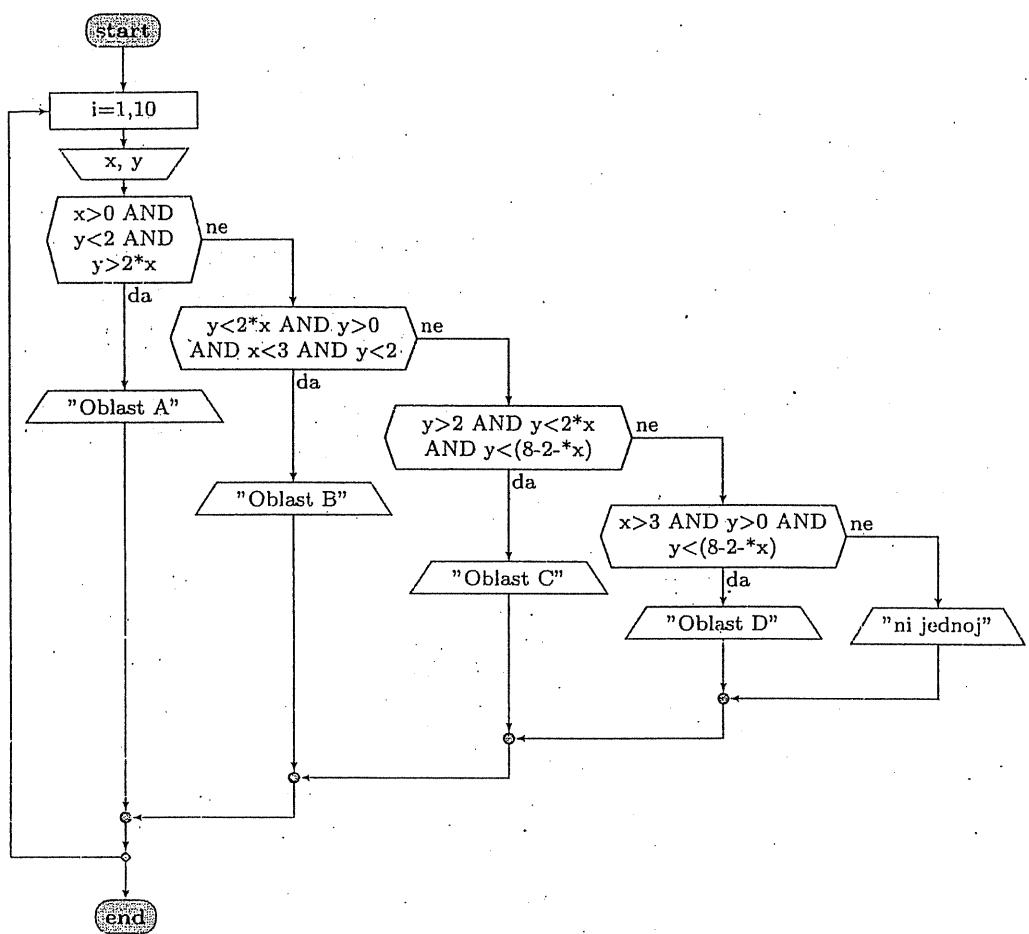
Napomena: za tačke na pravama proizvoljno odrediti pripadnost oblastima.



Slika 1.4

Rešenje

Zbirka zadataka



```

1 #include "stdio.h"
2
3 void main()
4 {
5     float x,y;
6     int i;
7
8     for (i=1;i<=10;i++)
9     {
10         scanf ("%f%f",&x,&y);
11         if (x>0 && y<2 && y>2*x)
12             printf("Oblast_A\n");
13         else if (y<2*x && y>0 && x<3 && y<2)
14             printf("Oblast_B\n");
15         else if (y>2 && y<2*x && y<(8-2*x))
16             printf("Oblast_C\n");
17         else if (x>3 && y>0 && y<(8-2*x))
```

Zbirka zadataka

```
18         printf("Oblast D\n");
19     else
20         printf("Tacka ne pripada ni jednoj oblasti\n");
21     }
22 }
```

1	1	1
2	2.0	8
3	-1	-0.5
4	2	3
5	0.1	1
6	3.2	0.5
7	10	11.1
8	2	3.5
9	0	0
10	15	-2.5

1	Oblast B
2	Tacka ne pripada ni jednoj oblasti
3	Tacka ne pripada ni jednoj oblasti
4	Oblast C
5	Oblast A
6	Oblast D
7	Tacka ne pripada ni jednoj oblasti
8	Oblast C
9	Tacka ne pripada ni jednoj oblasti
10	Tacka ne pripada ni jednoj oblasti

Zbirka zadataka

2

Linearne indeksirane strukture podataka

2.1 Nizovi

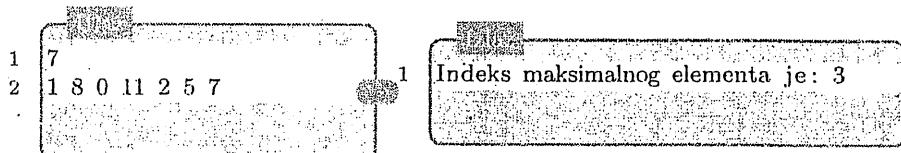
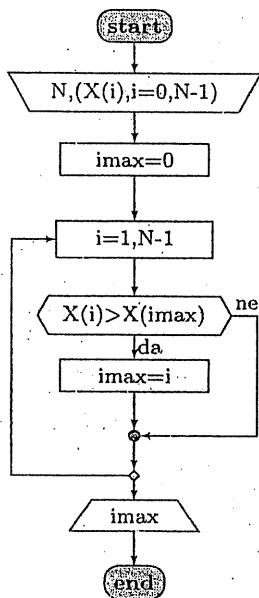
2.1.1 Brojevni nizovi

Zadatak 2.36

Nacrtati struktturni dijagram toka algoritma i na programskom jeziku C napisati struktturni program koji određuje i prikazuje indeks maksimalnog elementa u celobrojnom nizu X sa N elemenata.

Rešenje

```
1 #include <stdio.h>
2 void main()
3 {
4     int i,imax,X[30],N;
5     scanf ("%d",&N);
6     for ( i = 0; i < N; i++)
7         scanf ("%d",&X[i]);
8     imax=0;
9     for ( i = 1; i < N; i++)
10        if ( X[i] > X[imax])
11            imax=i;
12     printf ("Indeks_maksimalnog_elementa_je:%d", imax);
13 }
```



Zadatak 2.37

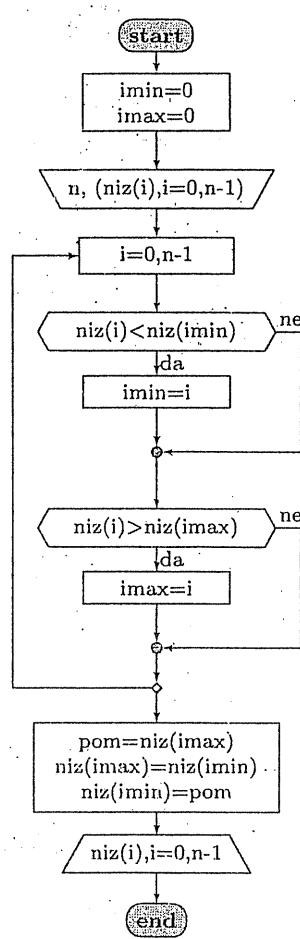
Nacrtati strukturalni dijagram toka algoritma i na programskom jeziku C napisati strukturalni program koji vrši zamenu mesta minimalnom i maksimalnom elementu niza. Broj elemenata niza i elemente niza zadaje korisnik. Prikazati niz nakon transformacije.

Rešenje

```

1 #include <stdio.h>
2 void main()
3 {
4     int niz[100],n,i,imin,imax, pom;
5     imin=0;
6     imax=0;
7     scanf("%d",&n);
8     for(i = 0;i<n;i++)
9         scanf("%d",&niz[i]);
10    for(i=0;i<n; i++)
  
```

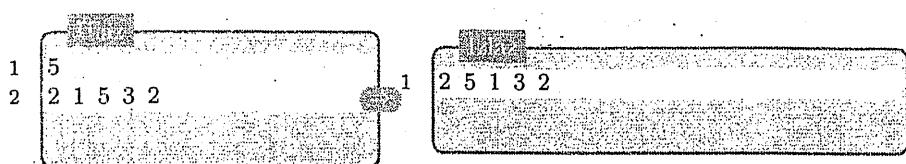
Zbirka zadataka



```

11  {
12      if(niz[i] < niz[imin])
13          imin = i;
14      if(niz[i]>niz[imax])
15          imax = i;
16  }
17  pom = niz[imax];
18  niz[imax] = niz[imin];
19  niz[imin] = pom;
20  for(i=0;i<n; i++)
21      printf("%d ", niz[i]);
22  }
  
```

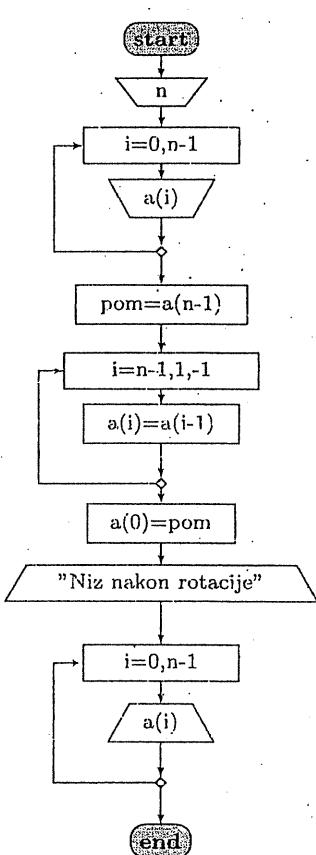
Zbirka zadataka



Zadatak 2.38

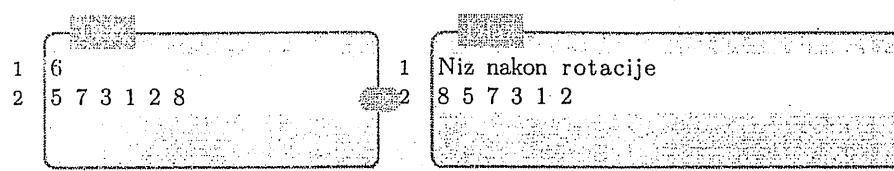
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji elemente niza A_N rotira za jedno mesto u desno. Broj elemenata niza i vrednosti elemenata niza zadaje korisnik. Prikazati niz nakon transformacije.

Rešenje

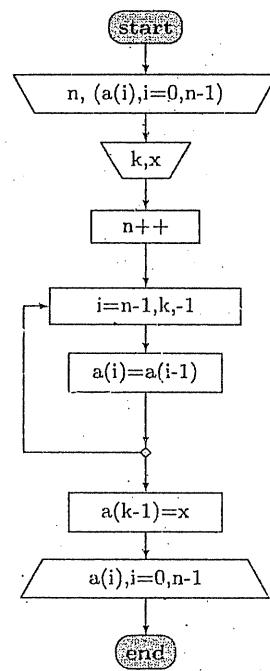


```

1 #include <stdio.h>
2 void main()
3 {
4     int i,pom,a[50],n;
5     scanf ("%d",&n);
6     for(i=0;i<n;i++)
7         scanf ("%d",&a[i]);
8     pom=a[n-1];
9     for(i=n-1;i>0;i--)
10        a[i]=a[i-1];
11     a[0]=pom;
12     printf("Niz nakon
13     rotacije\n");
14     for(i=0;i<n;i++)
15         printf ("%d ",a[i]);
}
  
```

**Zadatak 2.39**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji u niz A od N elemenata umeće novi element na k -tu poziciju u nizu (indeks novog elementa je $k - 1$). Broj elemenata niza i vrednosti elemenata zadaje korisnik. Nakon unosa niza korisnik zadaje vrednost novog elementa i poziciju na koju je potrebno umetnuti element. Prikazati niz nakon umetanja vrednosti.

Rešenje

```

1 #include <stdio.h>
2 void main()
3 {
4     int n, i, x, a[10], k;
5     printf("Uneti_broj_elemenata_niza:\n");
6     scanf("%d",&n);
7     printf("Uneti_elemente_niza:\n");
8     for (i=0;i<n;i++)
  
```

Zbirka zadataka

```

9      scanf("%d",&a[i]);
10     printf("Unesite poziciju novog elementa:\n");
11     scanf("%d",&k);
12     printf("Unesite vrednost novog elementa:\n");
13     scanf("%d",&x);
14     n++;
15     for(i=n-1;i>=k;i--)
16         a[i]=a[i-1];
17     a[k-1]=x;
18     printf("Izgled niza nakon umetanja:\n");
19     for(i=0;i<n;i++)
20         printf("%d ",a[i]);
21     printf("\n");
22 }

```

1 5
2 1 3 7 8 9
3 3
4 25

1 Uneti broj elemenata niza:
2 Uneti elemente niza:
3 Unesite poziciju novog elementa:
4 Unesite vrednost novog elementa:
5 Izgled niza nakon umetanja:
6 1 3 25 7 8 9

Zadatak 2.40

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji u niz A_N , čiji su elementi uređeni u neopadajući redosled, umeće 10 novih elemenata, tako da nakon umetanja svakog novog elementa niz zadrži uređenje. Prikazati niz nakon umetanja vrednosti.

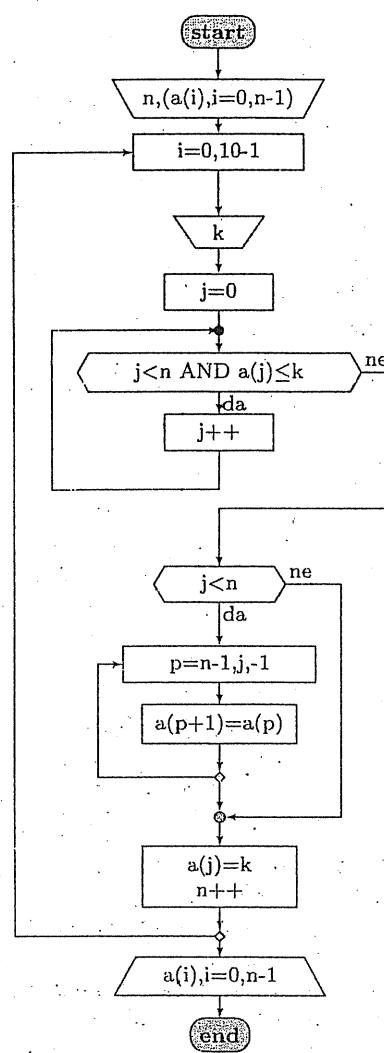
Rešenje

```

1 #include <stdio.h>
2 void main()
3 {
4     int n,i,p,j,k,a[20];
5     scanf ("%d",&n);
6     for ( i = 0; i < n; i++ )
7         scanf ("%d",&a[i]);
8     for ( i = 0; i < 10; i++ )
9     {
10         scanf ("%d",&k);
11         j=0;
12         while ( j < n && a[j] <= k )
13             j++;

```

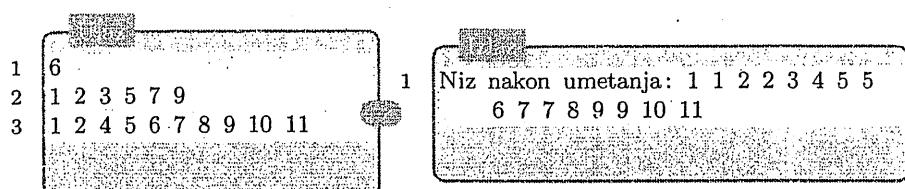
Zbirka zadataka



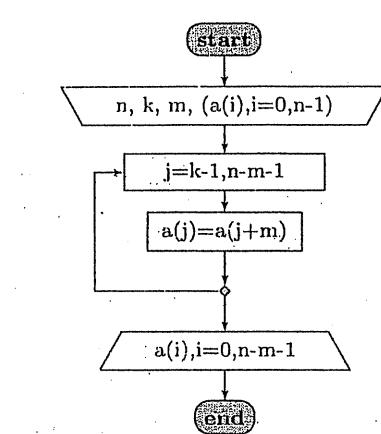
```

14 if ( j < n )
15     for ( p = n-1; p >= j; p-- )
16         a[p+1] = a[p];
17     a[j] = k;
18     n++;
19 }
20 printf ("Niz nakon umetanja: ");
21 for ( i = 0; i < n; i++ )
22     printf ("%d ", a[i]);
23 }
  
```

Zbirka zadataka

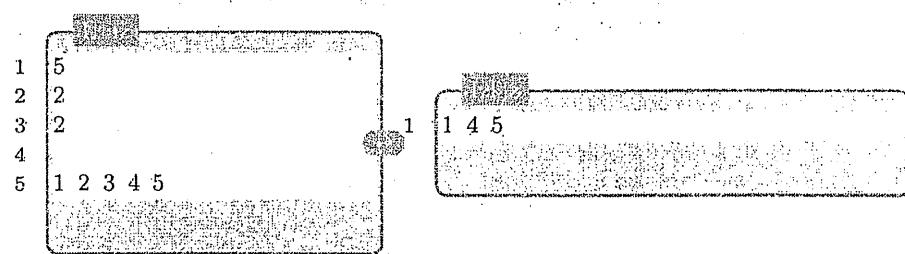
**Zadatak 2.41**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji iz niza od n elemenata briše m elemenata, počev od elementa sa indeksom $k - 1$. Broj elemenata niza, vrednosti elemenata, i parametre k i m zadaje korisnik. Predvideti pomeranje elemenata kako bi se popunio novonastali prazan prostor. Prikazati niz nakon brisanja elemenata.

Rešenje

```

1 #include <stdio.h>
2
3 void main ()
4 {
5     int a[20], m, n, k, i, j;
6     scanf("%d%d%d", &n, &k, &m);
7     for (i=0; i<n; i++)
8         scanf("%d", &a[i]);
9     for (j=k-1; j<=n-m-1; j++)
10        a[j]=a[j+m];
11     for (i=0; i<n-m; i++)
12         printf("%d ", a[i]);
13 }
  
```

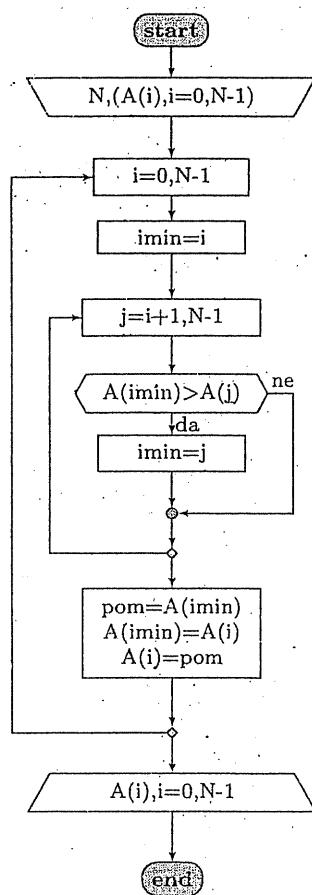
**Zadatak 2.42**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji niz A_N uređuje u rastući redosled metodom selekcije (*selection sort*)

Zbirka zadataka

sort). Broj elemenata niza i vrednosti elemenata zadaje korisnik. Prikazati niz nakon sortiranja.

Rešenje



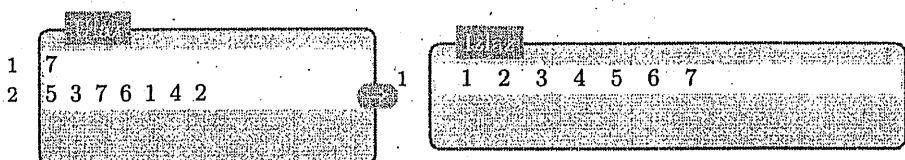
```

1 #include "stdio.h"
2 void main()
3 {
4     int N, A[50], i, j, imin, pom;
5     scanf("%d",&N);
6     for(i=0; i<N; i++)
7         scanf("%d",&A[i]);
8     for (i=0; i<N; i++)
9     {
10         imin = i;
11         for (j=i+1; j<N; j++)
12             if (A[imin] > A[j])
  
```

```

13         imin = j;
14         pom = A[imin];
15         A[imin] = A[i];
16         A[i] = pom;
17     }
18     for(i=0; i<N; i++)
19         printf("%3d",A[i]);
20 }

```



Zadatak 2.43

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji niz A_N uređuje u rastući redosled metodom umetanja (*insertion sort*). Broj elemenata niza i vrednosti elemenata zadaje korisnik. Prikazati niz nakon sortiranja.

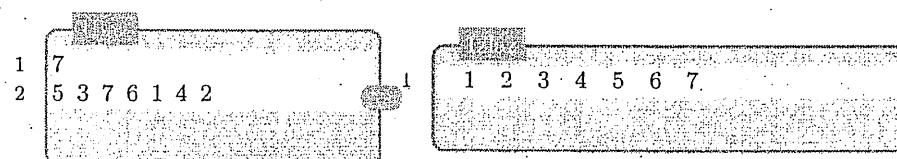
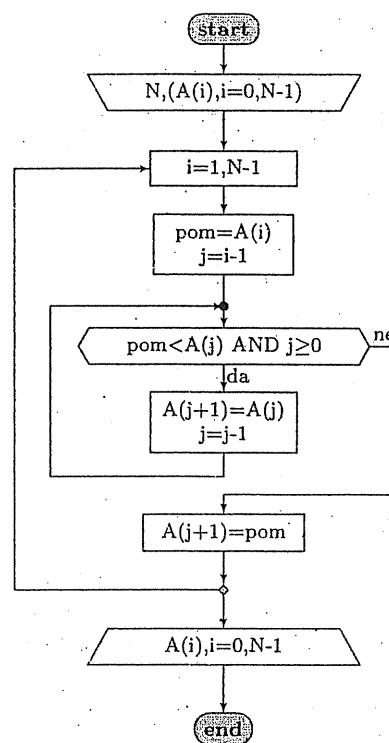
Rešenje

```

1 #include "stdio.h"
2 void main()
3 {
4     int N, A[50], i, j, pom;
5     scanf("%d",&N);
6     for(i=0; i<N; i++)
7         scanf("%d",&A[i]);
8     for (i=1; i<N; i++)
9     {
10         pom = A[i];
11         j = i - 1;
12         while (pom < A[j] && j >= 0)
13         {
14             A[j+1] = A[j];
15             j = j - 1;
16         }
17         A[j+1] = pom;
18     }
19     for(i=0; i<N; i++)
20         printf("%3d",A[i]);
21 }

```

Zbirka zadataka



Zadatak 2.44

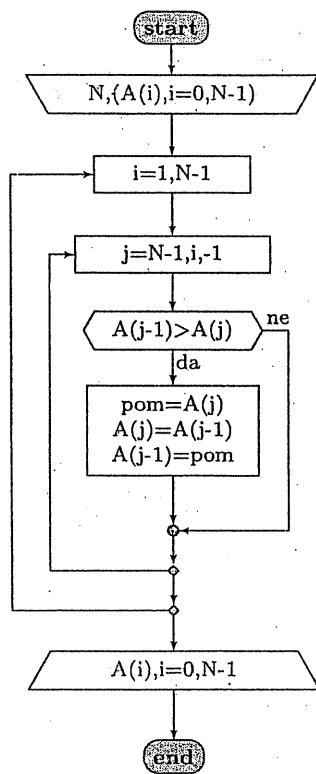
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji niz A_N uređuje u rastući redosled metodom zamene mesta. (*bubble sort*). Broj elemenata niza i vrednosti elemenata zadaje korisnik. Prikazati niz nakon sortiranja.

Rešenje

```

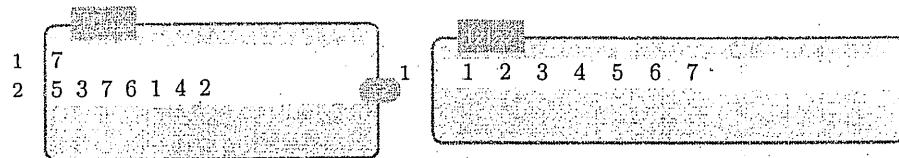
1 #include "stdio.h"
2 void main()
3 {
4     int N, A[50], i, j, imin, pom;
    
```

Zbirka zadataka



```

5     scanf("%d",&N);
6     for(i=0; i<N; i++)
7         scanf("%d",&A[i]);
8     for (i=1; i<N; i++)
9         for (j=N-1; j>=i; j--)
10            if (A[j-1] > A[j])
11            {
12                pom = A[j];
13                A[j] = A[j-1];
14                A[j-1] = pom;
15            }
16     for(i=0; i<N; i++)
17         printf("%d",A[i]);
18 }
  
```



Zadatak 2.45

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji niz od N elemenata uređuje u opadajući redosled. Broj elemenata niza i vrednosti elemenata zadaje korisnik. Podrazumevati da pri unosu nije bilo ponavljanja elemenata. Zatim učitati broj K ($K < N$) i K brojeva redom. Posle svakog učitanog broja, ukloniti iz niza element koji je jednak učitanom broju, a ostale elemente pomeriti u levo kako bi se popunila nastala praznina. Ukoliko se broj ne nalazi u nizu, niz ostaje nepromenjen. Prikazivati niz nakon sortiranja i nakon svakog izbacivanja.

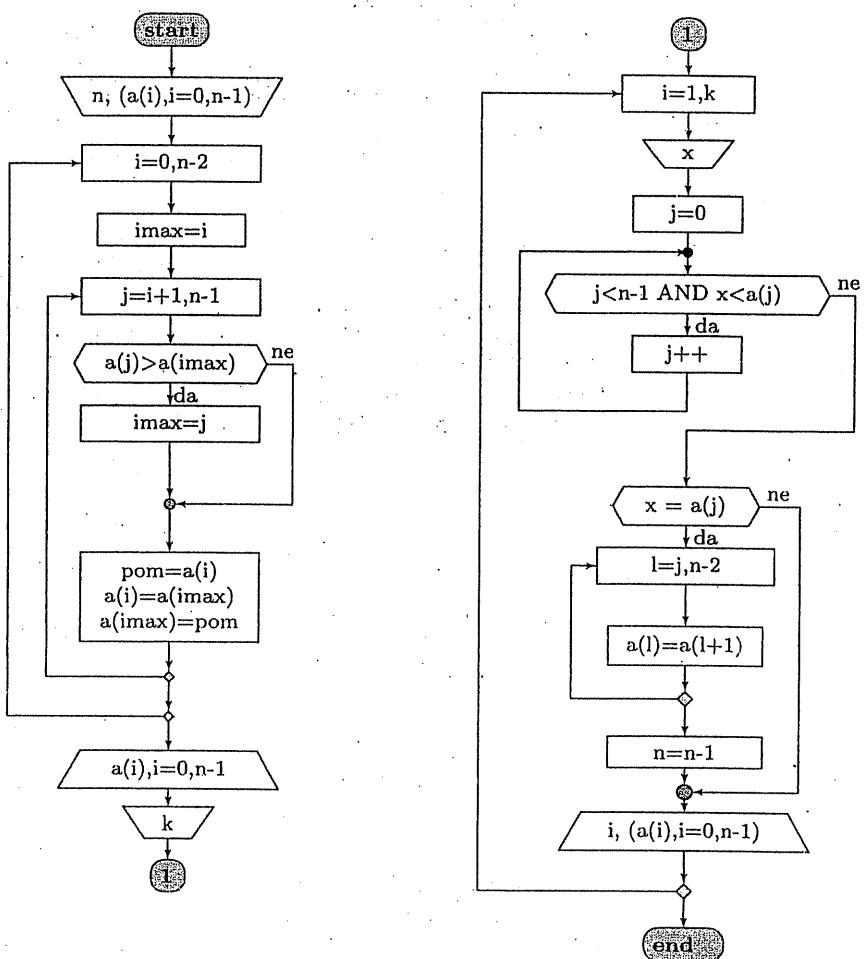
Rešenje

```

1 #include <stdio.h>
2 void main()
3 {
4     int n,a[50],x,i,j,l,k,imax,pom;
5     scanf("%d",&n);
6     for (i=0; i<n; i++)
7         scanf("%d",&a[i]);
8     for (i=0; i<n-1; i++)
9     {
10         imax=i;
11         for (j=i+1; j<n; j++)
12             if ( a[j] > a[imax] )
13                 imax=j;
14         pom=a[i];
15         a[i]=a[imax];
16         a[imax]=pom;
17     }
18     printf("Niz_posle_sortiranja\n");
19     for (i=0;i<n; i++)
20         printf("%d ",a[i]);
21     printf("\n");
22     scanf("%d",&k);
23     for (i=1; i<=k; i++)
24     {
25         scanf("%d",&x);
26         j=0;
27         while ( j < n-1 && x < a[j] )
28             j++;
29         if ( x == a[j] )

```

Zbirka zadataka



```

30
31         {
32             for (l=j; l<n-1; l++)
33                 a[l]=a[l+1];
34             n--;
35         }
36         printf("Niz_posle %d. ubrisanja\n", i);
37         for (l=0; l<n; l++)
38             printf("%d ", a[l]);
39         printf("\n");
40     }

```

Zbirka zadataka

```

1 7
2 9 8 3 1 4 5 6
3 3
4 6 4 1

```

```

1 Niz posle sortiranja
2 9 8 6 5 4 3 1
3 Niz posle 1. brisanja
4 9 8 5 4 3 1
5 Niz posle 2. brisanja
6 9 8 5 3 1
7 Niz posle 3. brisanja
8 9 8 5 3

```

Zadatak 2.46

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati program kojim se u celobrojnom nizu A od N elemenata pronađe svi parovi elemenata čiji je zbir jednak zadatom broju M . Broj elemenata niza, vrednost elemenata i parametar M zadaje korisnik. Na ekranu prikazati parove indeksa elemenata koji zadovoljavaju uslov.

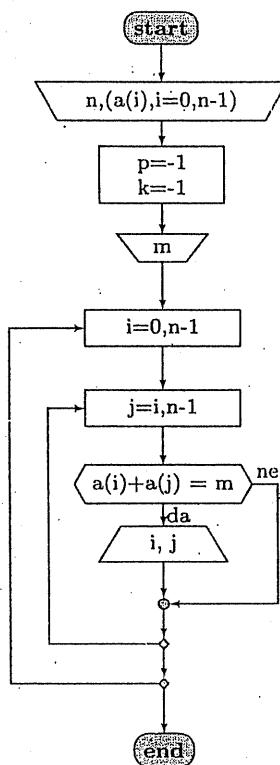
Rešenje

```

1 #include <stdio.h>
2 void main()
3 {
4     int n,m,i,j,p,k,a[30];
5     scanf ("%d",&n);
6     p = -1;
7     k = -1;
8     for ( i = 0; i < n; i++ )
9         scanf ("%d",&a[i]);
10    scanf ("%d",&m);
11    for ( i = 0; i < n; i++ )
12    {
13        for ( j = i; j < n; j++ )
14            if ( a[i] + a[j] == m )
15                printf ("%d,%d",i,j);
16    }
17 }

```

Zbirka zadataka



1 10
2 1 2 3 4 5 1 2 3 4 5
3 5

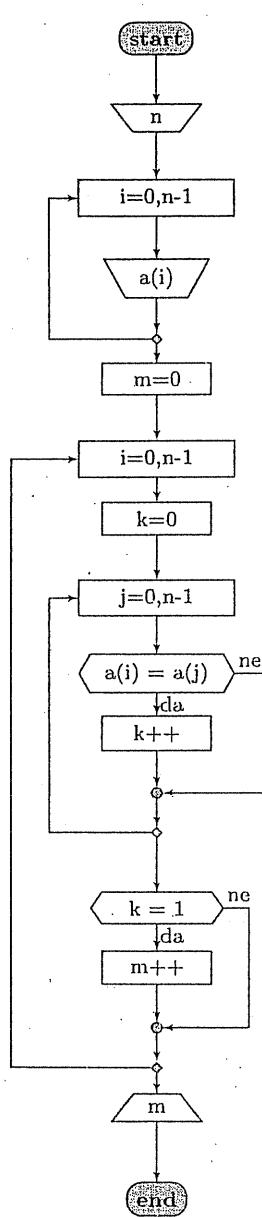
1 (0,3)(0,8)(1,2)(1,7)
2 (2,6)(3,5)(5,8)(6,7)

Zadatak 2.47

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje elemente niza koji se javljaju samo jednom. Broj elemenata niza i elemente niza zadaje korisnik.

Rešenje

Zbirka zadataka



```

1 #include <stdio.h>
2 void main()
3 {
4     int i,j,k,a[50],n,m;
5     scanf ("%d",&n);
6     for(i=0;i<n; i++)
7         scanf ("%d",&a[i]);
8     m=0;
9     for(i=0;i<n; i++)
10    {
11         k=0;
12         for(j=0;j<n; j++)
13             if( a[i] == a[j] )
14                 k++;
15         if ( k == 1 )
16             m++;
17     }
18     printf ("Broj elemenata u nizu koji se
19     javljaju samo jednom je %d",m);
  
```

1 8
2 5 1 3 1 4 2 2 7

1 Broj elemenata niza koji se javljaju samo jednom je 4

Zadatak 2.48

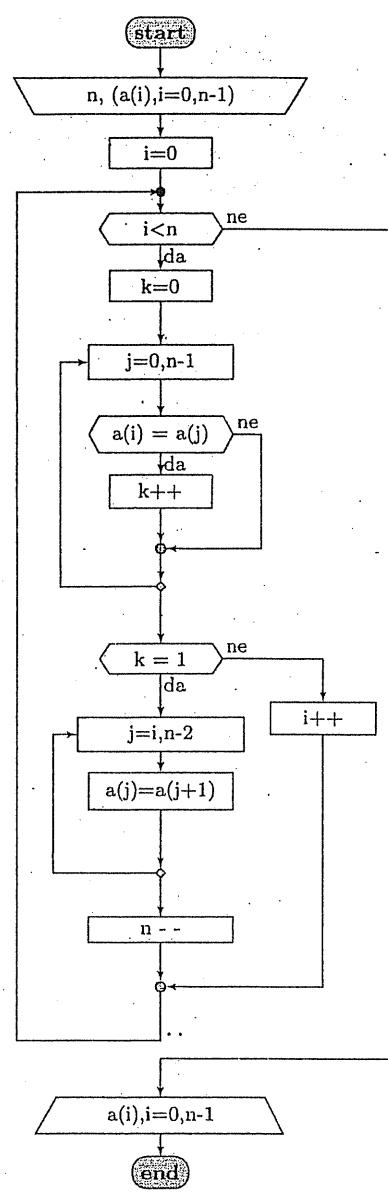
Nacrtati struktturni dijagram toka algoritma i na programskom jeziku C napisati struktturni program koji određuje i briše elemente niza koji se javljaju samo jednom. Nakon brisanja svakog elementa obezbediti da se novonastali "prazan prostor" popuni pomeranjem odgovarajućih elemenata niza. Prikazati niz nakon transformacije.

Rešenje

```

1 #include <stdio.h>
2 void main()
3 {
4     int i, j, k, a[50], n;
5     printf("Unesite broj elemenata niza:\n");
6     scanf("%d", &n);
7     printf("Unesite elemente niza:\n");
8     for(i=0; i<n; i++)
9         scanf("%d", &a[i]);
10    i=0;
11    while(i<n)
12    {
13        k=0;
14        for(j=0; j<n; j++)
15            if(a[i] == a[j])
16                k++;
17        if(k==1)
18        {
19            for(j=i; j<n-1; j++)
20                a[j] = a[j+1];
21            n--;
22        }
23        else
24            i++;
25    }
26    printf("Niz nakon izbacivanja elemenata:");
27    for(i=0; i<n; i++)
28    {
29        printf("%d", a[i]);
30    }
31 }
```

Zbirka zadataka



1 10
2 1 2 1 5 3 8 3 7 4 1

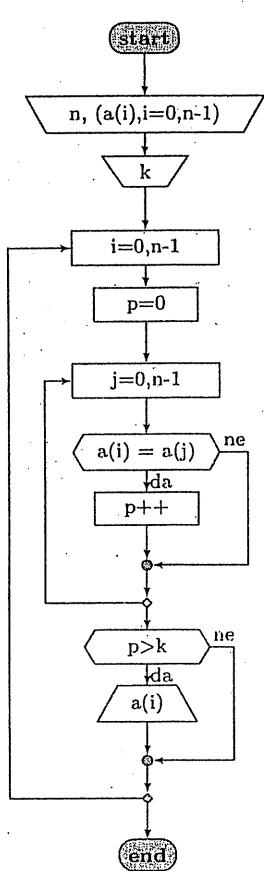
1 Unesite broj elemenata niza:
2 Unesite elemente niza:
3 Niz nakon izbacivanja elemenata: 1 1
3 3 1

Zbirka zadataka

Zadatak 2.49

Nacrtati strukturni dijagram toka algoritma i na programsку jeziku C napisati program koji u celobrojnom nizu A_N pronalazi i prikazuje sve elemente koji se javljaju više od K puta. Broj elemenata niza, elemente niza i parametar K zadaje korisnik.

Rešenje



```

1 #include <stdio.h>
2 void main()
3 {
4     int i,n,k,j,p,a[30];
5     scanf ("%d",&n);
6     for ( i = 0; i < n; i++ )
7         scanf ("%d",&a[i]);
8     scanf ("%d",&k);
9     printf ("Elementi niza koji se javljaju vise od %d puta su:\n",k);
10    for ( i = 0; i < n; i++ )
11    {
12        p = 0;
13        for ( j = 0; j < n; j++ )
14        {
15            if ( a[i] == a[j] )
16                p++;
17        }
18        if ( p > k )
19            printf ("%d", a[i]);
20    }
21 }
  
```

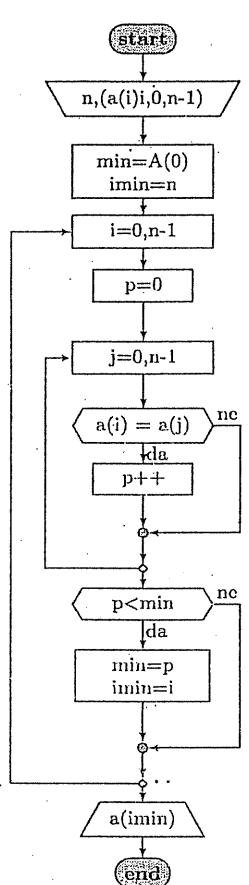
1 10
2 1 1 2 3 4 5 5 6 7 8
3 1

1 Elementi niza koji se javljaju vise od 1 puta su:
2 1 1 5 5

Zadatak 2.50

Nacrtati strukturni dijagram toku algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje element niza koji se javlja najmanji broj puta. Smatratи da u nizu postoji samo jedan takav element.

Rešenje



```

1 #include <stdio.h>
2 void main()
3 {
4     int n, i, p, j, min, imin, a[30];
5     scanf ("%d", &n);
6     for ( i = 0; i < n; i++ )
7         scanf ("%d", &a[i]);
8     min = A[0];
9     imin = n;
10    for ( i = 0; i < n; i++ )
11    {
12        p=0;
13        for ( j = 0; j < n; j++ )
14            if ( a[i] == a[j] ) p++;
15        if ( p < min )
16        {
17            min = p;
18            imin = i;
19        }
20    }
21    printf ("Element koji se javlja najmanji
22    broj puta: %d", a[imin]);
23 }
  
```

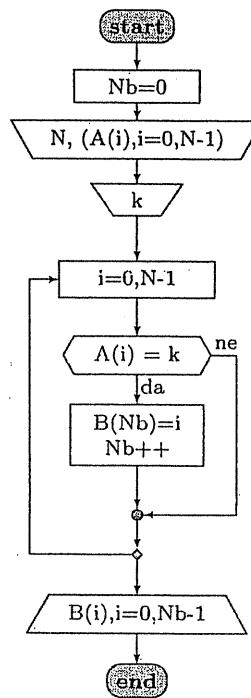
1 10
2 3 2 2 3 4 4 5 6 6 4

1 Element koji se javlja najmanji broj puta: 5

Zadatak 2.51

Nacrtati struktturni dijagram toka algoritma i na programskom jeziku C napisati struktturni program koji određuje pozicije (indeks) na kojima se zadata vrednost k javlja u nizu A od N elemenata. Pozicije na kojima se zadata vrednost javlja upisati u pomoćnini niz B . Prikazati elemente niza B nakon formiranja.

Rešenje



```

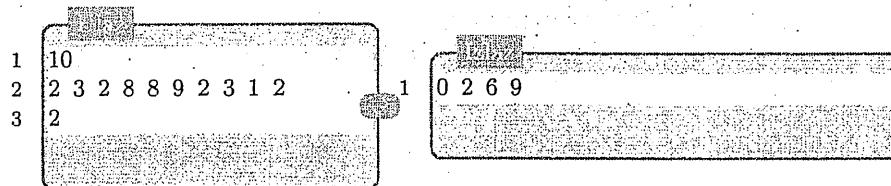
1 #include "stdio.h"
2 void main()
3 {
4     int N, A[100], B[100], k, Nb=0, i;
5     scanf("%d", &N);
6     for (i = 0; i < N; i++)
7         scanf("%d", &A[i]);
8     scanf("%d", &k);
9     for (i = 0; i < N; i++)
10        if (A[i]==k)
11        {
12            B[Nb] = i;
13            Nb++;
14        }
15     for (i = 0; i < Nb; i++)
  
```

Zbirka zadataka

```

16         printf("%d",B[i]);
17 }

```

**Zadatak 2.52**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje da li je niz B_M podniz niza A_N . Broj elemenata nizova i vrednosti clemenata zadaje korisnik. Smatrati da važi $M \leq N$. Ukoliko niz B jeste podniz niza A , prikazati indeks u nizu A od koga počinje niz B .

Rešenje

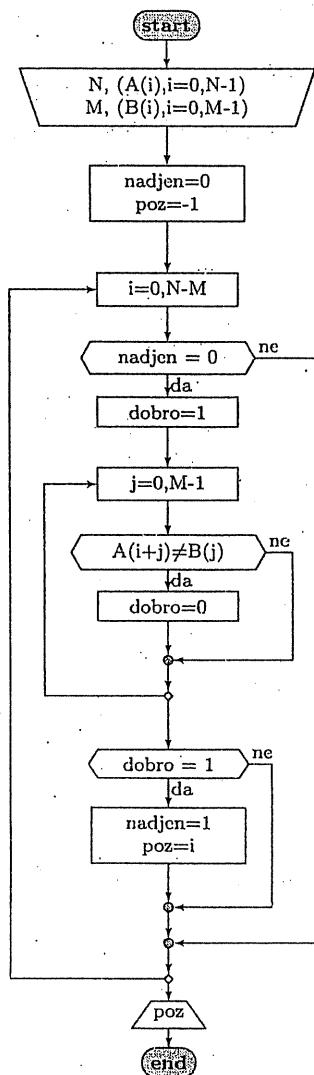
prvi način, bez optimizacije izvršenja

```

1 #include "stdio.h"
2 void main()
3 {
4     int A[100],B[100], N, M;
5     int i, j, nadjen, dobro, poz;
6
7     scanf("%d",&N);
8     for (i = 0; i < N; i++)
9         scanf("%d",&A[i]);
10
11    scanf("%d",&M);
12    for (i = 0; i < M; i++)
13        scanf("%d",&B[i]);
14
15    nadjen=0;
16    poz=-1;
17    for (i=0; i<=N-M; i++)
18    {
19        if(nadjen==0)
20        {
21            dobro=1;
22            for(j=0;j<M; j++)
23                if(A[i+j]!=B[j])
24                    dobro=0;
25        if(dobro==1)
26        {
27            nadjen=1;

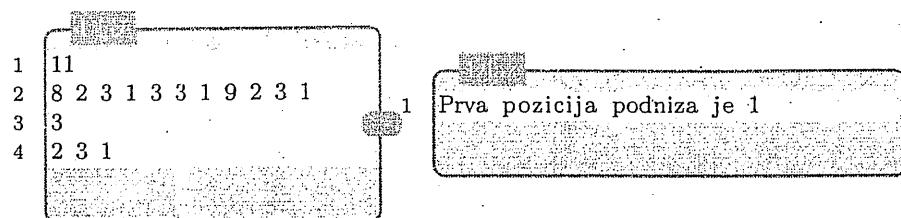
```

Zbirka zadataka

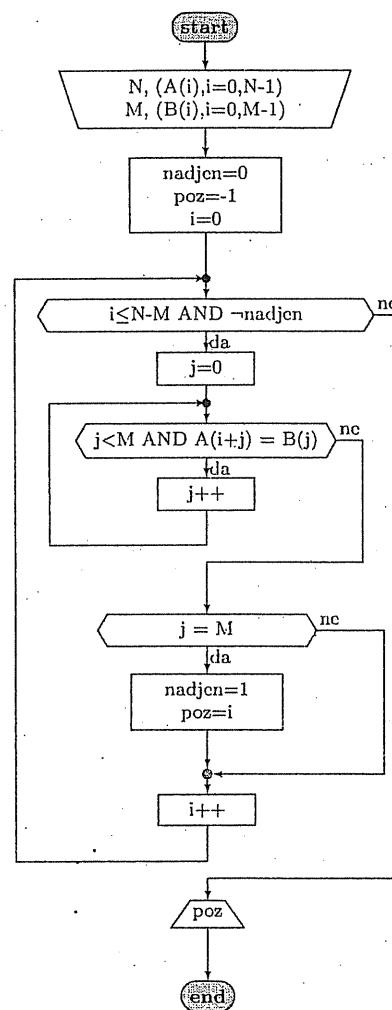


```

28         poz=i;
29     }
30 }
31 }
32 printf("Prva_pozicija_podniza_je %d\n", poz);
33 }
  
```



drugi način, sa optimizacijom izvršenja



Zbirka zadataka

```

1 #include "stdio.h"
2 void main()
3 {
4     int A[100],B[100], N, M;
5     int i, j, nadjen, poz;
6
7     scanf("%d",&N);
8     for (i = 0; i < N; i++)
9         scanf("%d",&A[i]);
10
11    scanf("%d",&M);
12    for (i = 0; i < M; i++)
13        scanf("%d",&B[i]);
14
15    nadjen=0;
16    poz=-1; //signal da nema pojavljivanja
17    i=0;
18    while(i<=N&& !nadjen) //od pocetka do poslednjeg indeksa gde
19    bi mogao da pocne podniz ALI samo dok "nadjen" ne postane 1
20    {
21        j=0;
22        while(j<M&& A[i+j]==B[j])
23            j++;
24
25        if(j==M)
26        {
27            nadjen=1;
28            poz=i;
29        }
30        i++;
31    }
32    printf("Prva pozicija podniza je %d\n",poz);
33 }

```

1 11
2 5 3 3 1 3 3 1 9 2 3 1
3 3
4 2 3 1

1 Prva pozicija podniza je 8

Zadatak 2.53

Nacrtati strukturni dijagram toka algoritma i napisati strukturni program na jeziku C koji smic̄ta novog posetioca do svog supružnika u redu sedišta na koncertu. Posmatra se jedan red sedišta u sali predstavljen nizom celih brojeva. Sa tastature se unosi broj

Zbirka zadataka

clemenata niza a zatim i elementi niza redom, pri čemu vrednosti elemenata odgovaraju osobama koje su sele na odgovarajuće sedište: 0 se unosi ako je sedište ostalo slobodno, ceo broj različit od nule ako je sedište zauzeto. Svaka vrednost predstavlja visinu u centimetrima osobe koja sedi na sedištu. Sa tastature se unosi visina novog posetioca. Poznato je da je njegov supružnik visine koja je najpribližnija visini novog posetioca. Treba smestiti novog posetioca na sedište pored supružnika. Ukoliko pored supružnika koji sedi nema slobodnog sedišta, pomeriti oba supružnika na prvi slobodan par susednih sedišta. Prikazati clemenate niza sedišta pre i nakon smještanja novog posetioca.

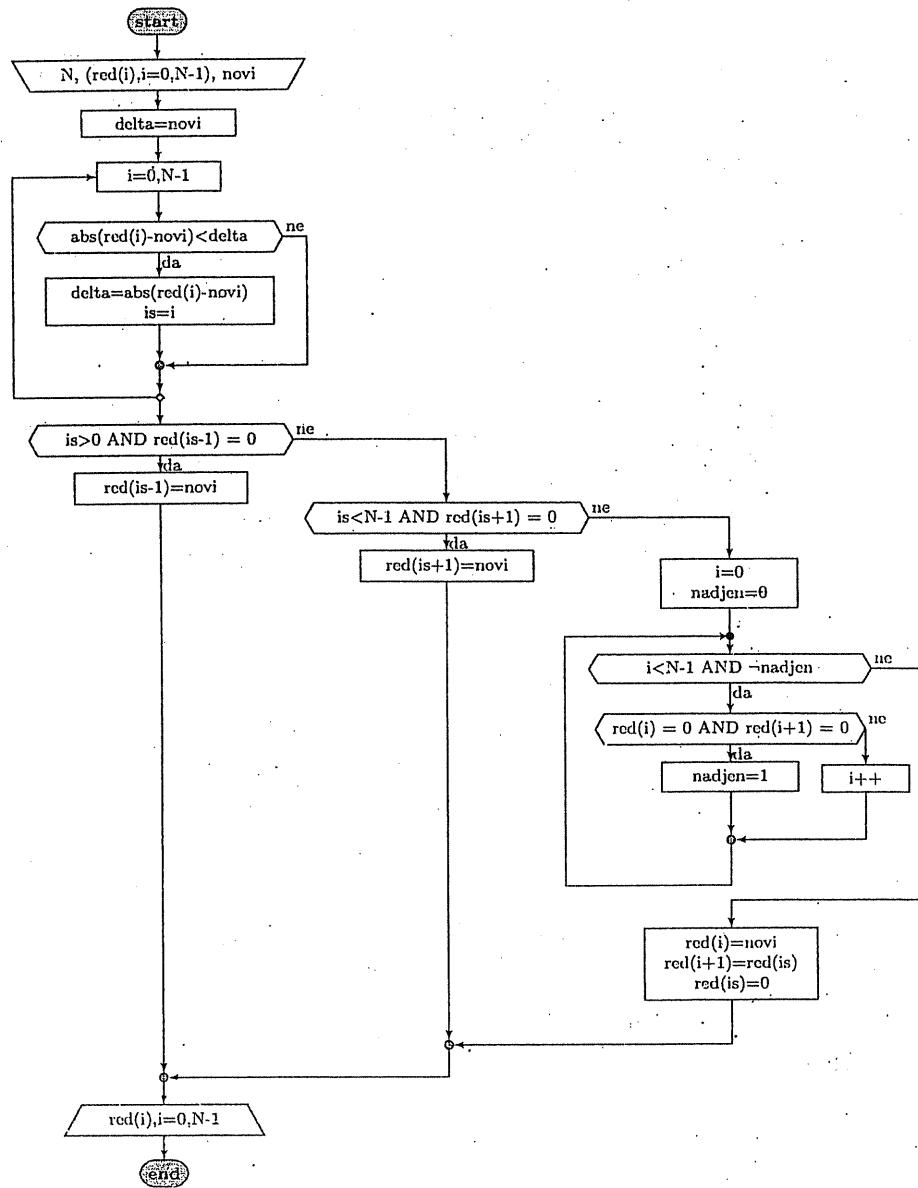
Rešenje

```

1 #include <stdio.h>
2 #include <math.h>
3 void main()
4 {
5     int N;
6     int red[50];
7     int novi;
8     int delta, is;
9     int i, nadjen;
10    scanf("%d",&N);
11    for (i=0; i<N; i++)
12        scanf("%d",&red[i]);
13    scanf("%d", &novi);
14    delta=novi;
15    for (i=0; i<N; i++)
16        if (abs(red[i]-novi)<delta)
17        {
18            delta=abs(red[i]-novi);
19            is=i;
20        }
21    if(is>0 && red[is-1]==0)
22        red[is-1]=novi;
23    else if (is<N-1 && red[is+1]==0)
24        red[is+1]=novi;
25    else
26    {
27        i=0;
28        nadjen = 0;
29        while (i<N-1 && !nadjen)
30        {
31            if (red[ i]==0 && red[ i+1]==0)
32                nadjen=1;
33            else
34                i++;
35        }
36        red[ i]=novi;
37        red[ i+1]=red[ is];

```

Zbirka zadataka



```

38     red[is]=0;
39 }
40 printf("\n");
41 printf("Stanje_sedista_nakon_smestanja_novog_posetioca:\n");
42 for (i=0; i<N; i++)
    
```

Zbirka zadataka

```

43         printf("%d\t",red[i]);
44     }

```

1 6
2 180 180 0 130 140 170
3 120

1
2 Stanje sedista nakon smestanja novog posetioca:
3 180 180 120 130
140 170

Operacije nad skupovima predstavljenim nizovima

Zadatak 2.54

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje:

- a) presek skupova $C = A \cap B$
- b) razliku skupova $C = A \setminus B$

Smatrati da su skupovi zadati nizovima A_N i B_M , čije elemente zadaje korisnik. Prikazati rezultujući skup.

Rešenje

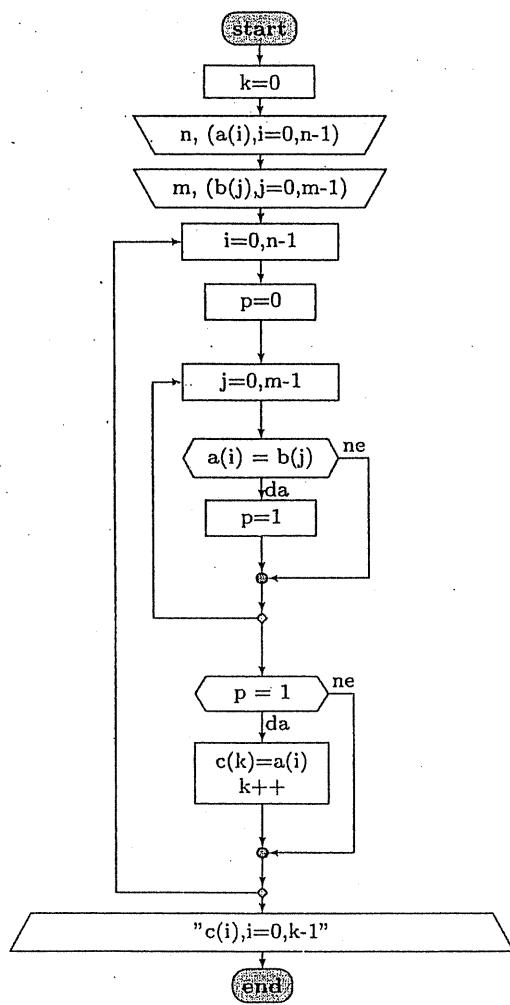
a)

```

1 #include <stdio.h>
2 void main()
3 {
4     int i,j,a[20],b[20],c[20],n,m,k=0,p;
5     printf("Unesi_broj_elemlenata_skupa_A_i_elemlente_skupa_A\n");
6     scanf ("%d",&n);
7     for(i=0;i<n;i++)
8         scanf ("%d",&a[i]);
9     printf("Unesi_broj_elemlenata_skupa_B_i_elemlente_skupa_B\n");
10    scanf ("%d",&m);
11    for(j=0;j<m;j++)
12        scanf ("%d",&b[j]);
13    for(i=0;i<n;i++)
14    {
15        p=0;
16        for(j=0;j<m;j++)
17            if( a[i] == b[j] )
18                p=1;
19        if( p == 1 )
20        {
21            c[k]=a[i];

```

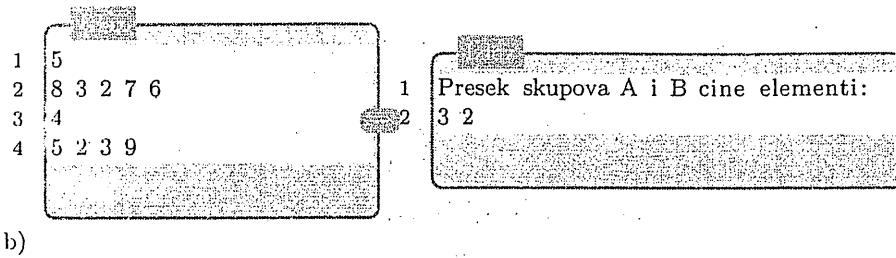
Zbirka zadataka



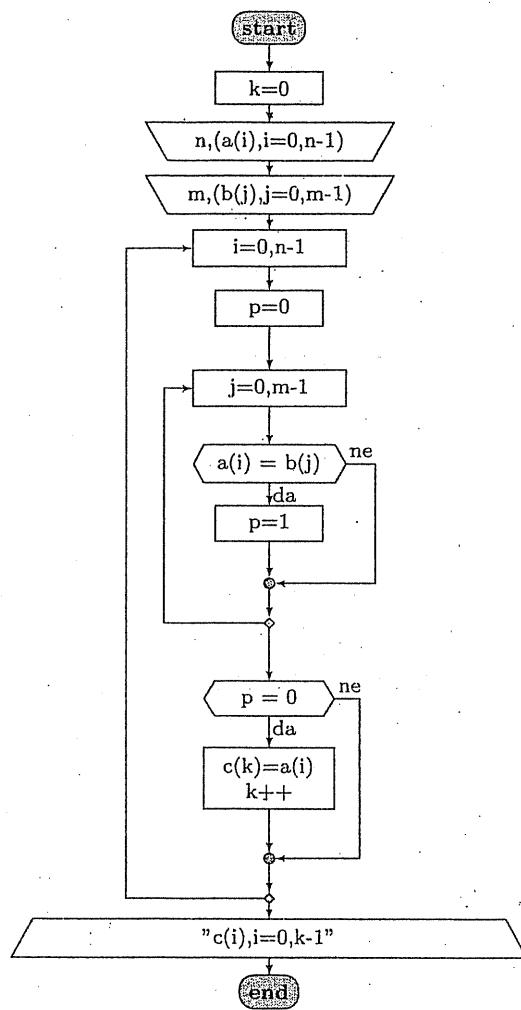
```

22           k++;
23       }
24   }
25   printf("Presek skupova A i B cine elementi:\n");
26   for(i=0;i<k; i++)
27     printf ("%d ", c[i]);
28 }
  
```

Zbirka zadataka



b)



```

1 #include <stdio.h>
2 void main()
3 {

```

Zbirka zadataka

```

4     int i,j,a[20],b[20],c[20],n,m,k=0,p;
5     printf("Unesi_broj_elemenata_skupa_A_i_elemente_skupa_A\n");
6     scanf ("%d",&n);
7     for(i=0;i<n; i++)
8         scanf ("%d",&a[i]);
9     printf("Unesi_broj_elemenata_skupa_B_i_elemente_skupa_B\n");
10    scanf ("%d",&m);
11    for(j=0;j<m; j++)
12        scanf ("%d",&b[j]);
13    for(i=0;i<n; i++)
14    {
15        p=0;
16        for(j=0;j<m; j++)
17            if( a[i] == b[j] )
18                p=1;
19        if( p == 0 )
20        {
21            c[k]=a[i];
22            k++;
23        }
24    }
25    printf("Razliku_skupova_A_i_B_cine_elementi:\n");
26    for(i=0;i<k; i++)
27        printf ("%d ",c[i]);
28 }

```

1 5
2 8 3 2 7 6
3 4
4 5 2 3 9

1 Unesi broj elemenata skupa A i
elemente skupa A
2 Unesi broj elemenata skupa B i
elemente skupa B
3 Presek skupova A i B cine elementi:
4 8 7 6

Zadatak 2.55

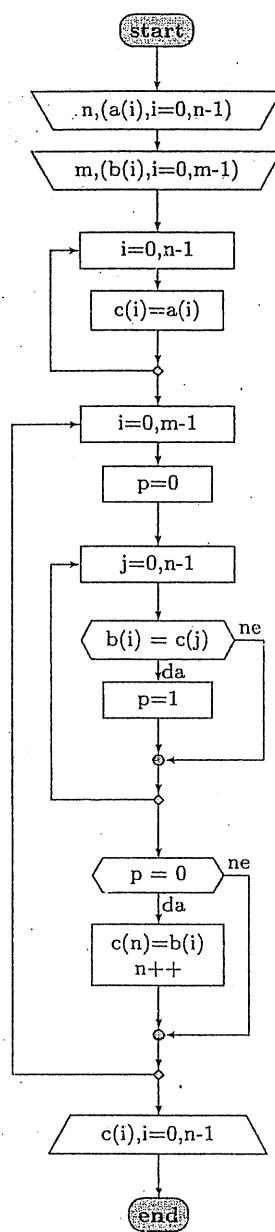
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje uniju skupova

$$C = A \cup B.$$

Smatrati da su skupovi zadati nizovima A_N i B_M , čije elemente zadaje korisnik. Prikazati rezultujući skup.

Rešenje

Zbirka zadataka



```

1 #include<stdio.h>
2 void main()
3 {
4     int i, j, n, m, p, a[20], b[20], c[40];
  
```

Zbirka zadataka

```

5      printf("Unesite_broj_elemlenata_skupa_A:\n");
6      scanf("%d", &n);
7      printf("Unesite_elemlente_skupa_A:\n");
8      for(i=0; i<n; i++)
9          scanf("%d", &a[i]);
10     printf("Unesite_broj_elemlenata_skupa_B:\n");
11     scanf("%d", &m);
12     printf("Unesite_elemlente_skupa_B:\n");
13     for(i=0; i<m; i++)
14         scanf("%d", &b[i]);
15     for(i=0; i<n; i++)
16         c[i] = a[i];
17     for (i=0; i<m; i++)
18     {
19         p = 0;
20         for(j=0; j<n; j++)
21             if(b[i] == c[j])
22                 p = 1;
23         if(p==0)
24         {
25             c[n] = b[i];
26             n++;
27         }
28     }
29     printf("Skup_C=A.U.B:");
30     for(i=0; i<n; i++)
31         printf("%d", c[i]);
32 }
```

1 7
2 2 4 1 6 9 7
3 5
4 3 1 5 6 4

1 Unesite broj elemlenata skupa A:
2 Unesite elemlente skupa A:
3 Unesite broj elemlenata skupa B:
4 Unesite elemlente skupa B:
5 Skup C = A U B: 2 4 1 6 9 7 5

Zadatak 2.56

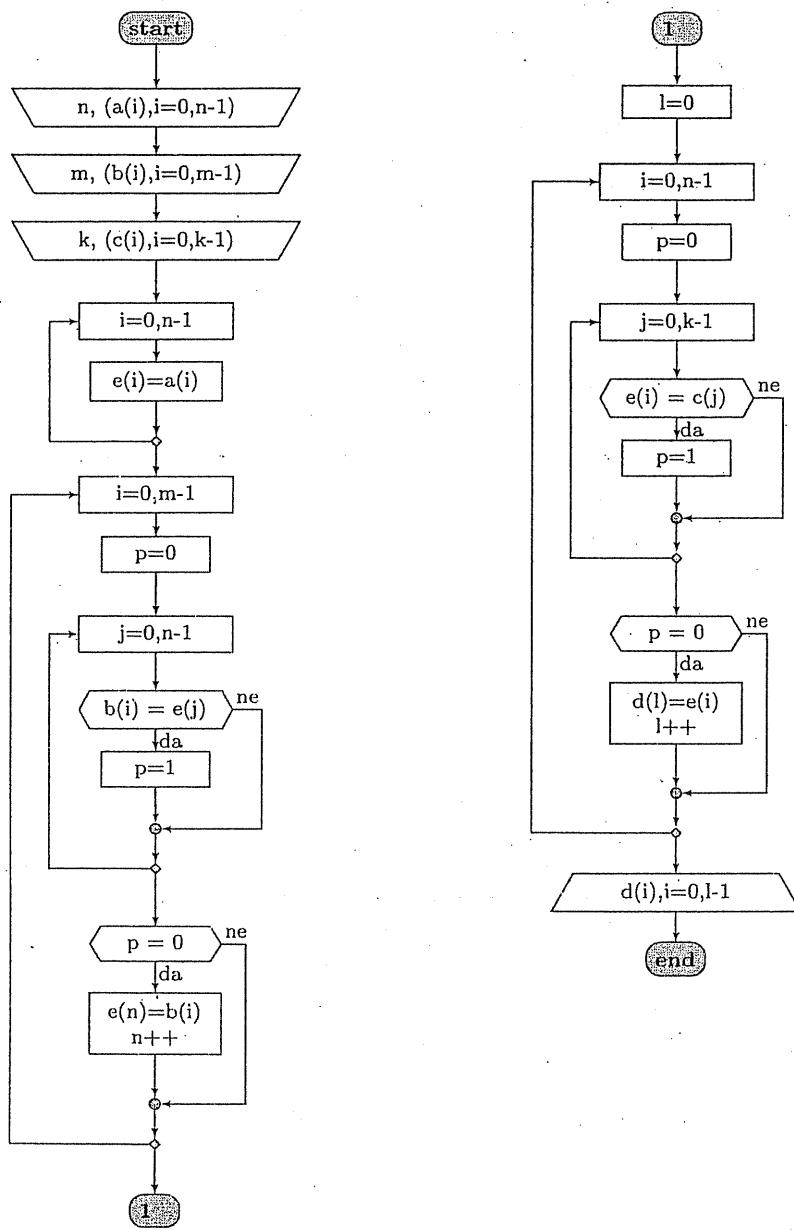
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje skup

$$D = (A \cup B) \setminus C.$$

Smatrati da su skupovi zadati nizovima A_N , B_M i C_K , čije elemente zadaje korisnik. Prikazati rezultujući skup.

Zbirka zadataka

Rešenje



```

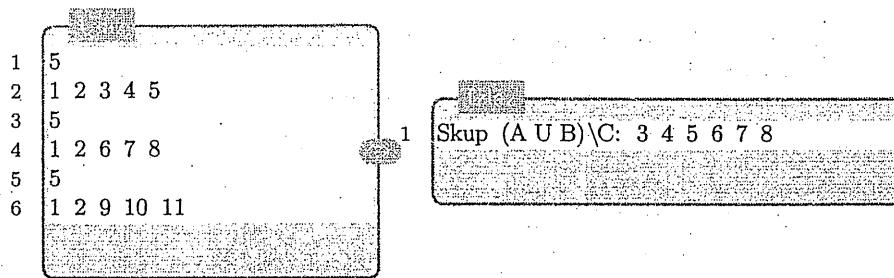
1 #include <stdio.h>
2 void main()
3 {

```

Zbirka zadataka

```

4      int i,j,n,l,m,k,a[20],b[20],c[40],d[20],e[20],p;
5      scanf ("%d",&n);
6      for ( i = 0; i < n; i++ )
7          scanf ("%d",&a[i]);
8      scanf ("%d",&m);
9      for ( i = 0; i < m; i++ )
10         scanf ("%d",&b[i]);
11     scanf ("%d",&k);
12     for ( i = 0; i < k; i++ )
13         scanf ("%d",&c[i]);
14     for ( i = 0; i < n; i++ )
15         e[i] = a[i];
16     for ( i = 0; i < m; i++ )
17     {
18         p = 0;
19         for (j = 0; j < n; j++)
20             if ( b[i] == e[j] )
21                 p = 1;
22         if ( p == 0)
23         {
24             e[n] = b[i];
25             n++;
26         }
27     }
28     l = 0;
29     for ( i = 0; i < n; i++ )
30     {
31         p = 0;
32         for (j = 0; j < k; j++)
33             if ( e[i] == c[j] )
34                 p = 1;
35         if ( p == 0)
36         {
37             d[l] = e[i];
38             l++;
39         }
40     }
41     printf ("Skup-(A.U.B)\\C:\n");
42     for ( i = 0; i < l; i++ )
43         printf ("%d",d[i]);
44 }
```

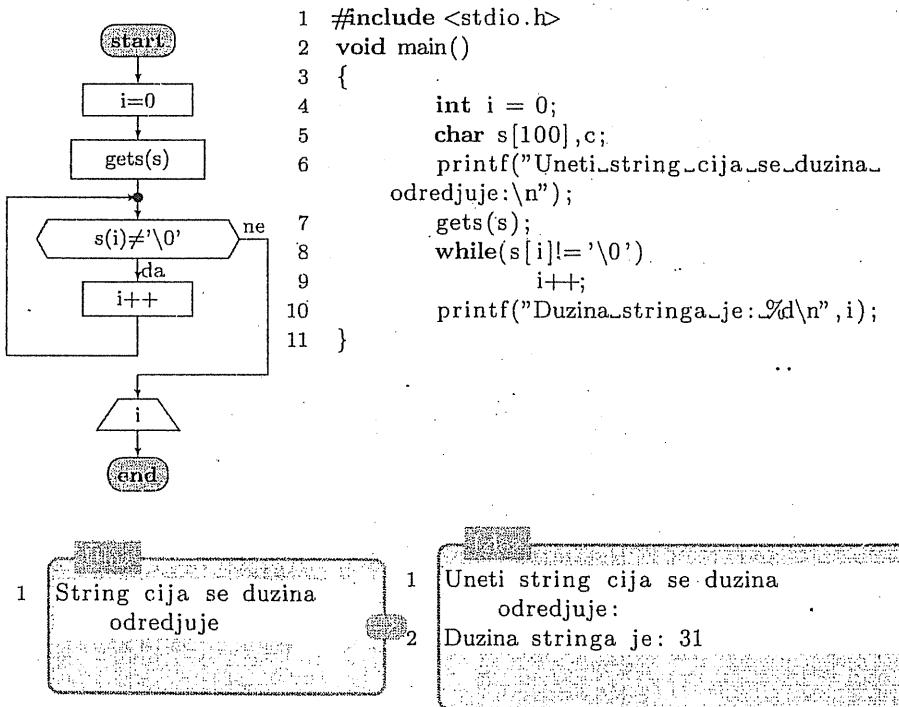


2.1.2 Znakovni nizovi - stringovi

Zadatak 2.57

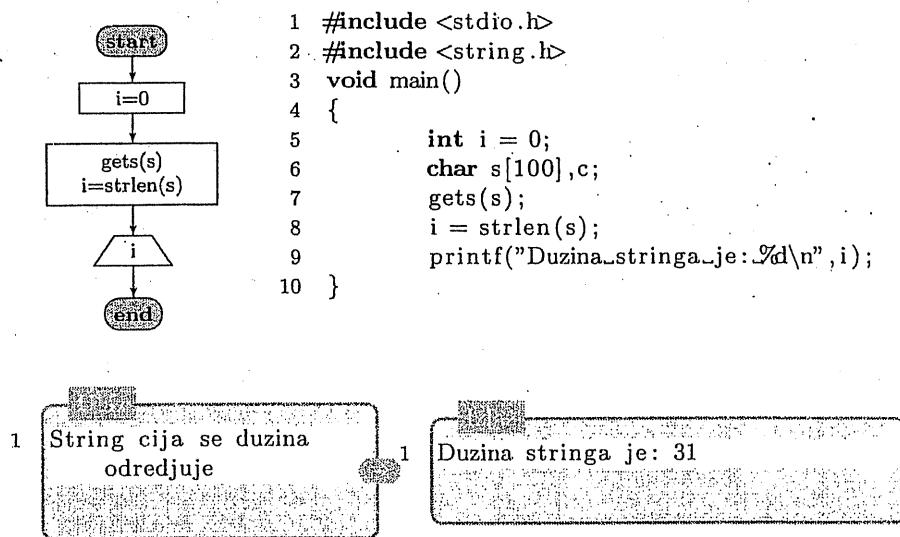
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje dužinu unetog stringa bez korišćenja bibliotečkih funkcija.

Rešenje



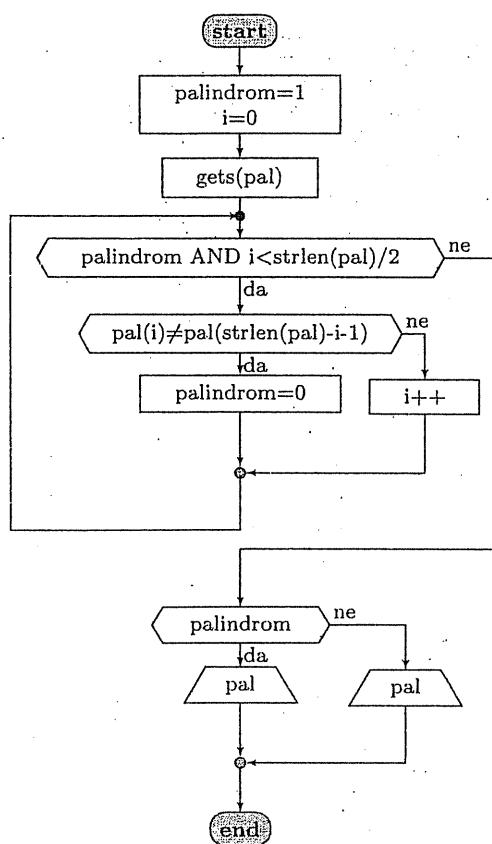
Zadatak 2.58

Nacrtati struktturni dijagram toka algoritma i na programskom jeziku C napisati struktturni program koji određuje i prikazuje dužinu unetog stringa korišćenjem bibliotečke funkcije *strlen*.

Rešenje**Zadatak 2.59**

Nacrtati struktturni dijagram toka algoritma i na programskom jeziku C napisati struktturni program koji ispituje da li je uneti string palindrom. String je palindrom ukoliko se isto čita i sa leva u desno i sa desna u levo. Prikazati odgovarajuću poruku.

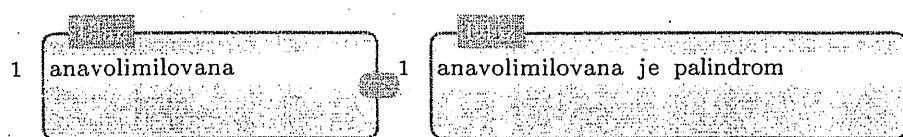
Rešenje



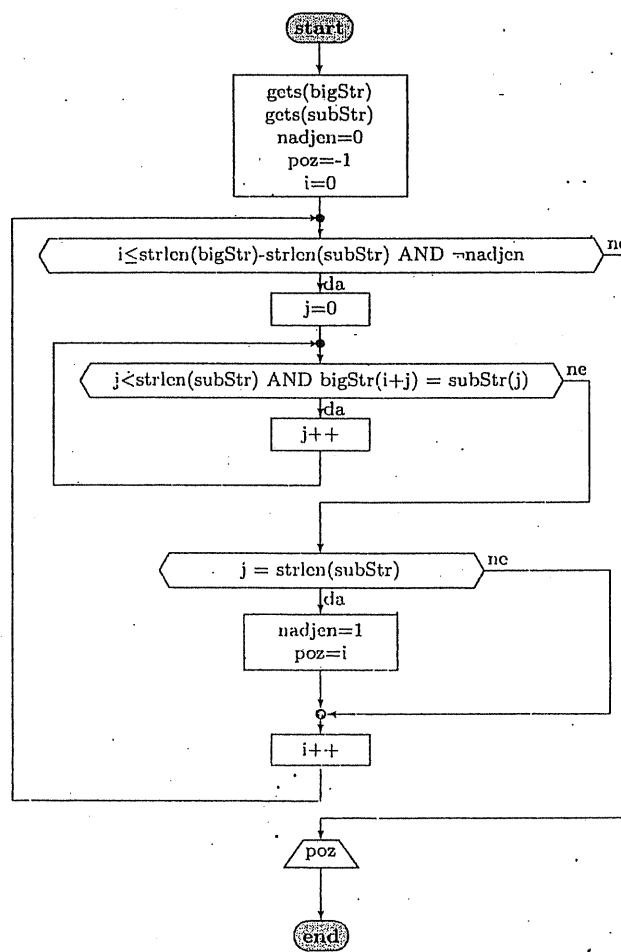
```

1 #include <stdio.h>
2 #include <string.h>
3 void main()
4 {
5     char pal[50];
6     int palindrom = 1, i = 0;
7     gets(pal);
8     while ( palindrom && i<strlen(pal) / 2 )
9     {
10         if ( pal[i] != pal[strlen(pal)-i-1] )
11             palindrom = 0;
12         else
13             i++;
14     }
15     if ( palindrom )
16         printf ("%s je palindrom", pal);
17     else
18         printf ("%s nije palindrom", pal);
19 }
  
```

Zbirka zadataka

**Zadatak 2.60**

Naćrtati struktturni dijagram toka algoritma i na programskom jeziku C napisati struktturni program koji pronalazi prvu poziciju na kojoj se string *subStr* javlja kao podstring u stringu *bigStr*.

Rešenje

Zbirka zadataka

```

1 #include "stdio.h"
2 #include "string.h"
3 void main()
4 {
5     char bigStr[100], subStr[100];
6     int i, j, nadjen, poz;
7     gets(bigStr);
8     gets(subStr);
9     nadjen=0;
10    poz=-1;
11    i=0;
12    while(i<strlen(bigStr)-strlen(subStr) && !nadjen)
13    {
14        j=0;
15        while(j<strlen(subStr) && bigStr[i+j]==subStr[j])
16            j++;
17        if(j==strlen(subStr))
18        {
19            nadjen=1;
20            poz=i;
21        }
22        i++;
23    }
24    printf("Prva pozicija podniza je %d\n", poz);
25 }

```

1 Ana anaanbanal ana
2 ana

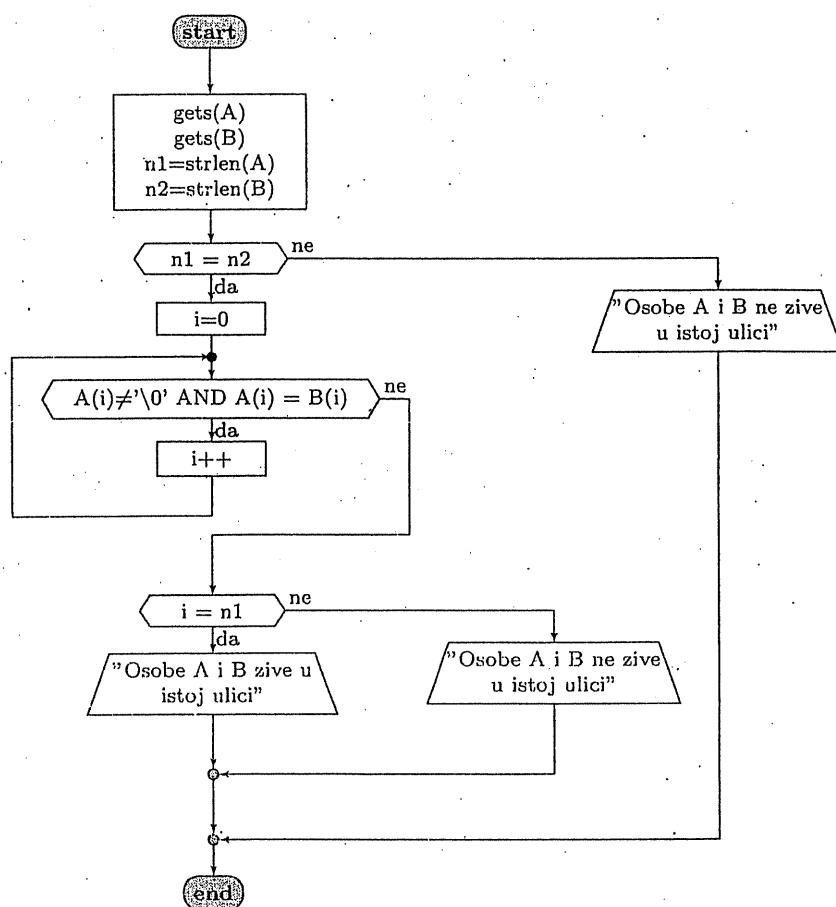
1 Prva pozicija podniza je 13

Zadatak 2.61

Nactrati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni koji na osnovu unetih naziva ulica određuje i prikazuje da li osobe žive u istoj ulici.

Rešenje
prvi način

Zbirka zadataka



```

1 #include <stdio.h>
2 #include <string.h>
3 void main()
4 {
5     char A[50], B[50];
6     int n1,n2,i;
7     gets(A);
8     gets(B);
9     n1=strlen(A);
10    n2=strlen(B);
11    if ( n1 == n2 )
12    {
13        i=0;
14        while (A[i]!='\0' && A[i]==B[i])
15            i++;
16        if (i==n1)
17            printf("Osobe A i B zive u istoj ulici\n");
18        else

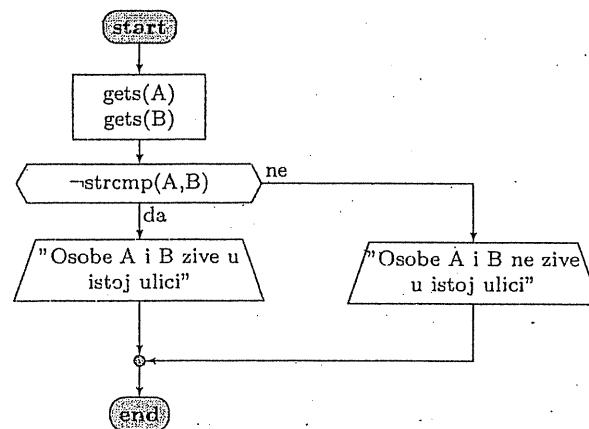
```

```

19         printf("Osobe A i B zive u istoj ulici\n");
20     }
21 else
22     printf ("Osobe A i B ne zive u istoj ulici\n");
23 }

```

drugi način



```

1 #include <stdio.h>
2 #include <string.h>
3 void main()
4 {
5     char A[50], B[50];
6     gets(A);
7     gets(B);
8     if (!strcmp(A,B))
9         printf("Osobe A i B zive u istoj ulici\n");
10    else
11        printf ("Osobe A i B ne zive u istoj ulici\n");
12 }

```

1 Oblacica Rada
2 Oblacica Rada

1 Osobe A i B zive u istoj ulici

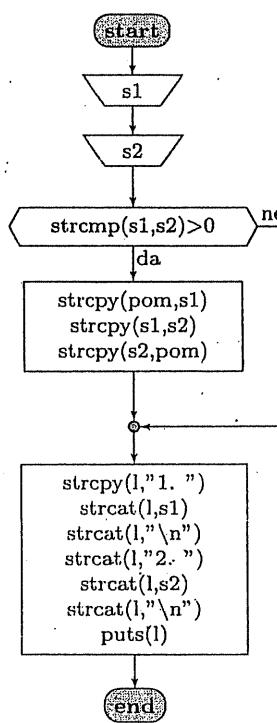
Zadatak 2.62

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji od dva uneta prezimena kreira novi string u kom su prezimena

Zbirka zadataka

uređena u leksički redosled. String kreirati tako da je ispred prvog prezimena redni broj 1, a ispred drugog 2. Prezimena odvojiti karakterom za novi red. Prikazati rezultujući string.

Rešenje



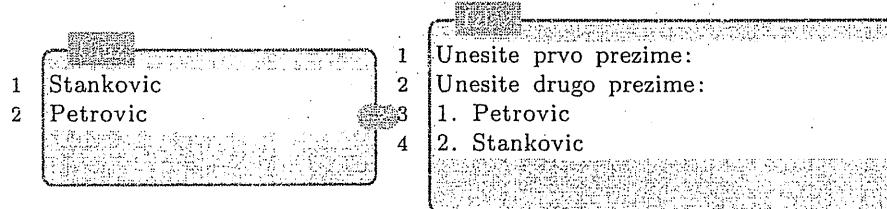
```

1 #include <stdio.h>
2 #include <string.h>
3
4 void main()
5 {
6     char s1[25], s2[25], l[100], pom[25];
7     printf("Unesite prvo prezime:\n");
8     gets(s1);
9     printf("Unesite drugo prezime:\n");
10    gets(s2);
11    if ( strcmp(s1,s2) > 0 )
12    {
13        strcpy(pom,s1);
14        strcpy(s1,s2);
15        strcpy(s2,pom);
16    }
17    strcpy(l,"1. ");
  
```

```

18     strcat(l,s1);
19     strcat(l,"\\n");
20     strcat(l,"2.");
21     strcat(l,s2);
22     strcat(l,"\\n");
23     puts(l);
24 }

```

**Zadatak 2.63**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji u zadatom stringu određuje redni broj najduže reči. Reči u rečenici su razdvojene blanko znacima (jednim ili višem). Prikazati redni broj najduže reči.

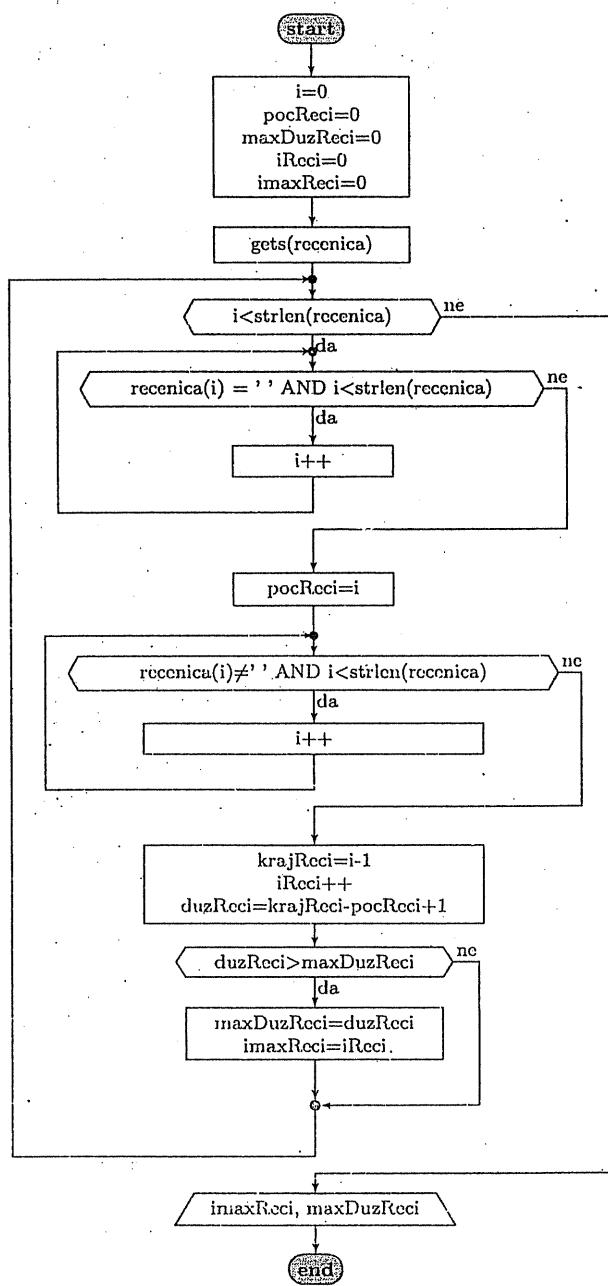
Rešenje

```

1 #include <string.h>
2 #include <stdio.h>
3 void main()
4 {
5     char recenica[70];
6     int i = 0, pocReci = 0, krajReci, duzReci, maxDuzReci = 0, iReci
7     = 0, imaxReci = 0;
8     gets (recenica);
9     while ( i < strlen ( recenica ) )
10    {
11        while ( recenica[i] == ' ' && i < strlen ( recenica ) )
12            i++;
13        pocReci = i;
14        while ( recenica[i] != ' ' && i < strlen ( recenica ) )
15            i++;
16        krajReci = i - 1;
17        iReci++;
18        duzReci = krajReci - pocReci + 1;
19        if ( duzReci > maxDuzReci )
20        {
21            maxDuzReci = duzReci;
22            imaxReci = iReci;
23        }
24    }

```

Zbirka zadataka



```

24     printf ("Najduza je rec br. %d i duzine je %d",
25 }      imaxReci,maxDuzReci);

```

1

Kroz Nis protice reka
Nisava

1

Najduza je rec br. 3 i duzine je 7

2.2 Matrice

Zadatak 2.64

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje zbir elemenata

- a) 3. vrste,
- b) 2. kolone,
- c) na glavnoj dijagonali,
- d) na sporednoj dijagonali,
- e) iznad glavne dijagonale,
- f) ispod sporedne dijagonale.

Rešenje

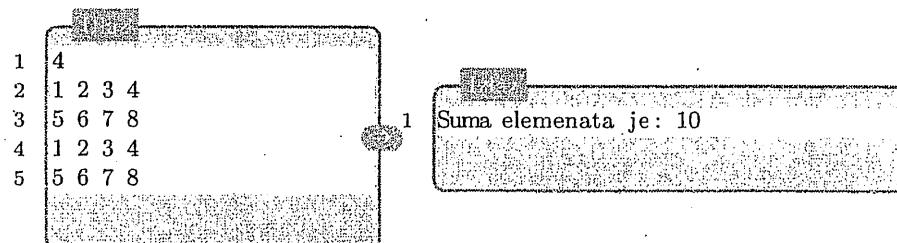
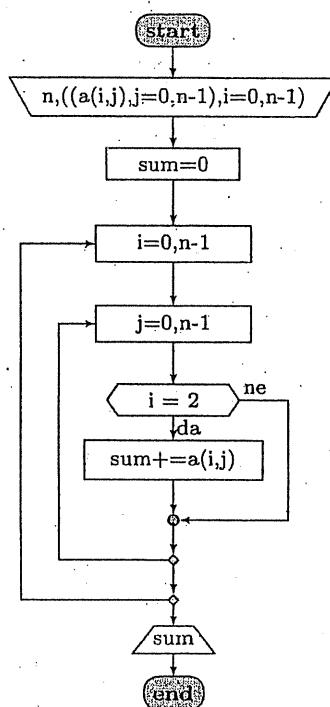
a)

```

1 #include <stdio.h>
2 void main()
3 {
4     int i,j,n,a[20][20],sum;
5     scanf("%d",&n);
6     for(i=0;i<n;i++)
7         for(j=0;j<n;j++)
8             scanf("%d",&a[i][j]);
9     sum = 0;
10    for(i=0;i<n;i++)
11        for(j=0;j<n;j++)
12            if( i == 2 )
13                sum+=a[i][j];
14    printf("Suma elemenata je: %d\n", sum);
15 }

```

Zbirka zadataka

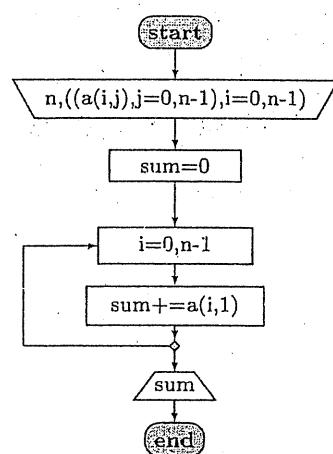


b)

```

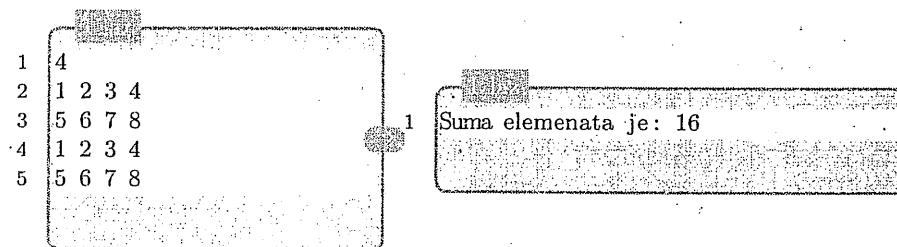
1 #include <stdio.h>
2 void main()
3 {
4     int i,j,n,a[20][20],sum;
5     scanf("%d",&n);
6     for(i=0;i<n;i++)
7         for(j=0;j<n;j++)
8             scanf("%d",&a[i][j]);
9     sum = 0;
10    for(i=0;i<n;i++)
  
```

Zbirka zadataka



```

11         sum+=a[i][1];
12         printf("Suma elemenata je: %d\n", sum);
13     }
  
```



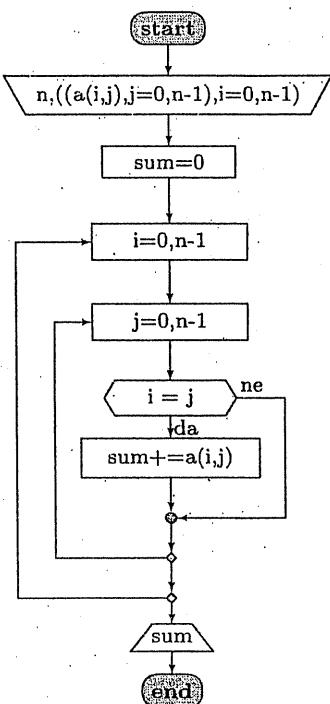
c)

```

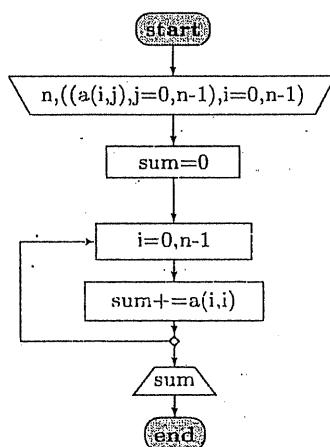
1 #include <stdio.h>
2 void main()
3 {
4     int i,j,n,a[20][20],sum;
5     scanf("%d",&n);
6     for(i=0;i<n;i++)
7         for(j=0;j<n;j++)
8             scanf("%d",&a[i][j]);
9     sum = 0;
10    for(i=0;i<n;i++)
11        for(j=0;j<n;j++)
12            if( i == j )
13                sum+=a[i][j];
  
```

Zbirka zadataka

varijanta 1



varijanta 2

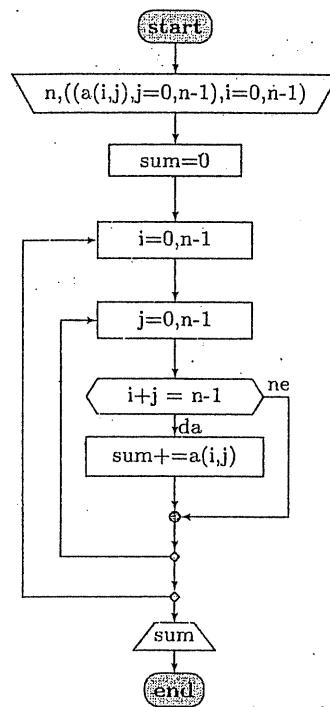


```

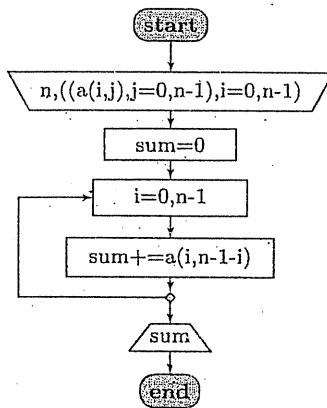
14     printf("Suma elemenata je : %d\n", sum);
15 }
  
```

Zbirka zadataka

d) varijanta 1



varijanta 2



```

1 #include <stdio.h>
2 void main()
3 {
4     int i,j,n,a[20][20],sum;
5     scanf("%d",&n);
  
```

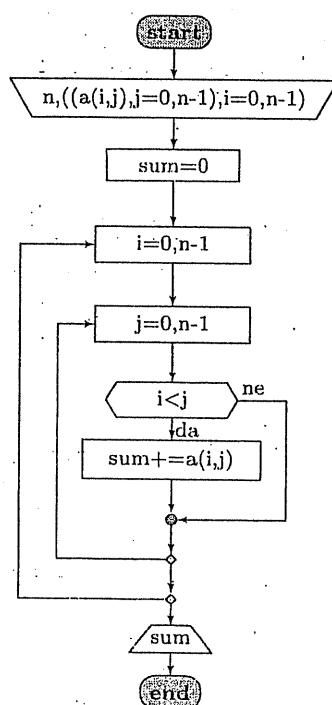
Zbirka zadataka

```

6     for(i=0;i<n; i++)
7         for(j=0;j<n; j++)
8             scanf("%d",&a[i][j]);
9     sum = 0;
10    for(i=0;i<n; i++)
11        for(j=0;j<n; j++)
12            if( i+j == n-1 )
13                sum+=a[i][j];
14    printf("Suma elemenata je : %d\n", sum);
15 }

```

e)



```

1 #include <stdio.h>
2 void main()
3 {
4     int i,j,n,a[20][20],sum;
5     scanf("%d",&n);
6     for(i=0;i<n; i++)
7         for(j=0;j<n; j++)
8             scanf("%d",&a[i][j]);

```

Zbirka zadataka