

# 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 Emina 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 pomoćni 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  
zadataka 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- [аутор]  
а) Програмирање - Вежбе б) Програмски  
језик "C" - Задаци  
COBISS.SR-ID 191375372

Preštampavanje ili umnožavanje ove knjige nije dozvoljeno bez pismene dozvole izdavača.

Tiraž: 300 primeraka

Štampa: Unigraf, Niš

---

# Predgovor

U ovoj zbirci zadataka metodički je prezentovana kolekcija rešenih problema vezanih za kreiranje strukturnih 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 ovladavanje 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. Obrađene 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. Obrađeni su tekstualni fajlovi sa sekvencijalnim pristupom.

Poglavlja su koncipirana tako da su na početku poglavlja jednostavni 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 kompajler autora doc. dr Vladimira Ćirića, koji vrši leksičku i sintaksnu analizu C programa i generiše dijagram toka korišćenjem  $\text{\LaTeX}$  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 greške i nedostaci u ponuđenim rešenjima. Autori će biti zahvalni ukoliko im se ukaže na eventualne greske.

Autori

# Sadržaj

|          |  |            |
|----------|--|------------|
| <b>1</b> | <b>Osnovne upravljačke strukture</b>                           | <b>1</b>   |
| 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         |
| <b>2</b> | <b>Linearne indeksirane strukture podataka</b>                 | <b>49</b>  |
| 2.1      | Nizovi . . . . .   | 49         |
| 2.1.1    | Brojevnii nizovi . . . . .                                     | 49         |
| 2.1.2    | Znakovni nizovi - stringovi . . . . .                          | 85         |
| 2.2      | Matrice . . . . .  | 95         |
| <b>3</b> | <b>Funkcije i rad sa fajlovima</b>                             | <b>121</b> |
| 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

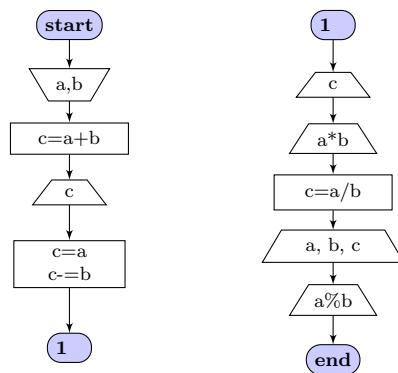
### 1.1 Struktura C programa i osnovne aritmetičke operacije

#### 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.

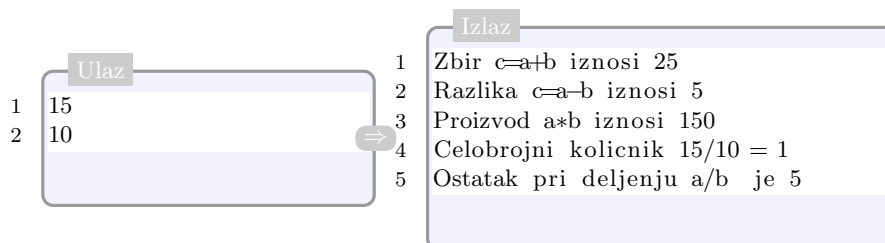
#### Rešenje



```

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.

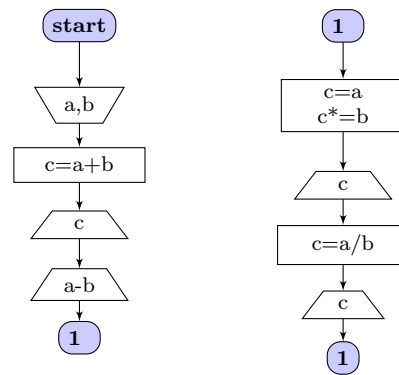
### Rešenje

```

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;

```

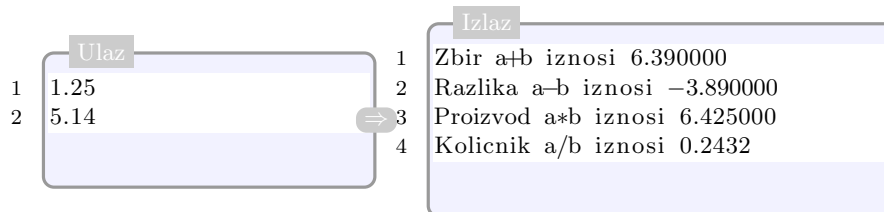
Zbirka zadataka



```

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 vrenosti promenljivih nakon zamene mesta.

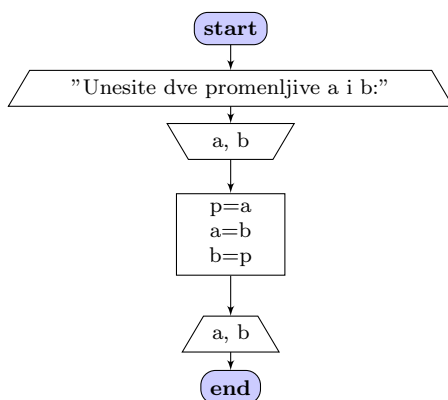
### Rešenje

```

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;

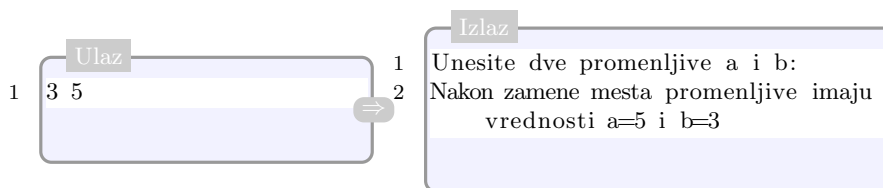
```

Zbirka zadataka



```

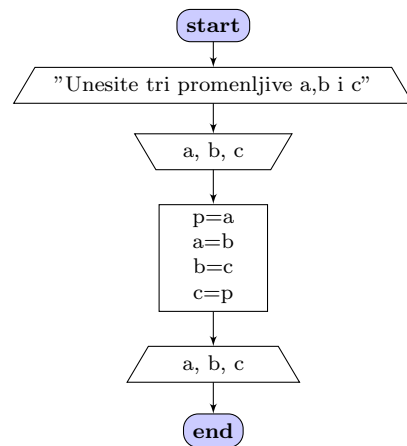
10     printf("Nakon zamene mesta promenljive imaju vrednosti a=%d i b=%d\n", a,b);
11 }
  
```



#### Zadatak 1.4

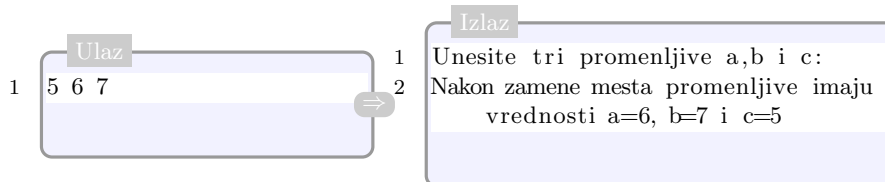
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program za unos tri celobrojne vrednosti u promenljive  $a$ ,  $b$  i  $c$ . Nakon unosa izvršiti rotaciju vrednosti promenljivih u levo, tako da se vrednost promenljive  $c$  nađe u promenljivoj  $b$ , vrednost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

#### Rešenje



```

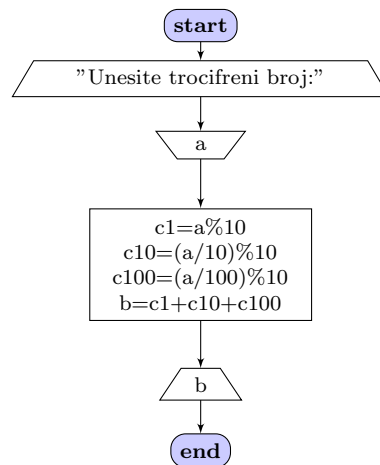
1  #include <stdio.h>
2
3  void main()
4  {
5      int a,b,c,p;
6      printf("Unesite tri promenljive a,b i c:\n");
7      scanf("%d%d%d",&a,&b,&c);
8      p=a;
9      a=b;
10     b=c;
11     c=p;
12     printf("Nakon zamene mesta promenljive imaju vrednosti a=%d, b=%d i c=%d\n", a,b,c);
13 }
  
```

**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.

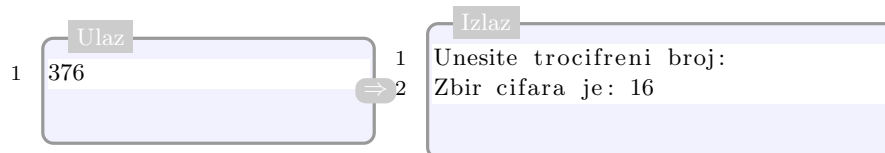
**Rešenje**

*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 }
  
```

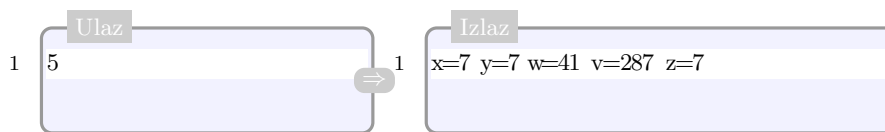
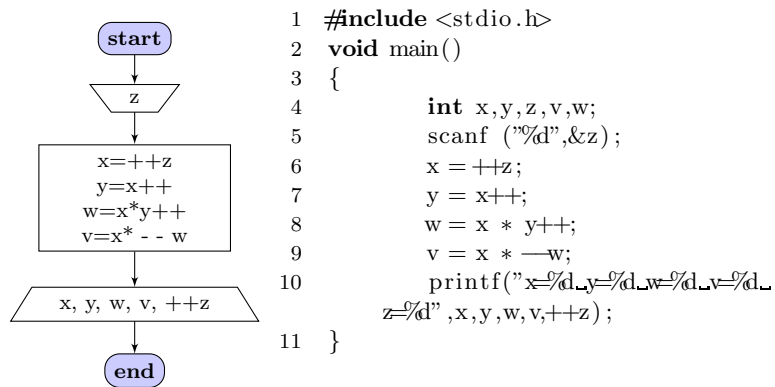


### Zadatak 1.6

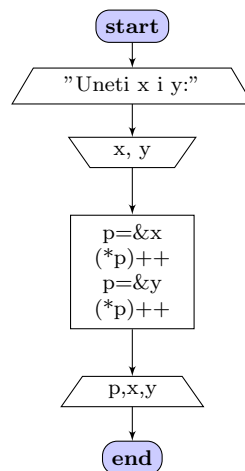
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji vrednost promenljive  $z$ , zadatu od strane korisnika, jednim iskazom jezika C inkrementira i dodeljuje promenljivoj  $x$ . Jednim iskazom vrednost promenljive  $x$  dodeliti promenljivoj  $y$  i uvećati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

### Rešenje

Zbirka zadataka

**Zadatak 1.7**

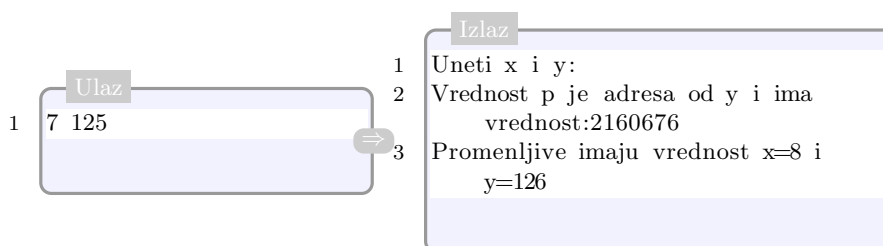
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji vrednosti promenljivih  $x$  i  $y$  uvećava za 1 korišćenjem jednog pokazivača  $p$ . Vrednosti  $x$  i  $y$  zadaje korisnik. Prikazati vrednosti promenljivih i vrednost pokazivača nakon modifikacije.

**Rešenje**

```

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 smeštena 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

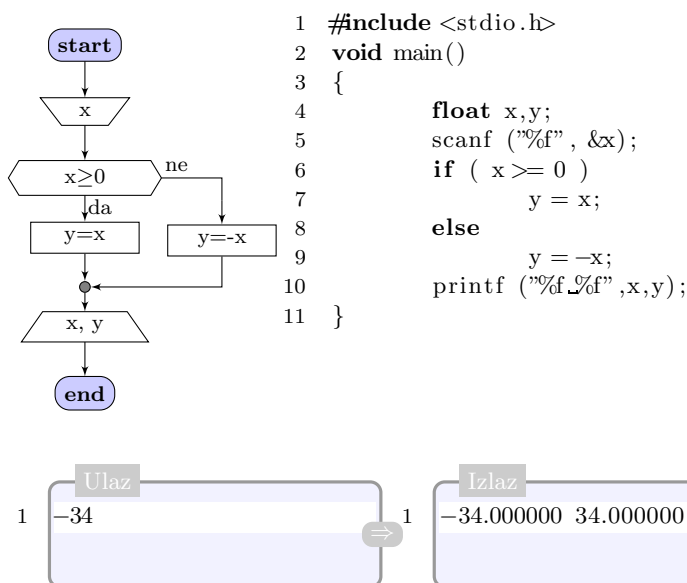
Apsolutna vrednost broja matematički se može izraziti kao

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

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**

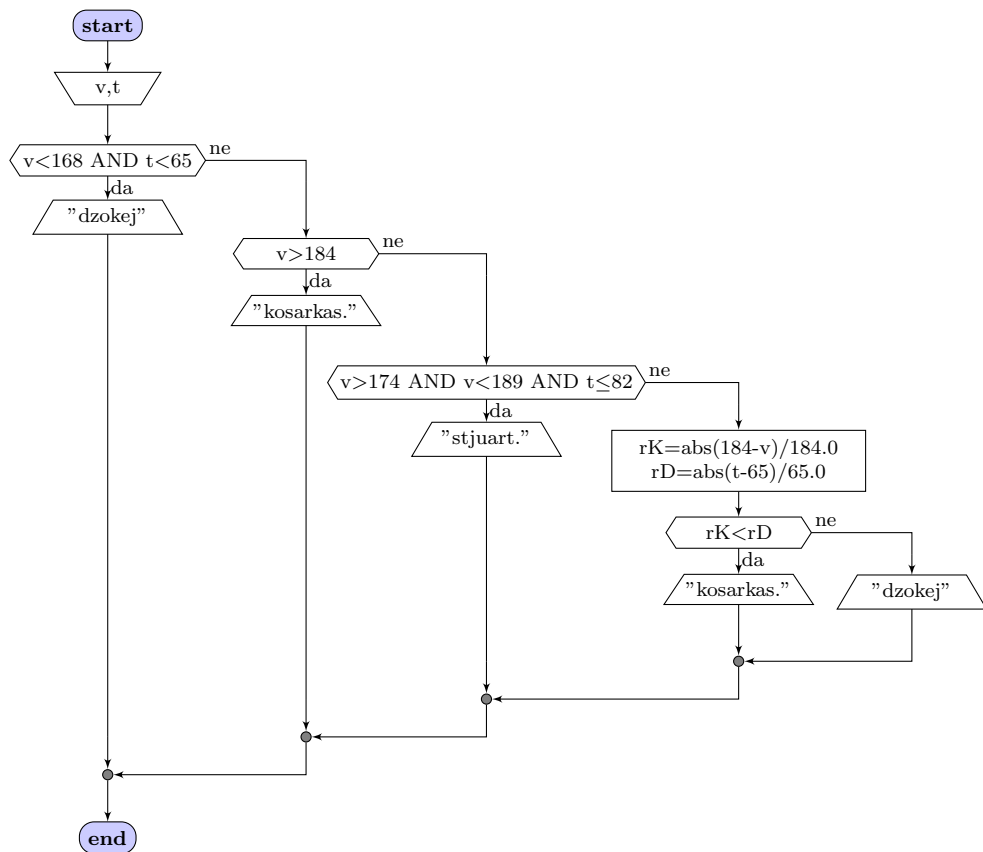
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji na osnovu karakterističnih podataka daje preporuke za životno opredeljenje. Program odlučuje, na osnovu unete visine i težine (celi brojevi, u centimetrima, odnosno kilogramima) koja od tri buduće profesije 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 težinu u kilogramima:\n");
10     scanf("%d",&t);
11     if (v<168 && t<65)
  
```

Zbirka zadataka

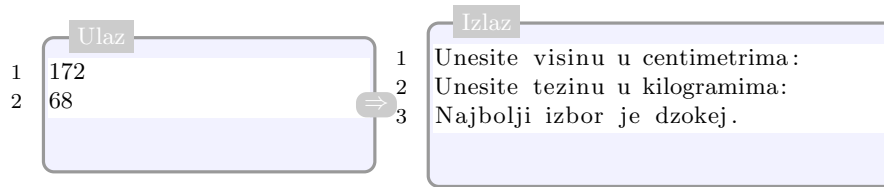


```

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

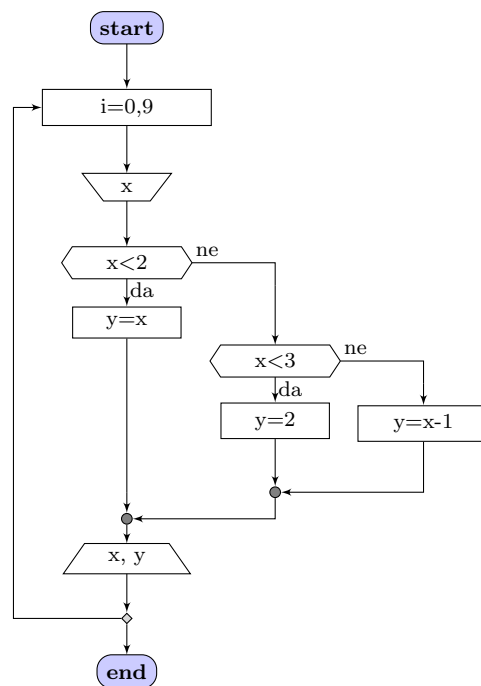
### Zadatak 1.10

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



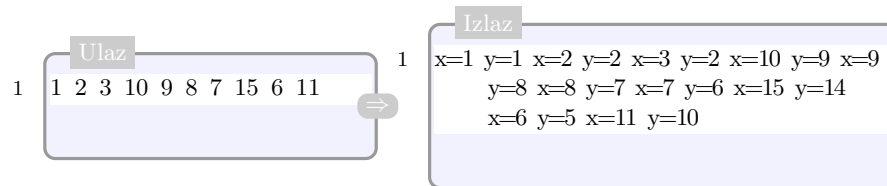
```
1 #include <stdio.h>
```

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 ("x=%d,y=%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.

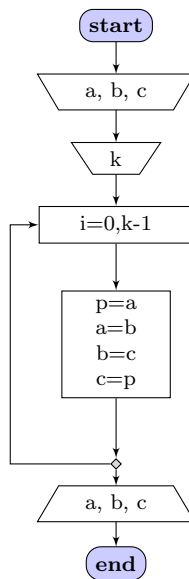
#### Rešenje

```

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    }

```

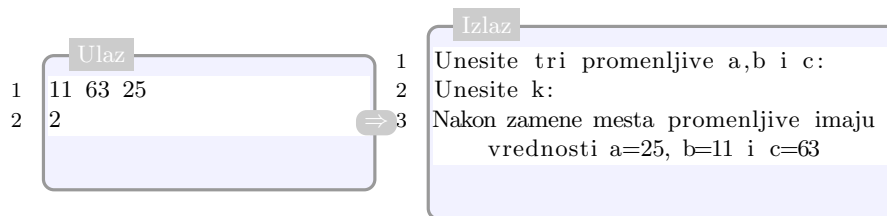
Zbirka zadataka



```

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

```



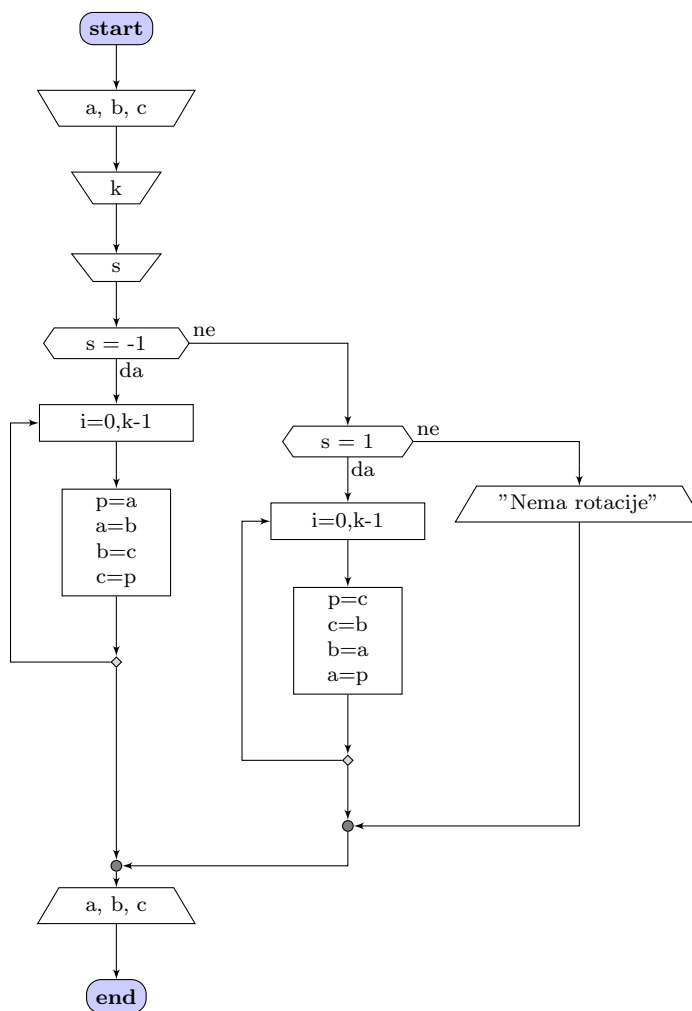
### Zadatak 1.12

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;

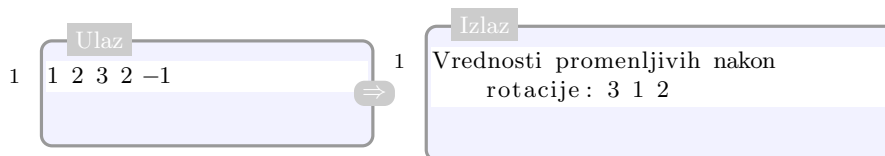
```

Zbirka zadataka

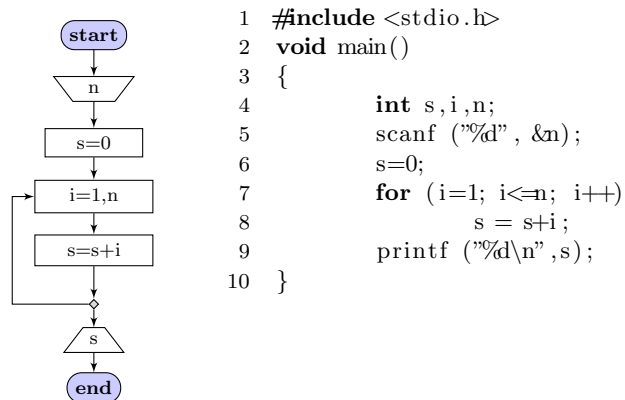
```

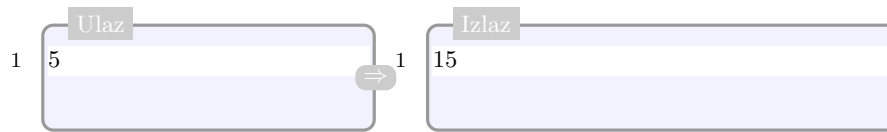
13             b = c;
14             c = p;
15         }
16     else if (s == 1)
17         for ( i = 0; i < k; i++ )
18         {
19             p = c;
20             c = b;
21             b = a;
22             a = p;
23         }
24     else
25         printf ("Nema rotacije");
26     printf ("Vrednosti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

```

**Zadatak 1.13**

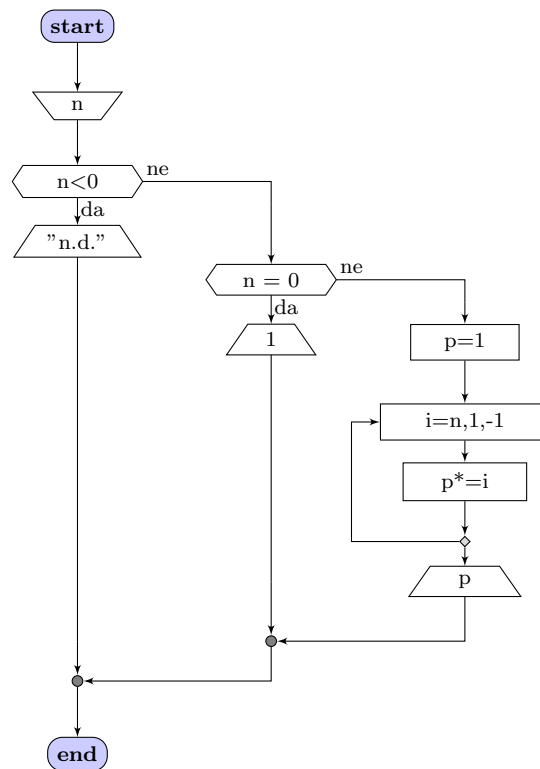
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje sumu prvih  $N$  prirodnih brojeva.

**Rešenje**

**Zadatak 1.14**

Nacrtati strukturni dijagram toka algoritma i napisati strukturni 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;

```

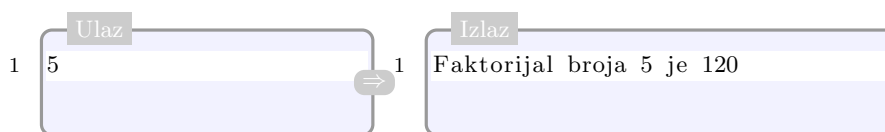
Zbirka zadataka



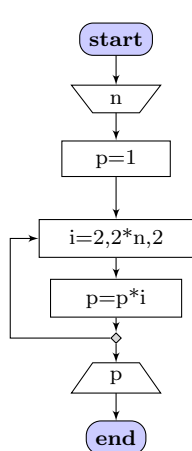
```

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**

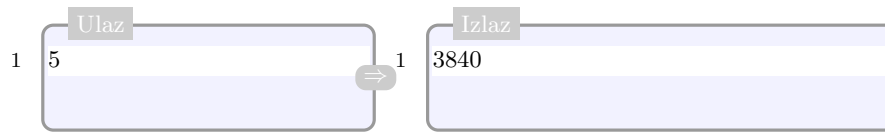
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje proizvod prvih  $N$  parnih prirodnih brojeva.

**Rešenje**

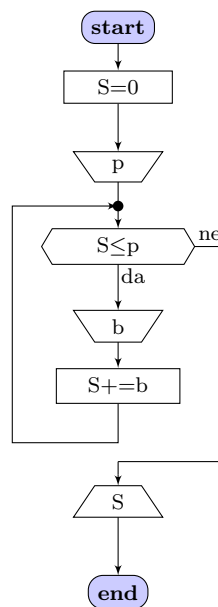
```

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 }

```

**Zadatak 1.16**

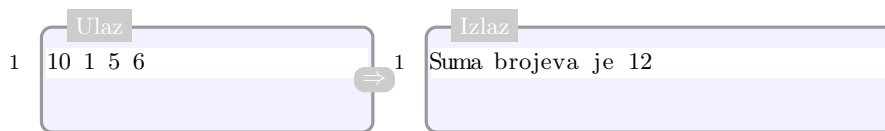
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program za sumiranje brojeva. Brojevi se unose sa tastature redom, a sumiranje se vrši sve dok suma ne postane veća od unapred zadate vrednosti.

**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**

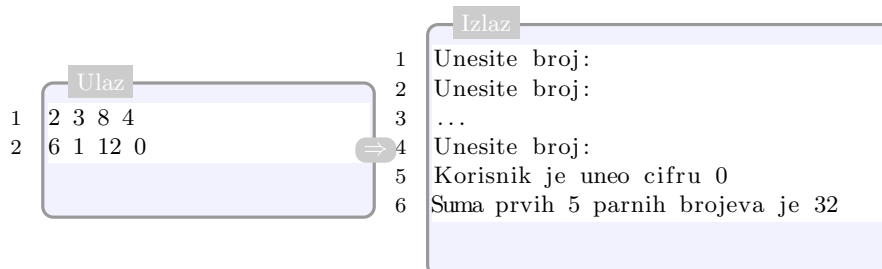
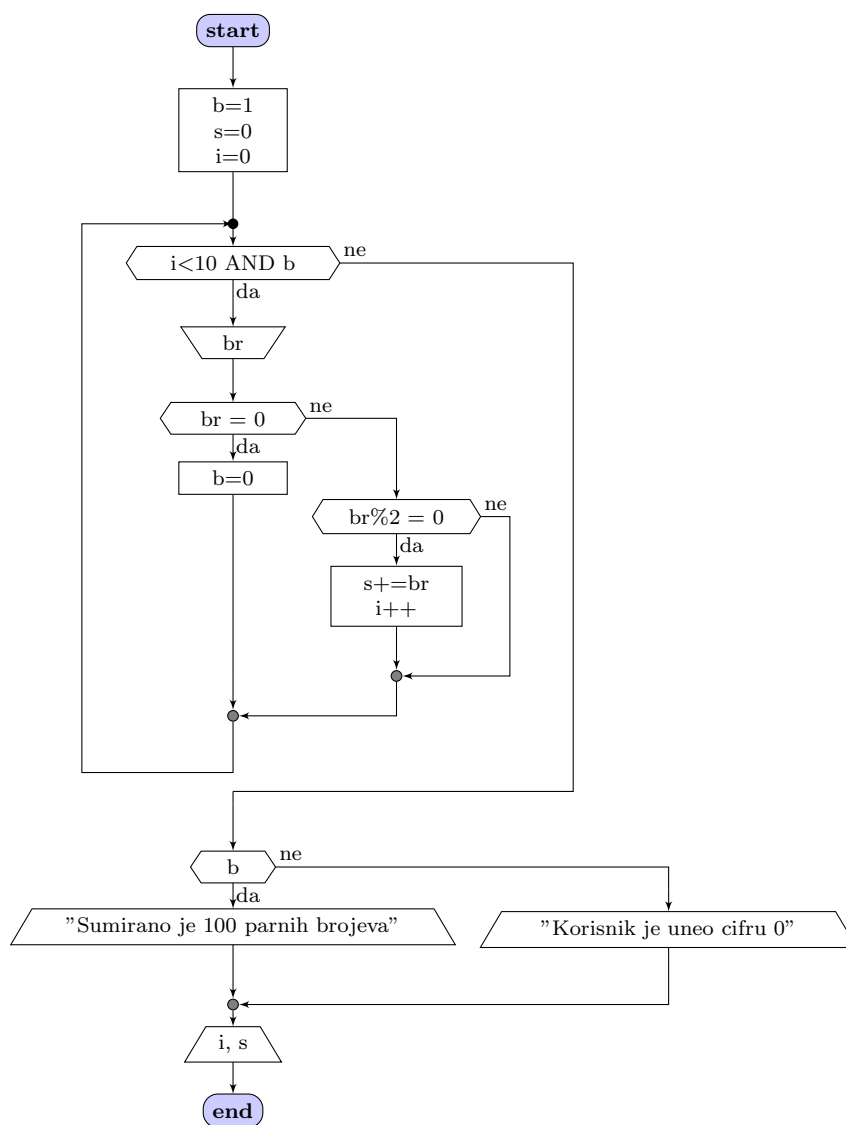
Nacrtati strukturni dijagram toka algoritma i na programsku jeziku C napisati program koji određuje sumu prvih 100 unetih parnih brojeva. Brojeve unosi korisnik redom, sve dok ne unese nulu, ili 100 parnih brojeva. Kada se unese nula prekinuti sumiranje brojeva bez obzira da li je izvršeno sumiranje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

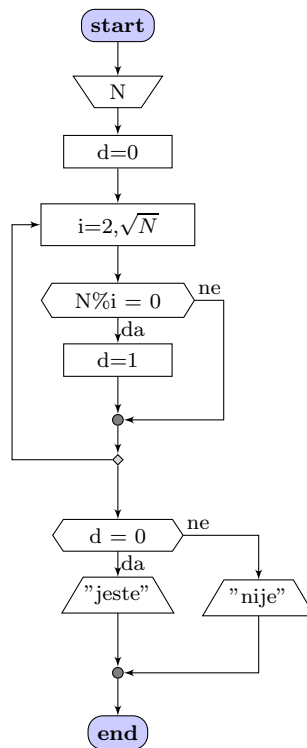


Zbirka zadataka

**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**

```

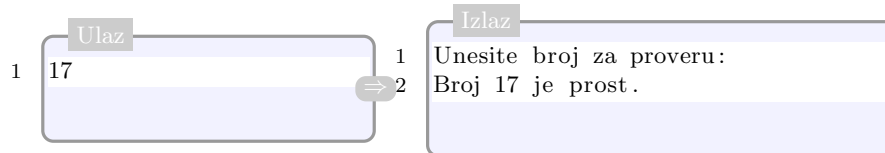
1 #include <stdio.h>
2 #include <math.h>
3 void main()
4 {
5     int N, i, d;
6     printf("Unesite broj za proveru:\n");
7     scanf("%d",&N);
8     d = 0;
9     for(i=2;i<=sqrt(N);i++)
  
```

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 }

```

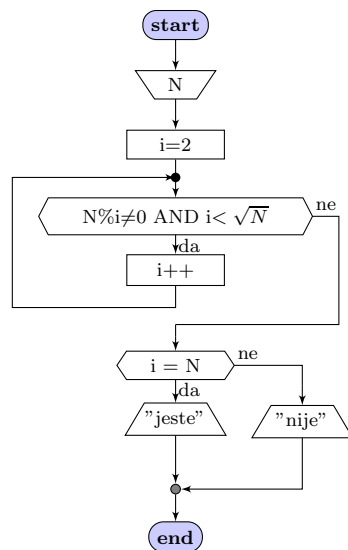


### Zadatak 1.19

Nacrtati 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. 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.

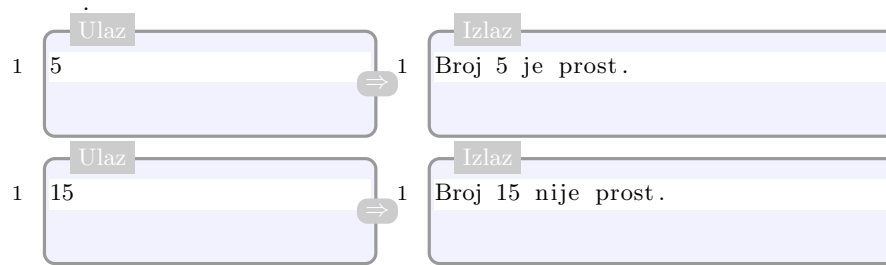
### Rešenje



```

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

```

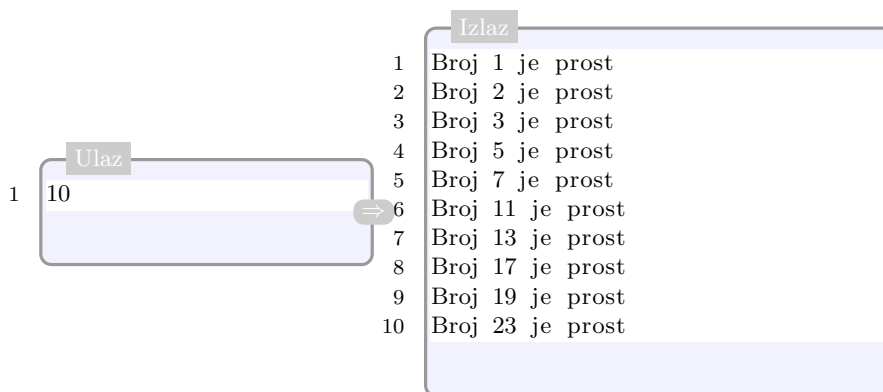
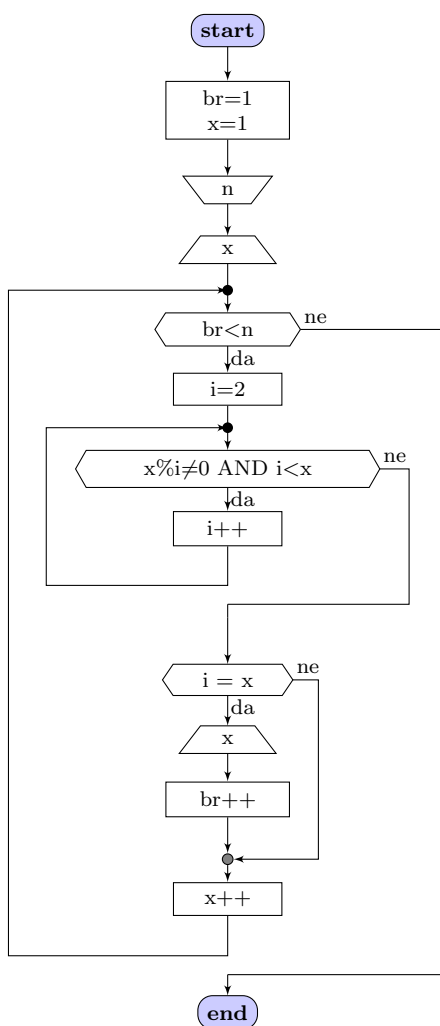
**Zadatak 1.20**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni 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 }
```

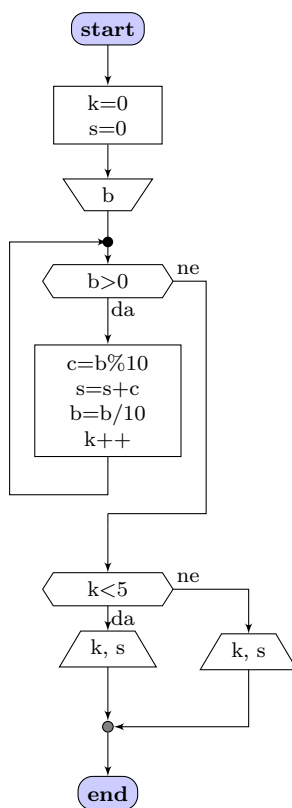


Zbirka zadataka



**Zadatak 1.21**

Nacrtati strukturni dijagram toka 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 zadanog 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;

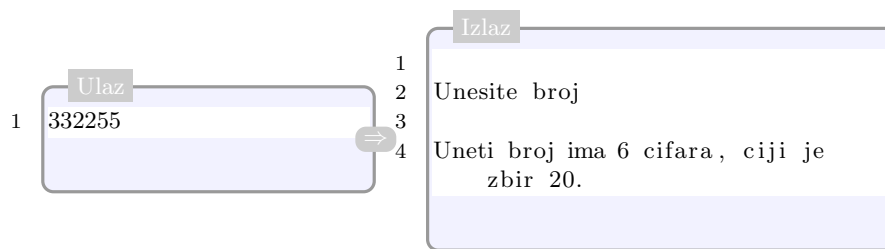
```

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 strukturni 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.

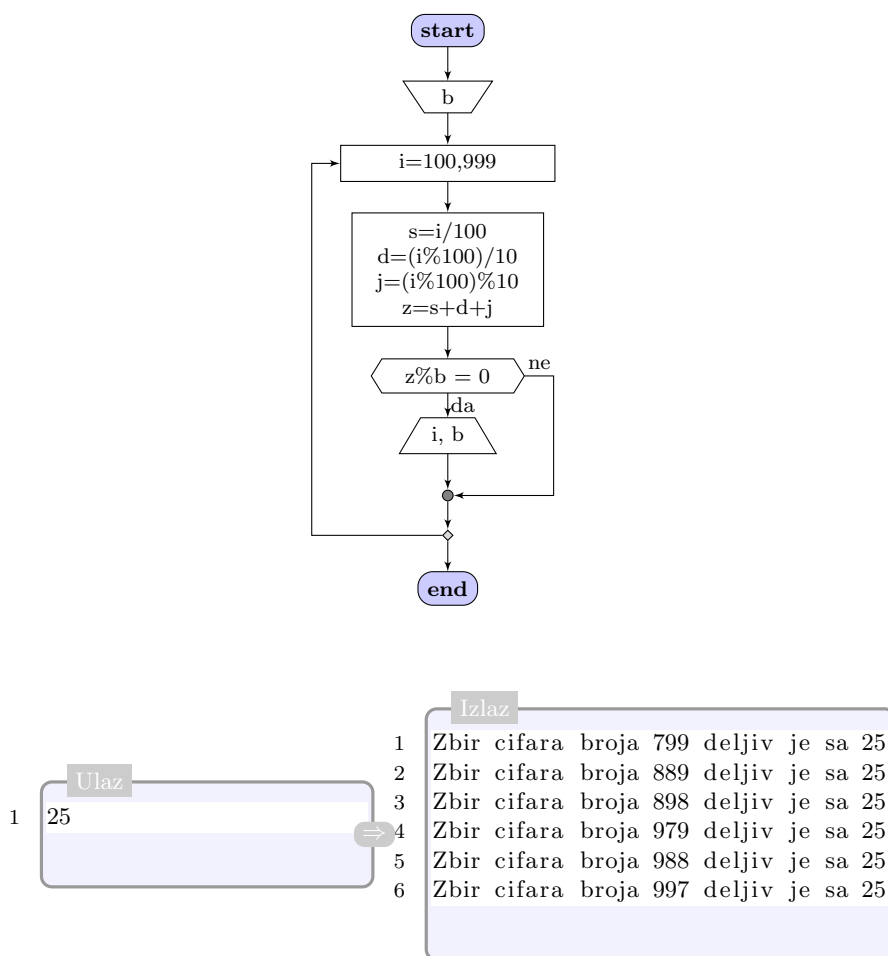
### Rešenje

```

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

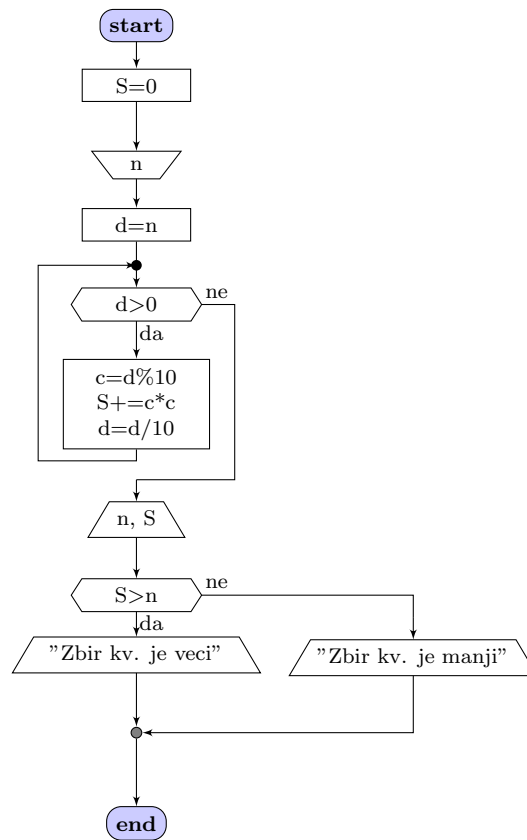
**Zadatak 1.23**

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*

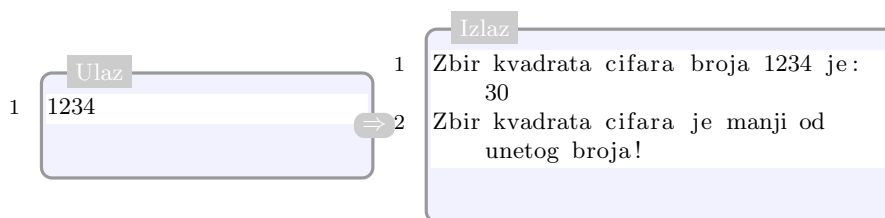


```

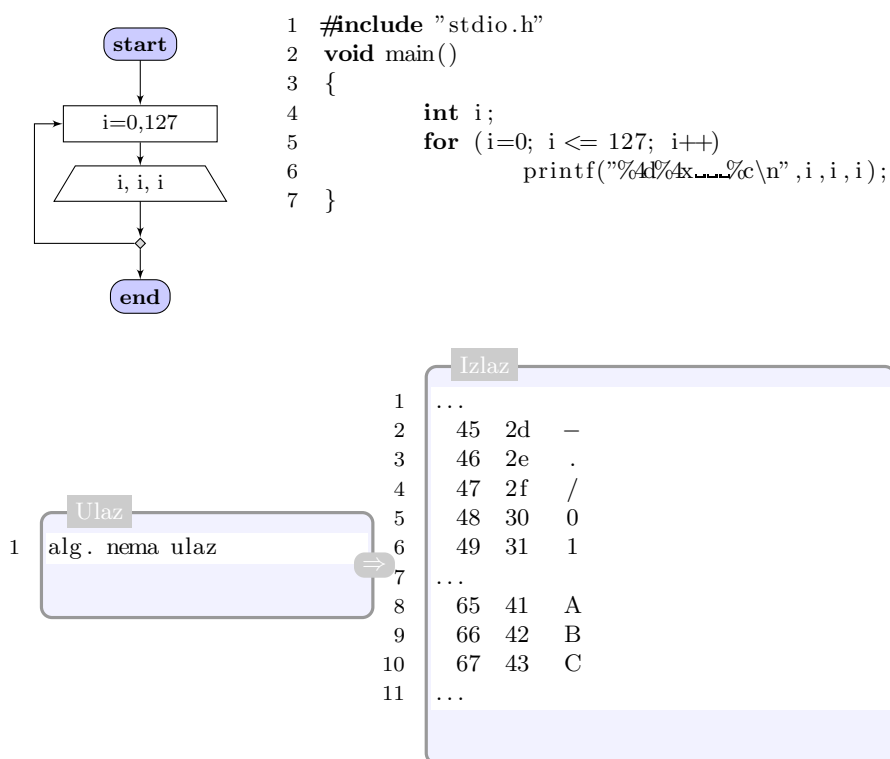
1  #include <stdio.h>
2  void main()
3  {
4      int n,S = 0,c,d;
5      scanf ("%d",&n);
6      d = n;
7      while ( d > 0 )
8      {
9          c = d % 10;
10         S += c * c;
11         d = d / 10;
12     }
13     printf ("Zbir kvadrata cifara broja %d je: %d\n",n,S);
14     if ( S > n )
15         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.

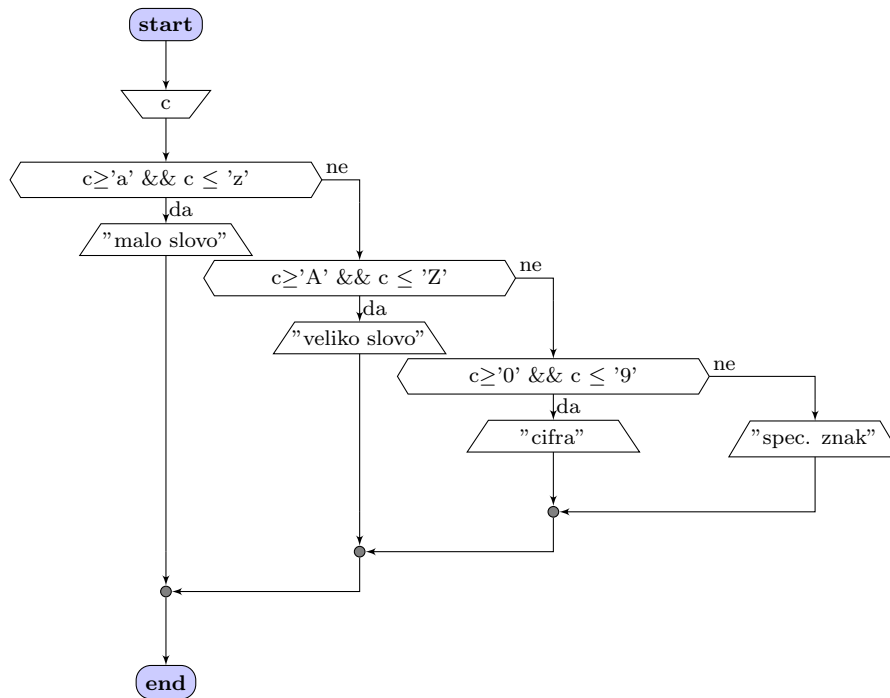
**Rešenje****Zadatak 1.25**

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.

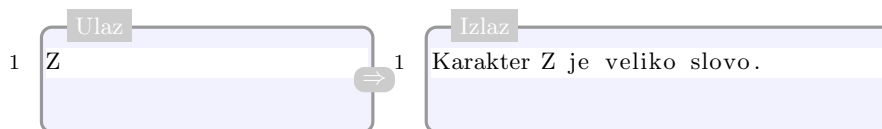
### Rešenje



```

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 }
  
```

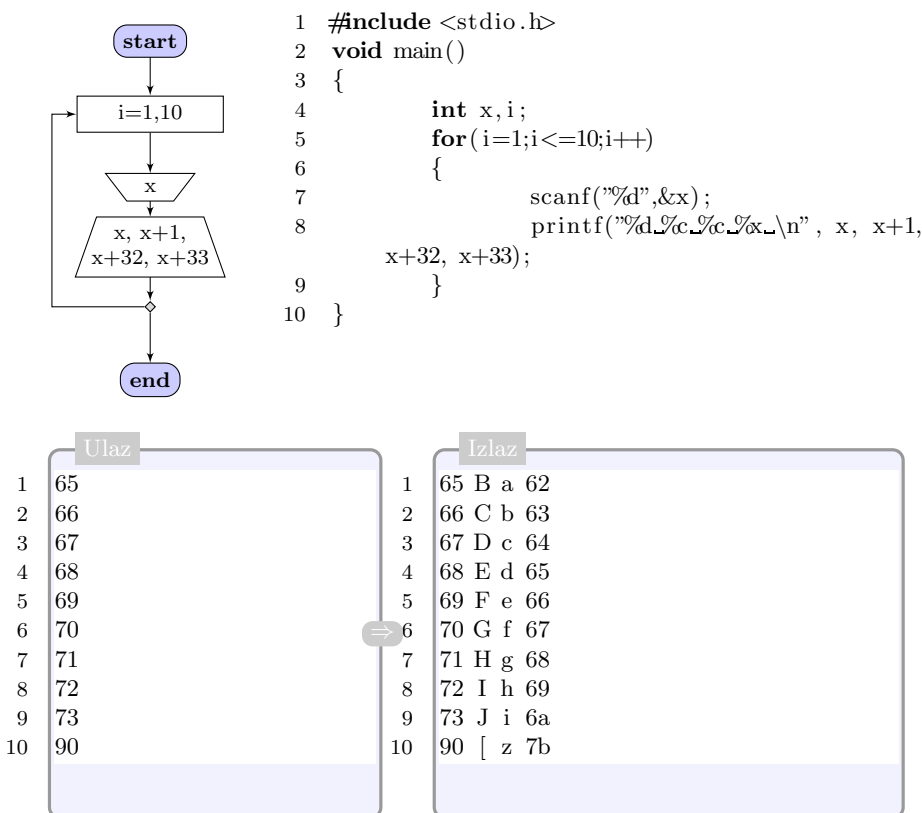
Zbirka zadataka



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).

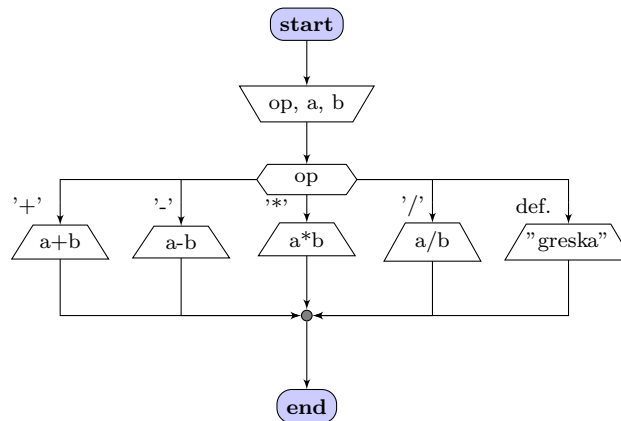
## Rešenje



Zbirka zadataka

**Zadatak 1.27**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program kojim se vrši unos jednog karaktera, koji predstavlja aritmetičku operaciju i dva cela broja. Prikazati unešene 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;

```

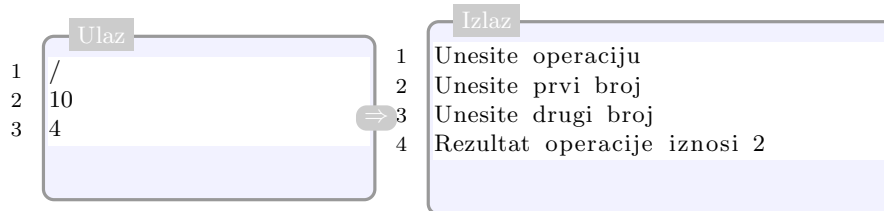
Zbirka zadataka



```

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

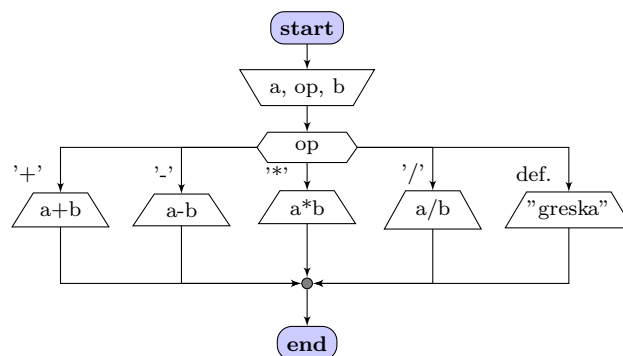
```

**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 {

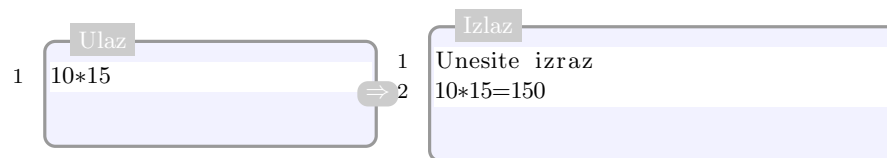
```

*Zbirka zadataka*

```

4      int a,b, rez;
5      char op;
6      printf("Unesite izraz\n");
7      scanf("%d%c%d",&a,&op,&b);
8      printf("%d%c%d",a,op,b);
9      printf("=");
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

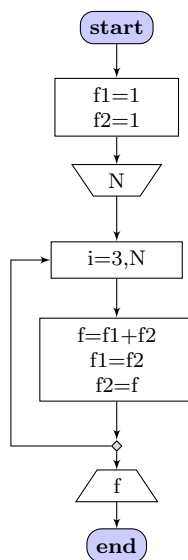
$$f_i = f_{i-1} + f_{i-2},$$

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

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

#### Rešenje

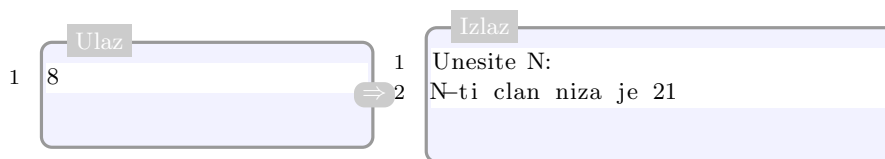
*Zbirka zadataka*



```

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", f);
14 }

```

**Zadatak 1.30**

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

*Zbirka zadataka*

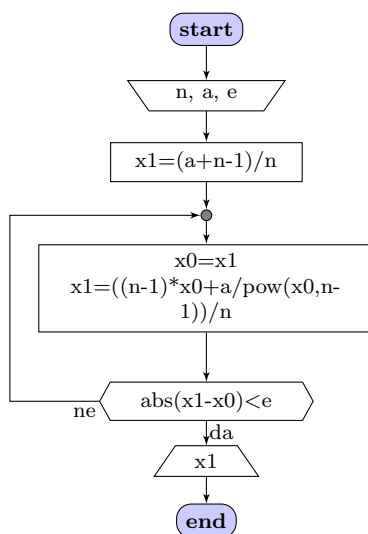
iterativnog postupka

$$x_0 = \frac{(a + n - 1)}{n}$$

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

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

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

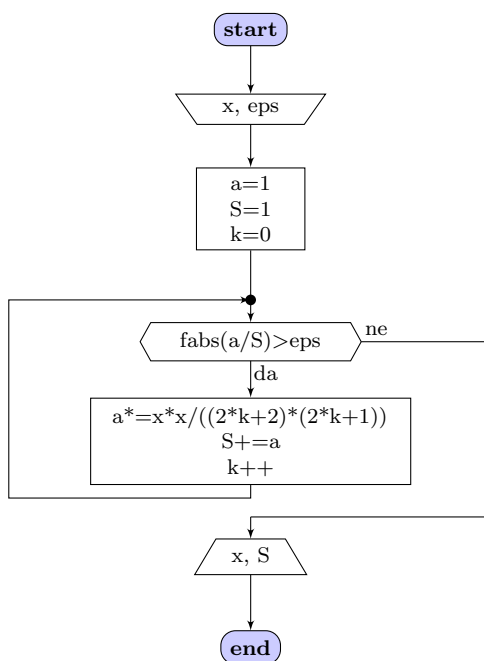
$$\frac{a_{k+1}}{a_k} = \frac{\frac{x^{2(k+1)}}{(2(k+1))!}}{\frac{x^{2k}}{(2k)!}} = \frac{x^2}{(2k+2)(2k+1)},$$

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:

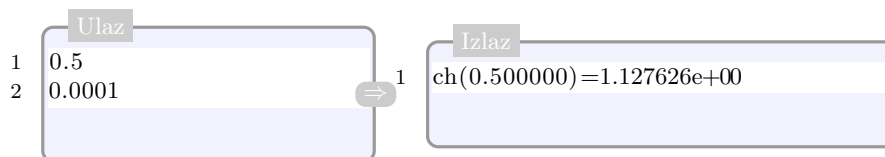
*Zbirka zadataka*



```

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 }

```



Zbirka zadataka

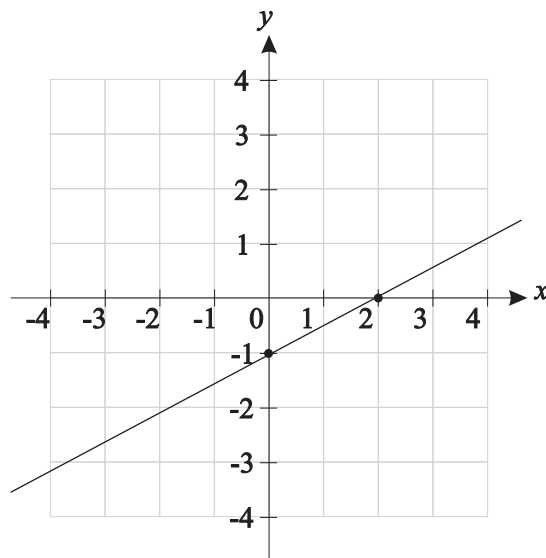
## 1.4 Geometrijski problemi

### 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.



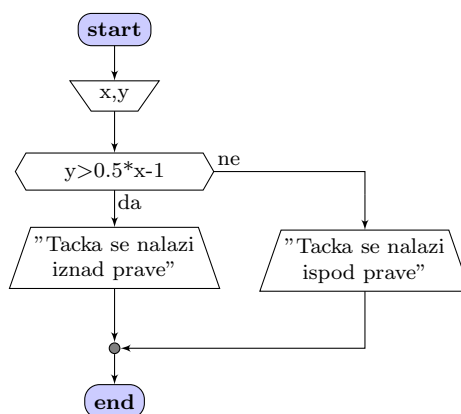
Slika 1.1

### 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 prave dobijamo jednačinu konkretne prave datu na slici kao

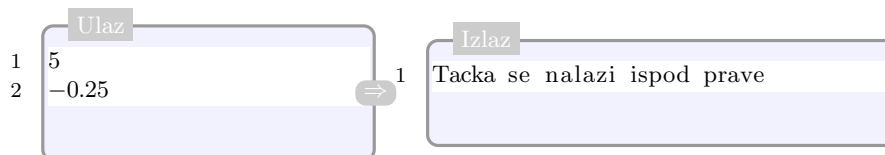
$$y = \frac{1}{2}x - 1.$$

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 konkretne 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 }
  
```



### Zadatak 1.33

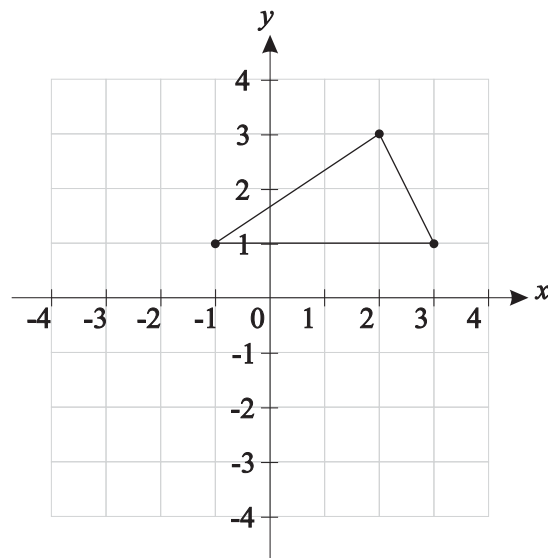
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje koliko se tačaka, od ukupno  $N$ , nalazi unutar, a koliko van trougla zadatog na slici. Parametar  $N$  zadaje korisnik, a za njim redom i koordinate tačaka  $(x, y)$ . Jednačina 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.

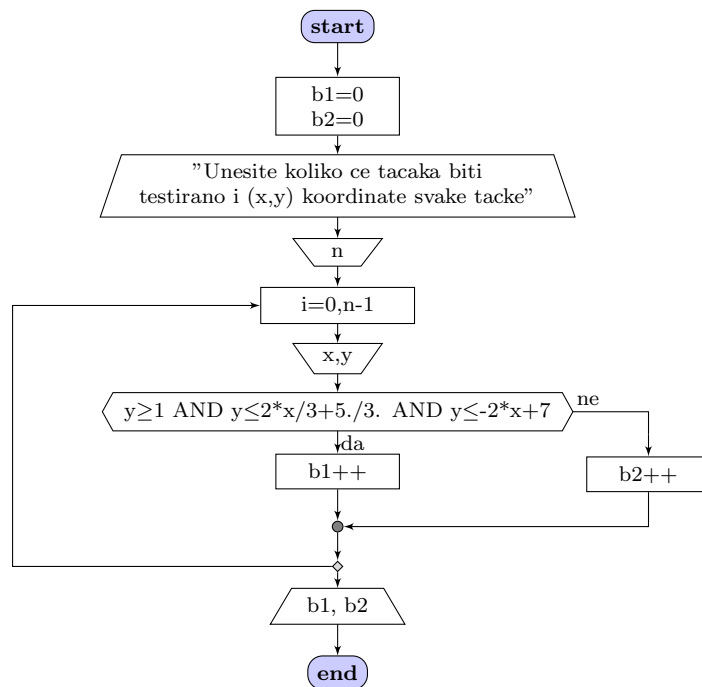
Zbirka zadataka





Slika 1.2

Rešenje

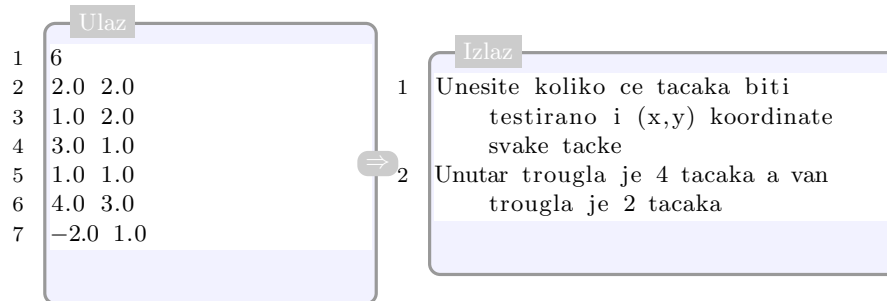


Zbirka zadataka

```

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 }

```



#### Zadatak 1.34

Dato je  $N$  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** 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:

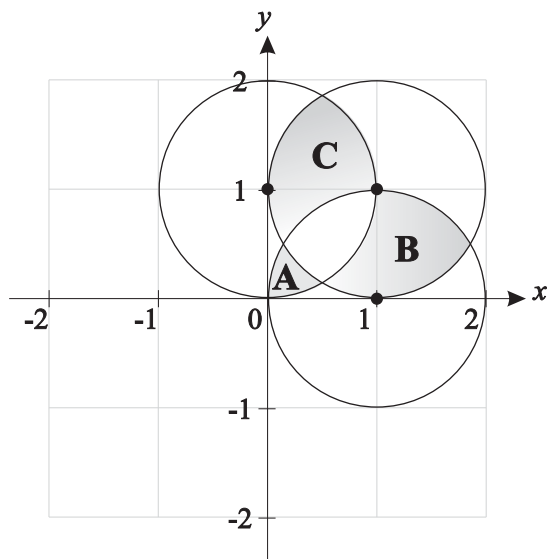
$$K_1 : (x - 1)^2 + y^2 = 1$$

$$K_2 : (x - 1)^2 + (y - 1)^2 = 1$$

$$K_3 : x^2 + (y - 1)^2 = 1$$

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

Zbirka zadataka



Slika 1.3

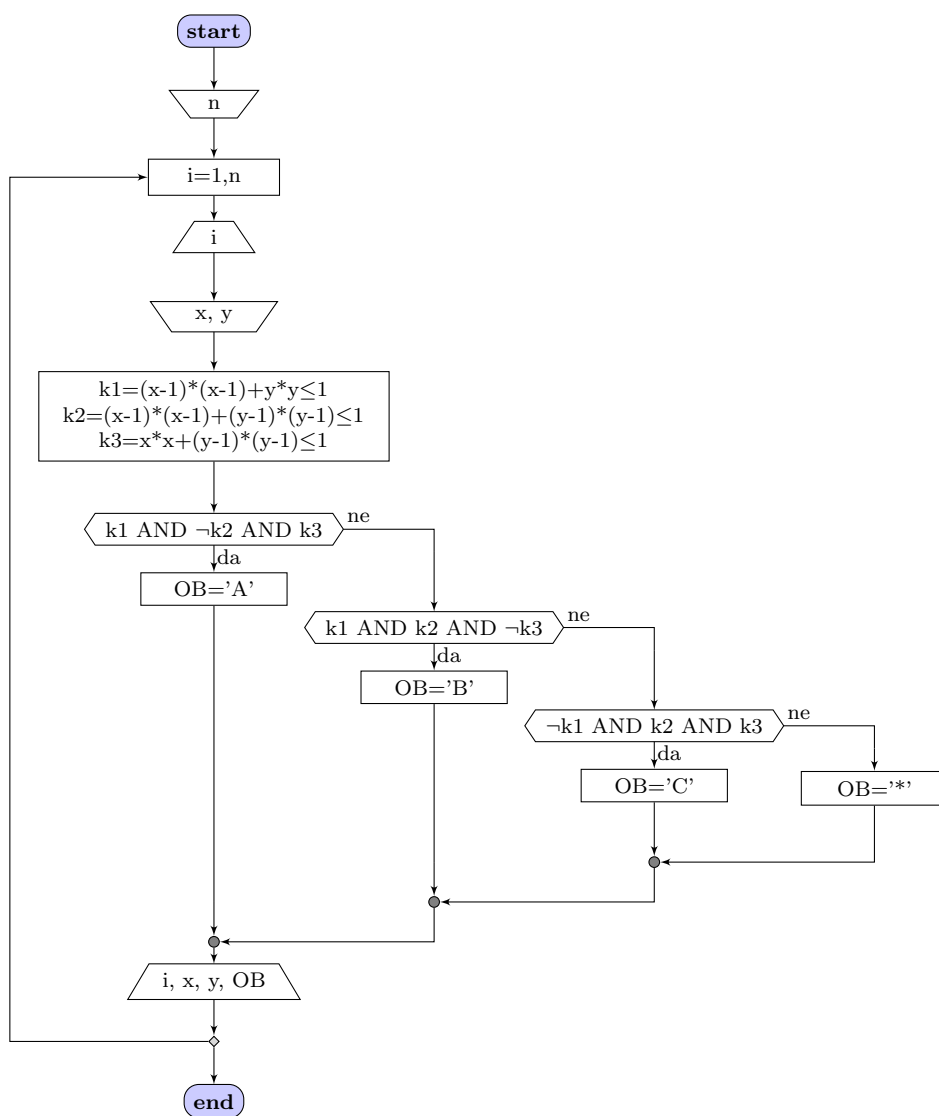
## Rešenje

```

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 koordinatu 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 = '*';

```

Zbirka zadataka

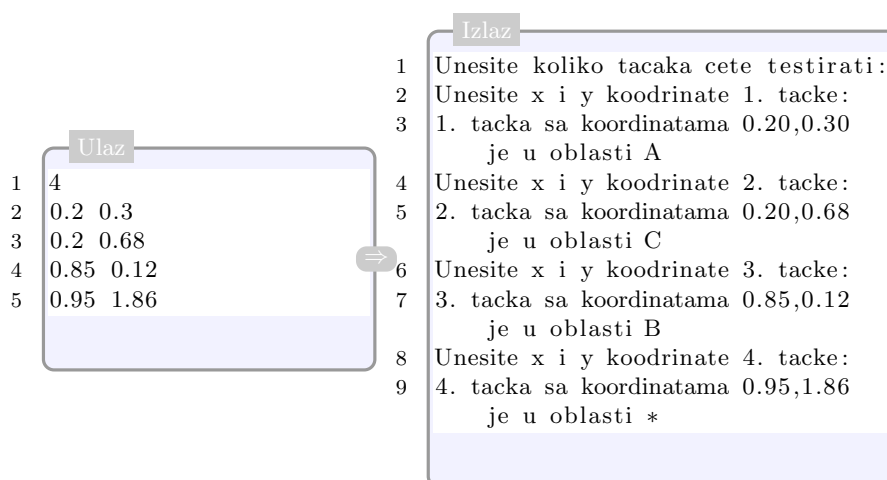


```

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

```

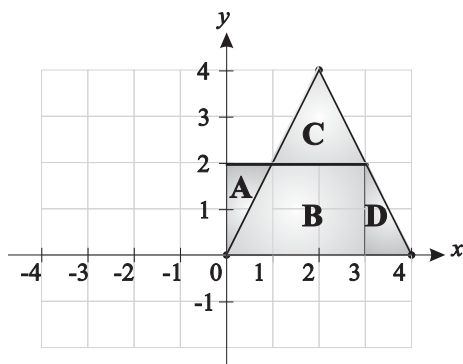
Zbirka zadataka

**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

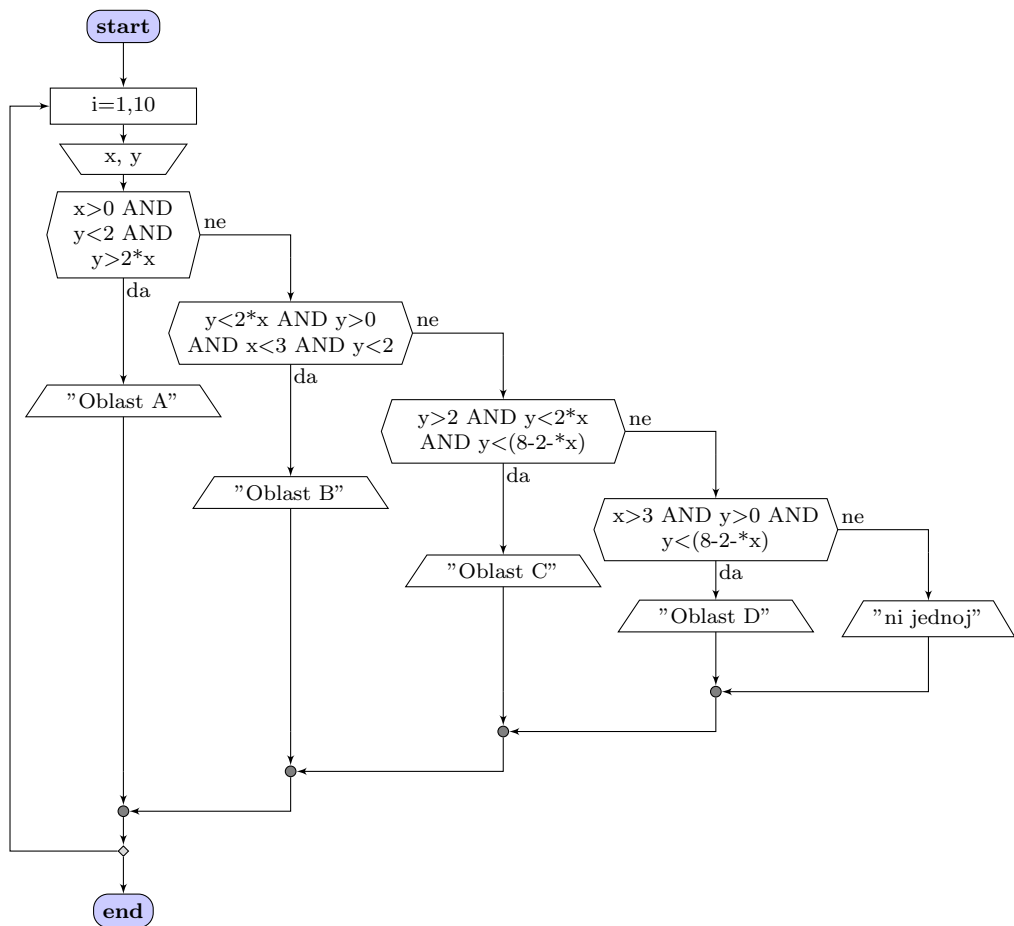
$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1).$$

Napomena: za tačke na pravama proizvoljno odrediti pripadnost oblastima.



Slika 1.4

**Rešenje***Zbirka zadataka*



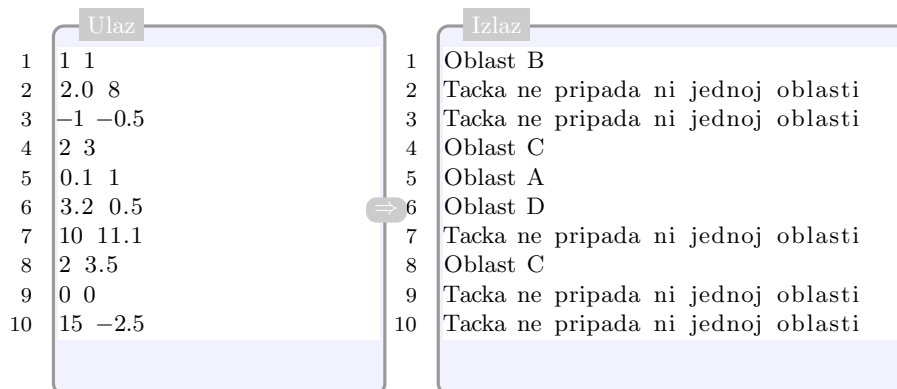
```

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 ("%d%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 zadatka

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







## 2

# Linearne indeksirane strukture podataka

## 2.1 Nizovi

### 2.1.1 Brojevni nizovi

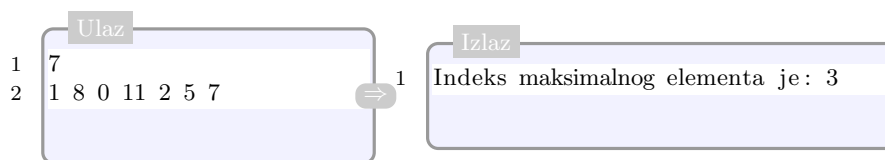
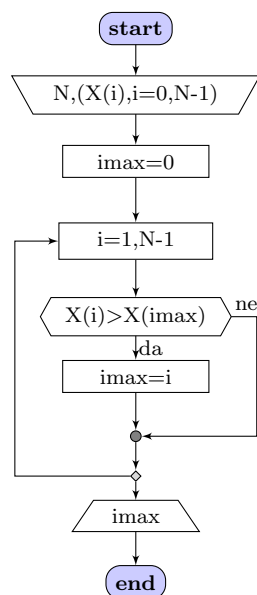
#### Zadatak 2.36

---

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

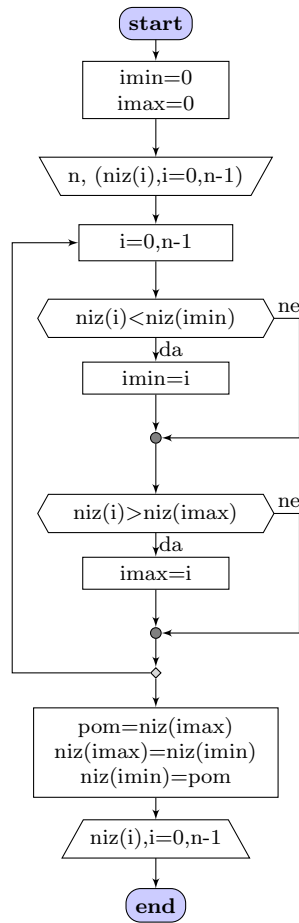
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji vrši zamenu mesta minimalnom i maksimalnom elementu niza. Broj elemenata niza i elemente niza zadaje korisnik. Prikazati niz nakon transformacije.

### 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

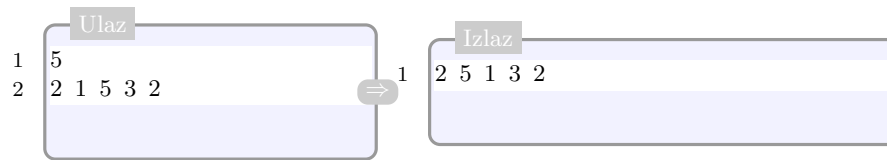


```

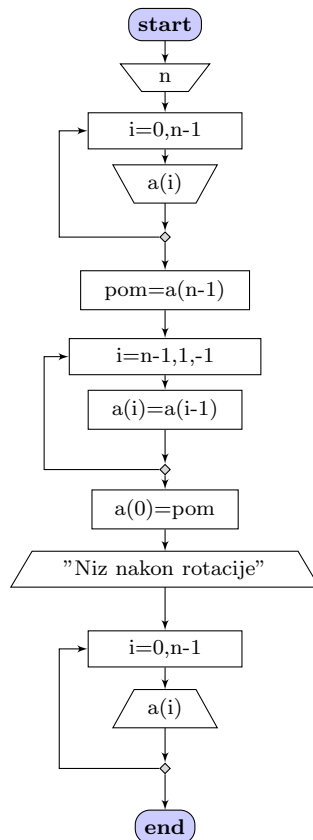
11     {
12         if(niz[i] < niz[imin])
13             imin = i;
14         if(niz[i] > niz[imax])
15             imax = i;
16     }
17     pom = niz[imax];
18     niz[imax] = niz[imin];
19     niz[imin] = pom;
20     for(i=0; i<n; i++)
21         printf("%d\\n", niz[i]);
22 }

```

Zbirka zadataka

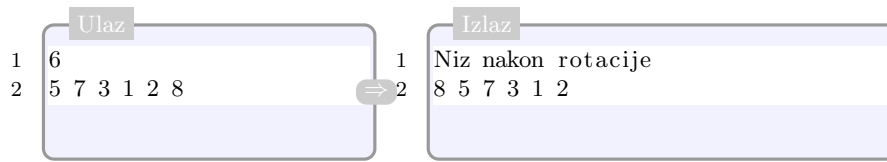
**Zadatak 2.38**

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.

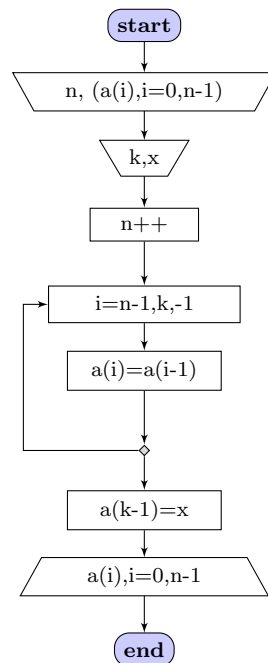
**Rešenje**

```

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 rotacije\n");
13     for (i=0; i<n; i++)
14         printf ("%d", a[i]);
15 }
  
```

**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**

```

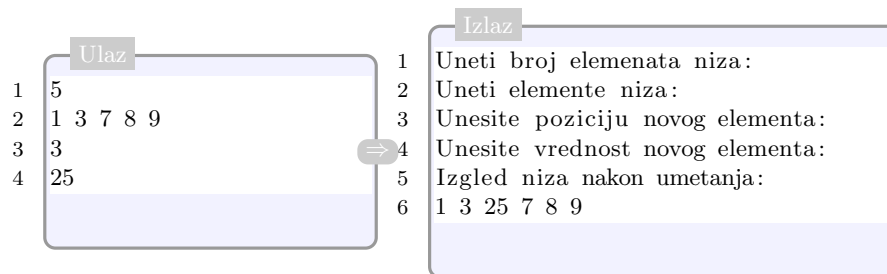
1 #include <stdio.h>
2 void main()
3 {
4     int n, i, x, a[10], k;
5     printf("Uneti broj elemenata niza:\n");
6     scanf("%d", &n);
7     printf("Uneti elemente niza:\n");
8     for (i=0; i<n; i++)
  
```

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    }

```



#### 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.

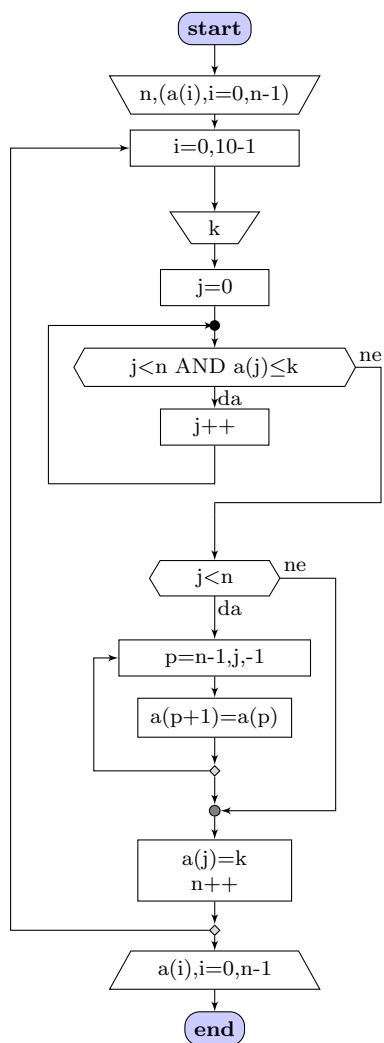
#### Rešenje

```

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

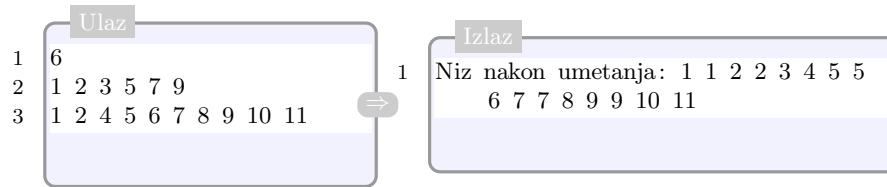


```

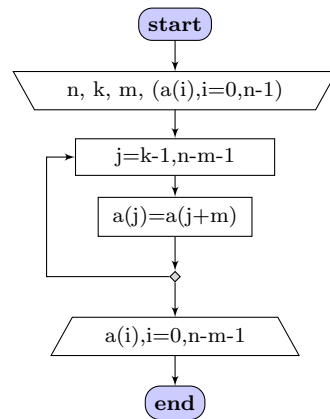
14         if ( j < n )
15             for ( p = n-1; p >= j; p-- )
16                 a[p+1] = a[p];
17         a[j] = k;
18         n++;
19     }
20     printf ("Niznakon umetanja: \n");
21     for ( i = 0; i < n; i++ )
22         printf ("%d\n", a[i]);
23 }

```

Zbirka zadataka

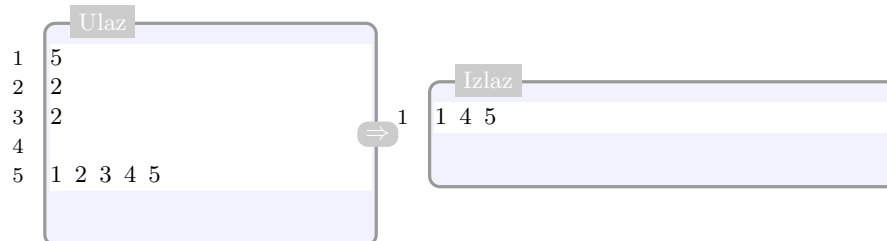
**Zadatak 2.41**

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**

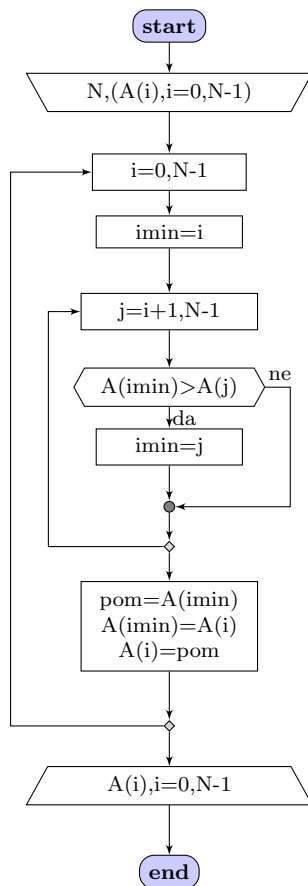
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji niz  $A_N$  uređuje u rastući redosled metodom selekcije (*selection*).

*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])

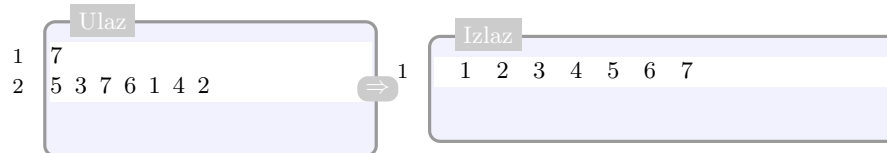
```

Zbirka zadataka

```

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.

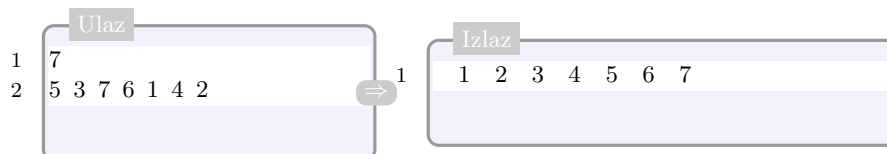
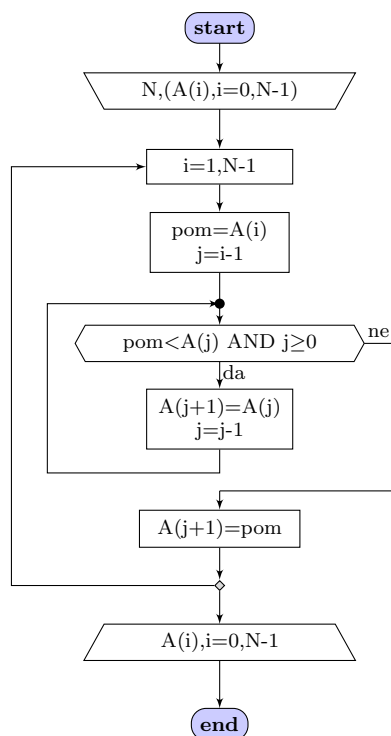
#### Rešenje

```

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.

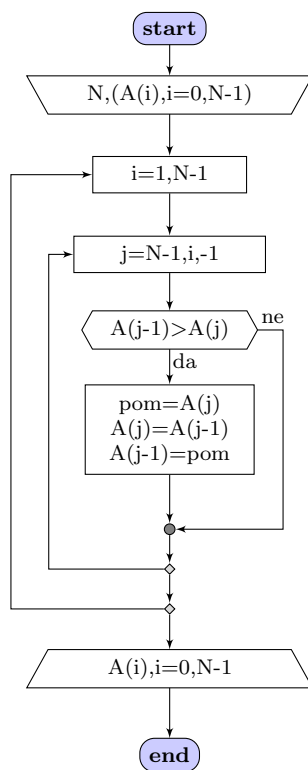
**Rešenje**

```

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

```

Zbirka zadataka

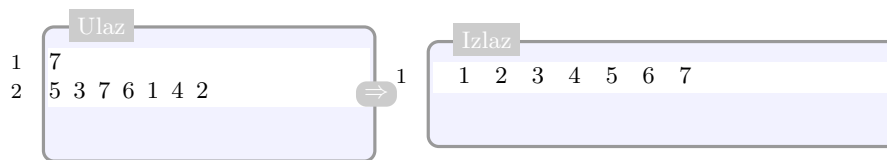


```

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("%3d",A[i]);
18 }

```

Zbirka zadataka

**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 učitano broj, ukloniti iz niza element koji je jednak učitano broj, 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.

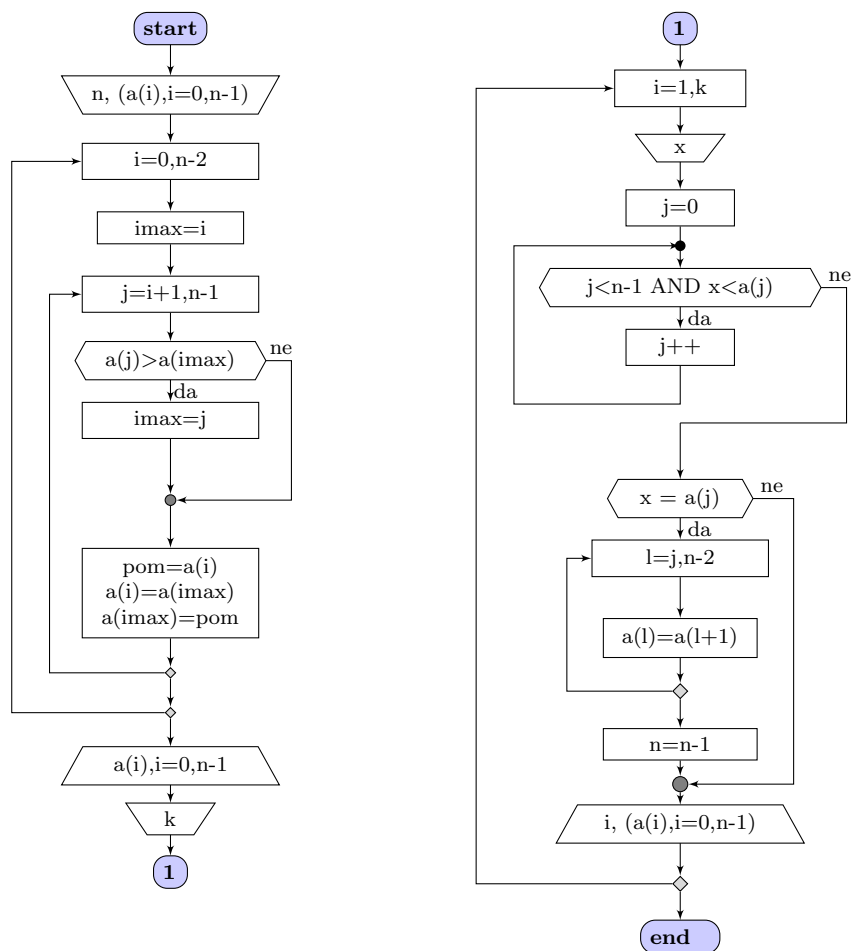
**Rešenje**

```

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] )

```

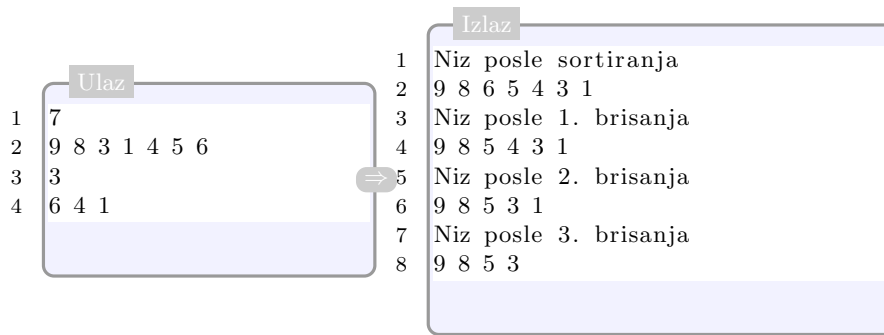
Zbirka zadataka



```

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

Zbirka zadataka

**Zadatak 2.46**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati program kojim se u celobrojnom nizu  $A$  od  $N$  elemenata pronalaze 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.

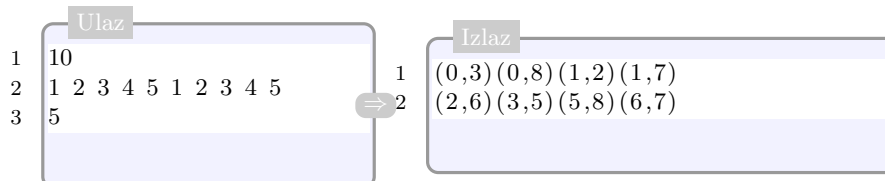
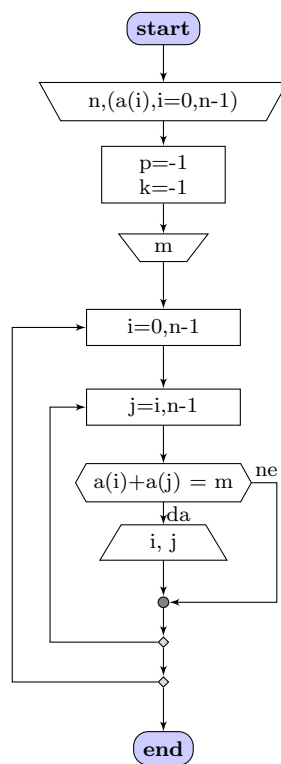
**Rešenje**

```

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

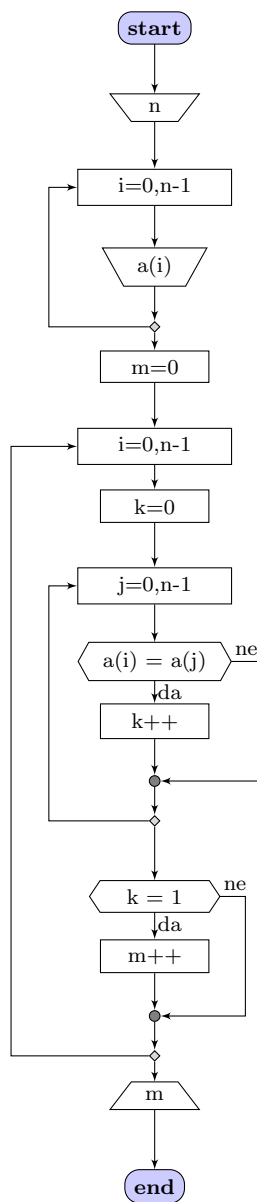
**Zadatak 2.47**

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**

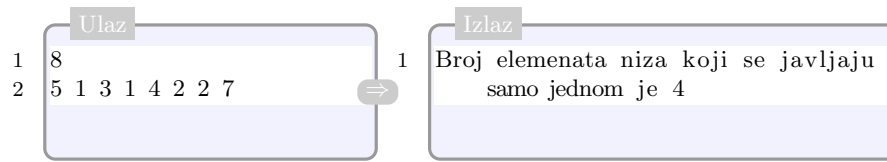
Zbirka zadataka





```

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 niza koji se
19    javljaju samo jednom je %d",m);
  
```

**Zadatak 2.48**

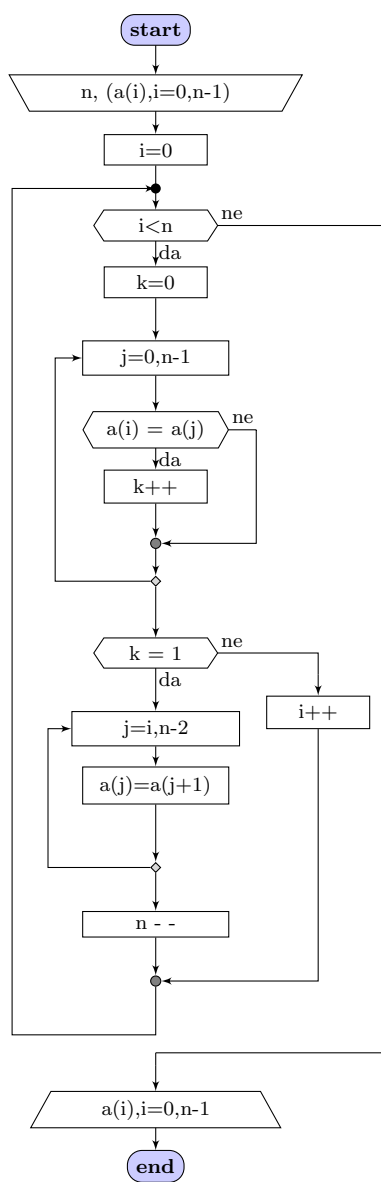
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje i briše elemente niza koji se javljaju samo jednom. Nakon brisanja svakog elementa obezbediti da se novonastali "prazan prostor" popuni pomeranjem odgovarajućih elemenata niza. Prikazati niz 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("%2d", a[i]);
30     }
31 }
  
```

Zbirka zadataka



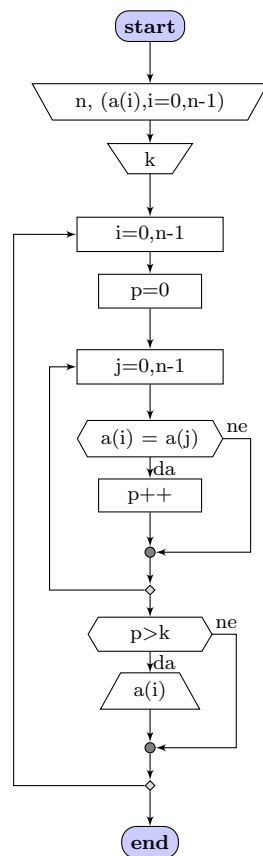
| Ulaz |                     | Izlaz |                                      |
|------|---------------------|-------|--------------------------------------|
| 1    | 10                  | 1     | Unesite broj elemenata niza:         |
| 2    | 1 2 1 5 3 8 3 7 4 1 | 2     | Unesite elemente niza:               |
| ⇒    |                     | 3     | Niz nakon izbacivanja elemenata: 1 1 |
|      |                     |       | 3 3 1                                |

Zbirka zadataka

## Zadatak 2.49

Nacrtati strukturni dijagram toka algoritma i na programsku jeziku C napisati program koji u celobrojnem 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



```

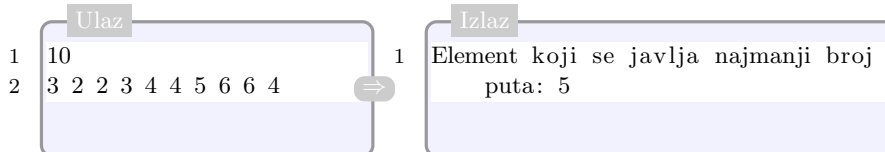
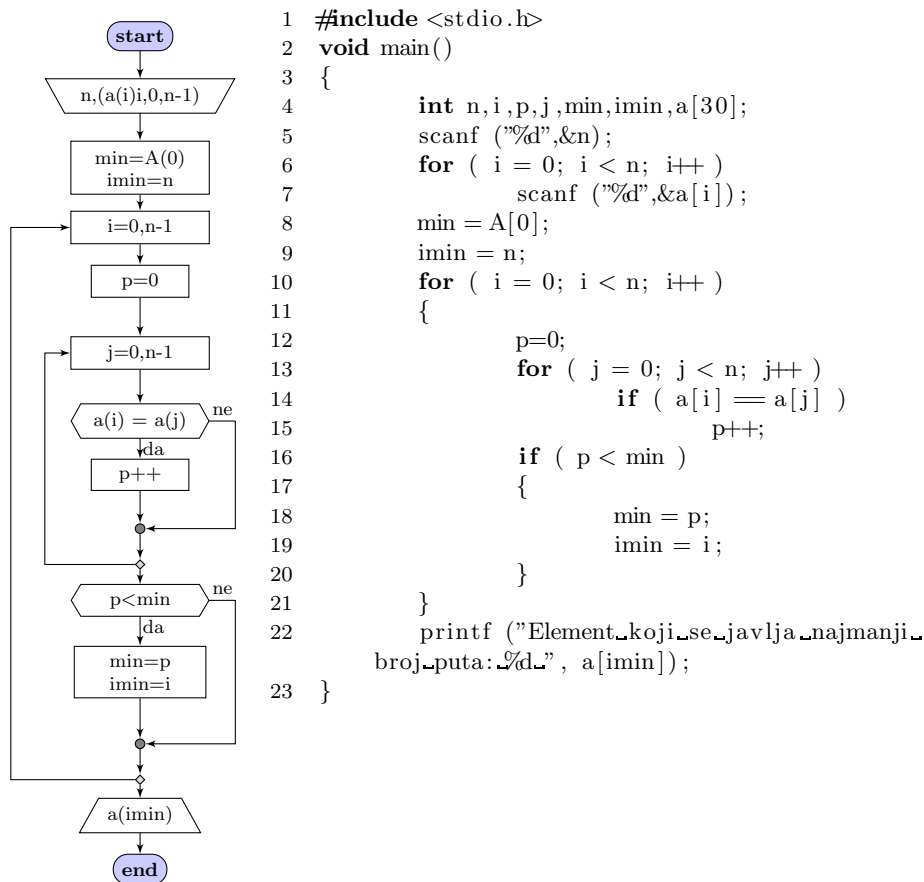
1  #include <stdio.h>
2  void main()
3  {
4      int i,n,k,j,p,a[30];
5      scanf ("%d",&n);
6      for ( i = 0; i < n; i++ )
7          scanf ("%d",&a[i]);
8      scanf ("%d",&k);
9      printf ("Elementi niza koji se javljaju
vise od 1 puta su:\n",k);
10     for ( i = 0; i < n; i++ )
11     {
12 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

| Ulaz |                     |
|------|---------------------|
| 1    | 10                  |
| 2    | 1 1 2 3 4 5 5 6 7 8 |
| 3    | 1                   |

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

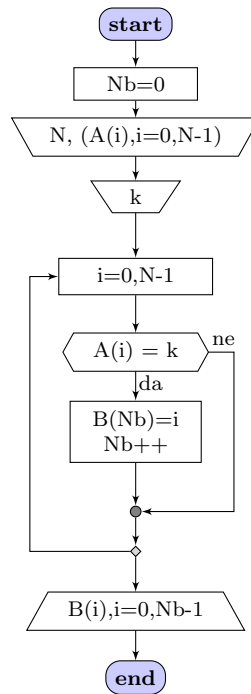
**Zadatak 2.50**

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

**Rešenje**

**Zadatak 2.51**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje pozicije (indekse) na kojima se zadata vrednost  $k$  javlja u nizu  $A$  od  $N$  elemenata. Pozicije na kojima se zadata vrednost javlja upisati u pomoćni 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++)

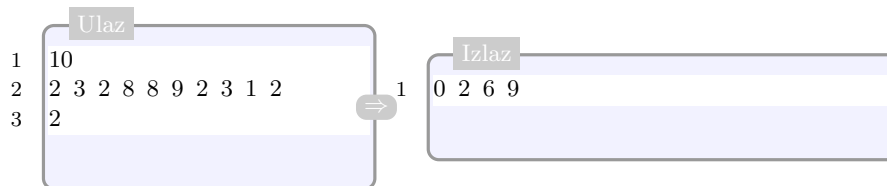
```

Zbirka zadataka

```

16         printf("%d\\n",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 elemenata 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

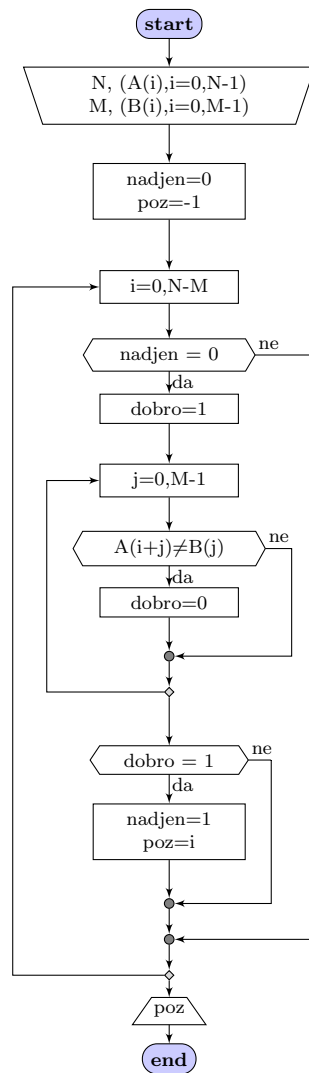
prvi način, bez optimizacije izvršenja

```

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

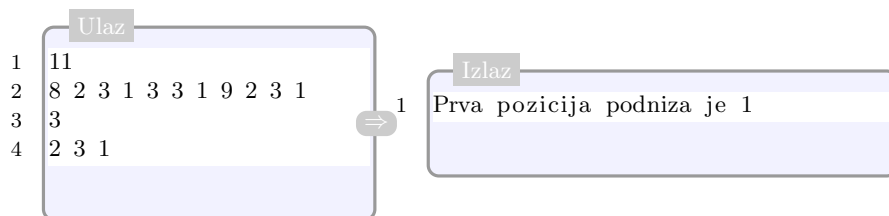


```

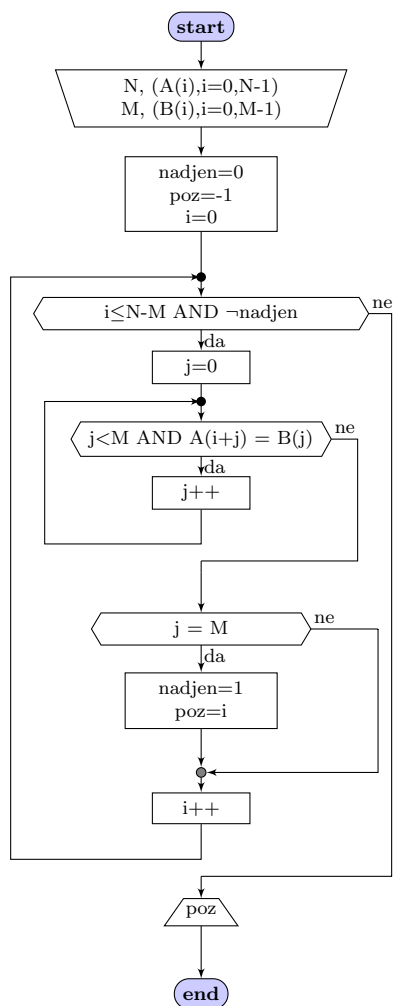
28         poz=i ;
29     }
30 }
31 }
32 printf("Prva pozicija podniza je %d\n", poz);
33 }

```





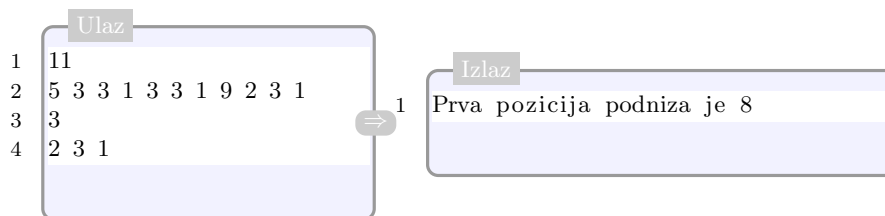
drugi način, sa optimizacijom izvršenja



```

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-M && !nadjen) //od pocetka do poslednjeg indeksa gde
    bi mogao da pocne podniz ALI samo dok "nadjen" ne postane 1
19     {
20         j=0;
21         while(j<M && A[i+j]!=B[j])
22             j++;
23
24         if(j==M)
25         {
26             nadjen=1;
27             poz=i;
28         }
29         i++;
30     }
31
32     printf("Prva pozicija podniza je %d\n", poz);
33 }

```



#### Zadatak 2.53

Nacrtati strukturni dijagram toka algoritma i napisati strukturni program na jeziku C koji smešta 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*

elemenata 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 elemente niza sedišta pre i nakon smeštanja novog posetioca.

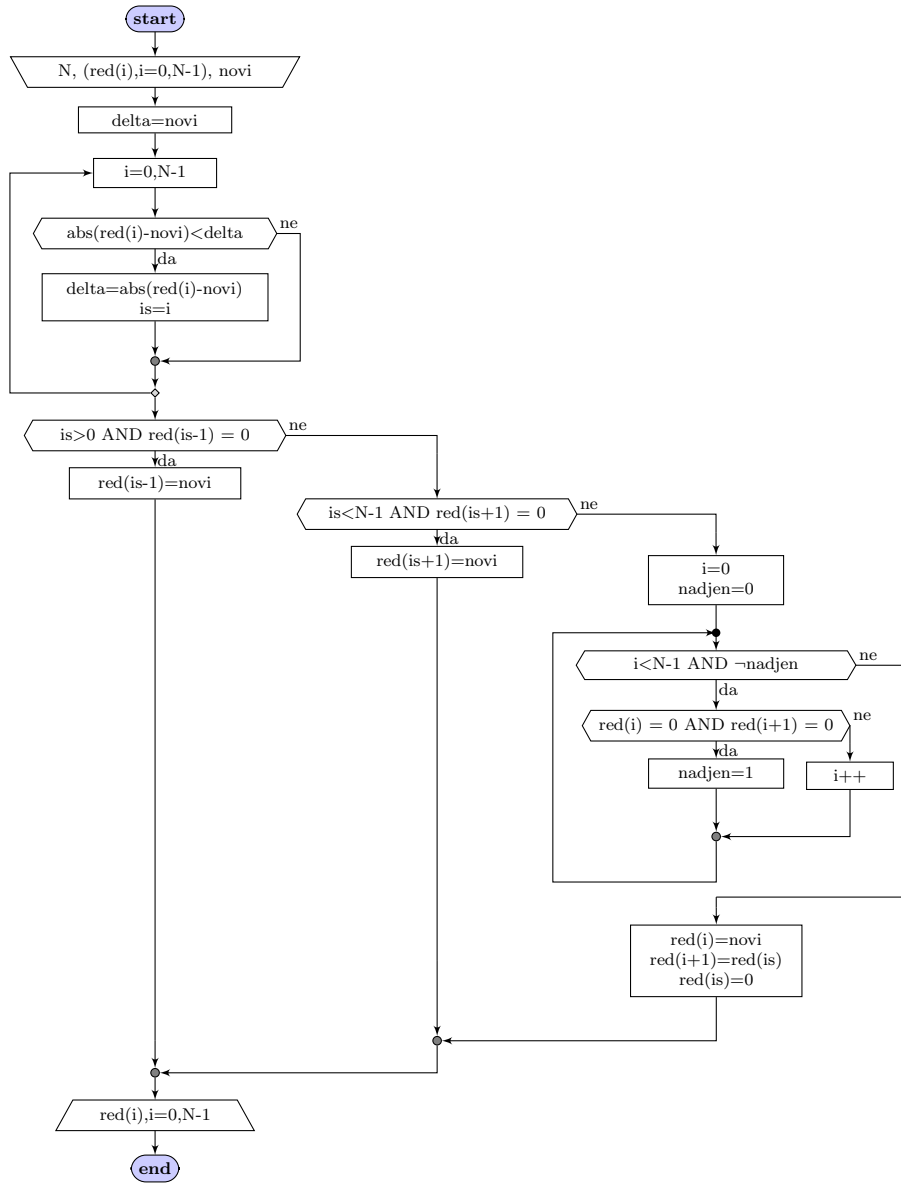
### Rešenje

```

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++)

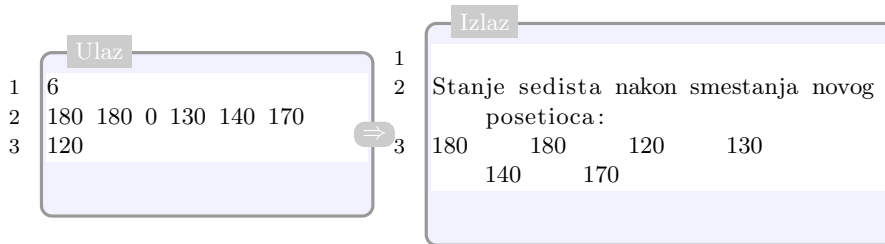
```

Zbirka zadataka

```

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

```



### Operacije nad skupovima predstavljnim nizovima

#### Zadatak 2.54

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

- presek skupova  $C = A \cap B$
- 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

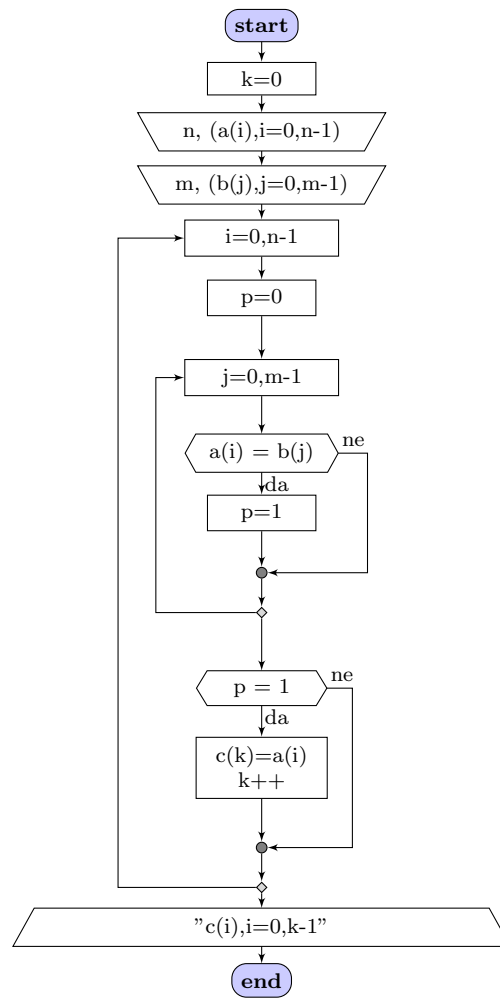
a)

```

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 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 == 1 )
20         {
21             c[k]=a[i];

```

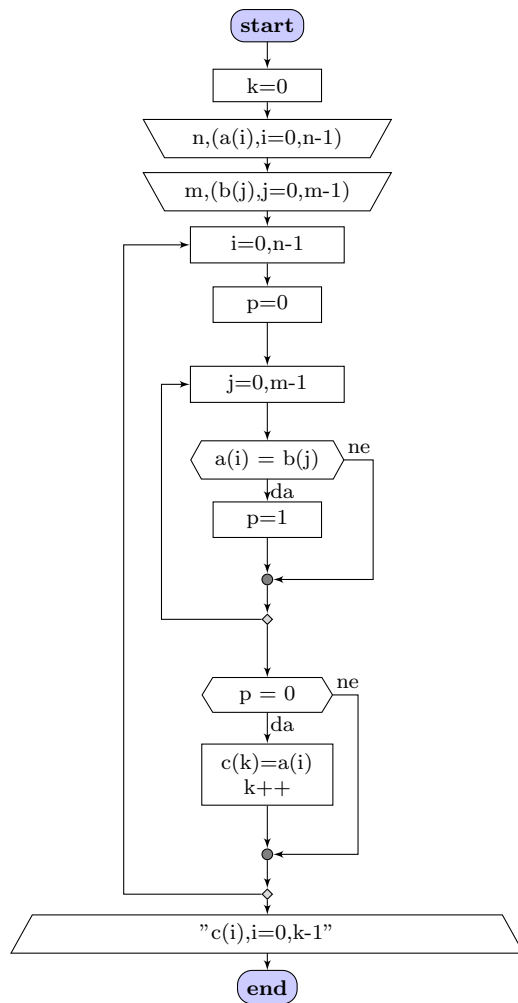
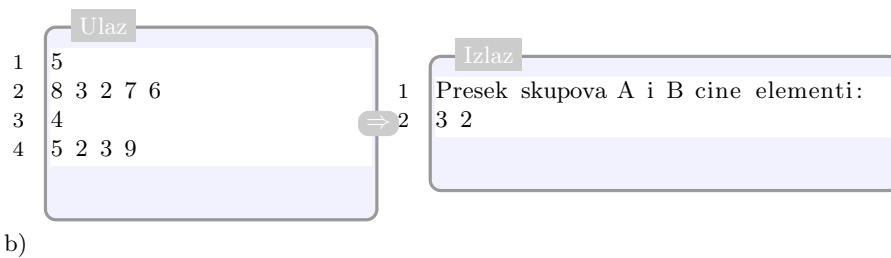
Zbirka zadataka



```

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 }

```



```

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

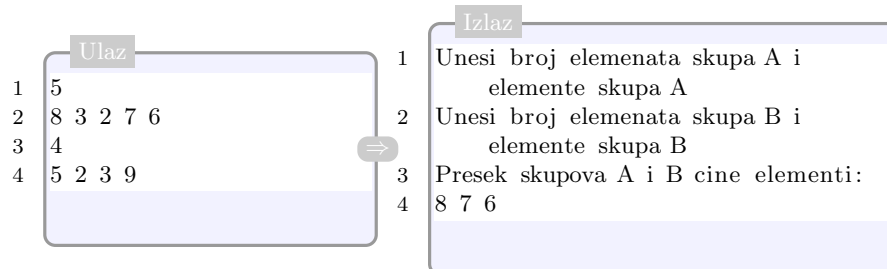
```

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 }

```

**Zadatak 2.55**

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

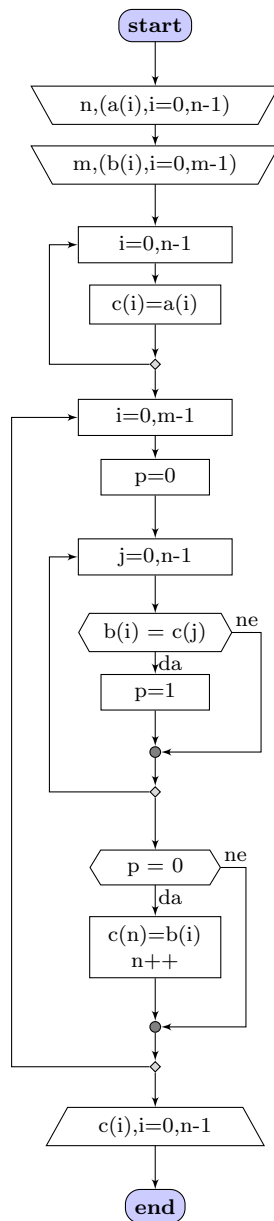
$$C = A \cup B.$$

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];

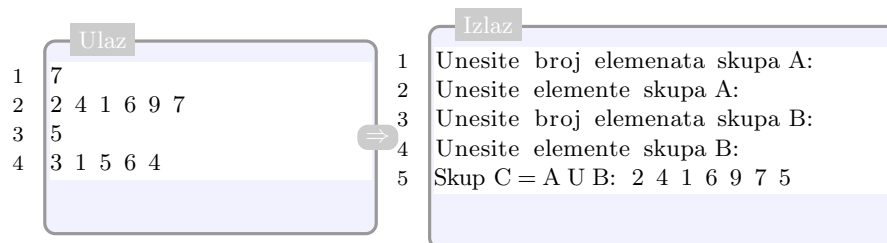
```

*Zbirka zadataka*

```

5     printf("Unesite broj elemenata skupa A:\n");
6     scanf("%d", &n);
7     printf("Unesite elemente skupa A:\n");
8     for(i=0; i<n; i++)
9         scanf("%d", &a[i]);
10    printf("Unesite broj elemenata skupa B:\n");
11    scanf("%d", &m);
12    printf("Unesite elemente 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("%2d", c[i]);
32 }

```



### Zadatak 2.56

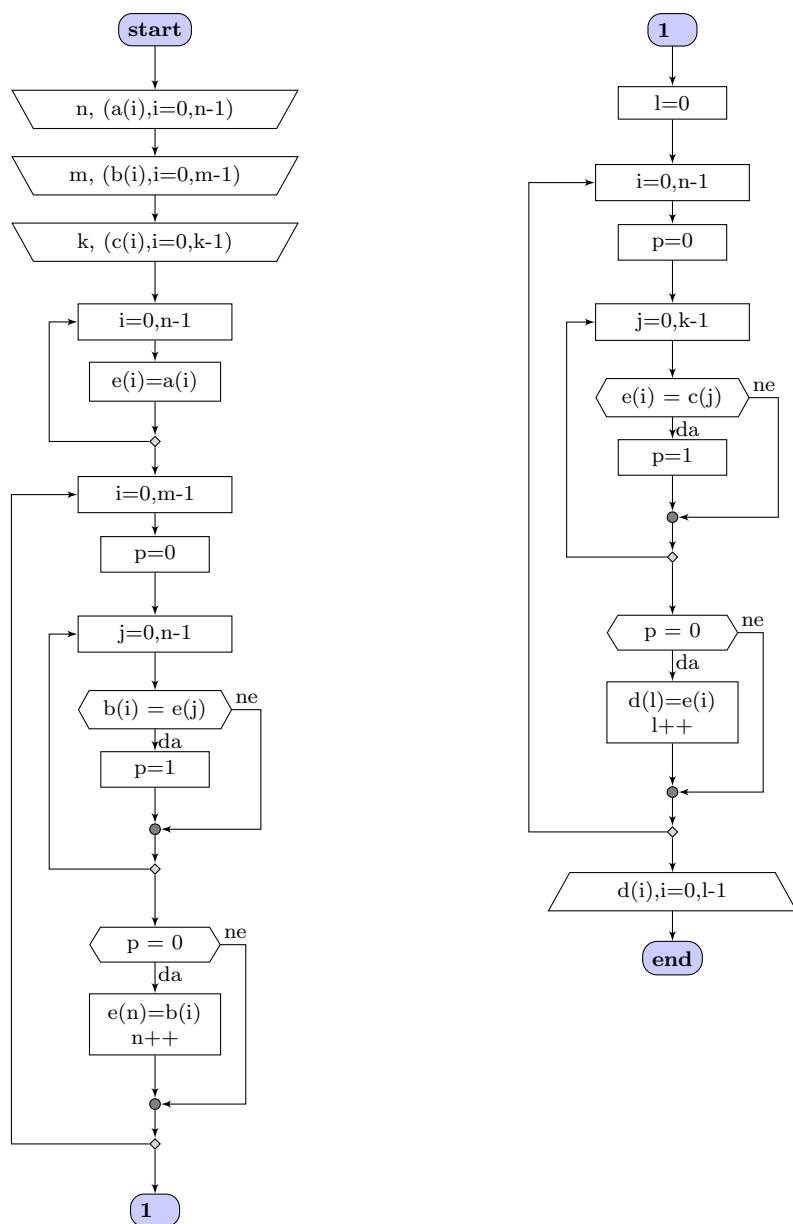
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje skup

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

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 {

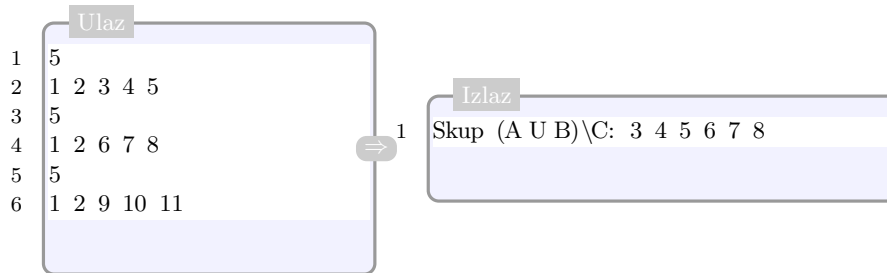
```

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 ∪ B) \\ C: ");
42     for ( i = 0; i < l; i++ )
43         printf ("%d",d[i]);
44 }

```

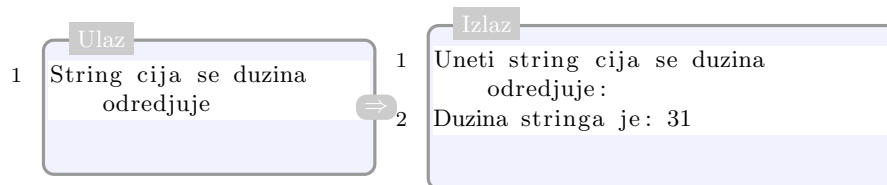
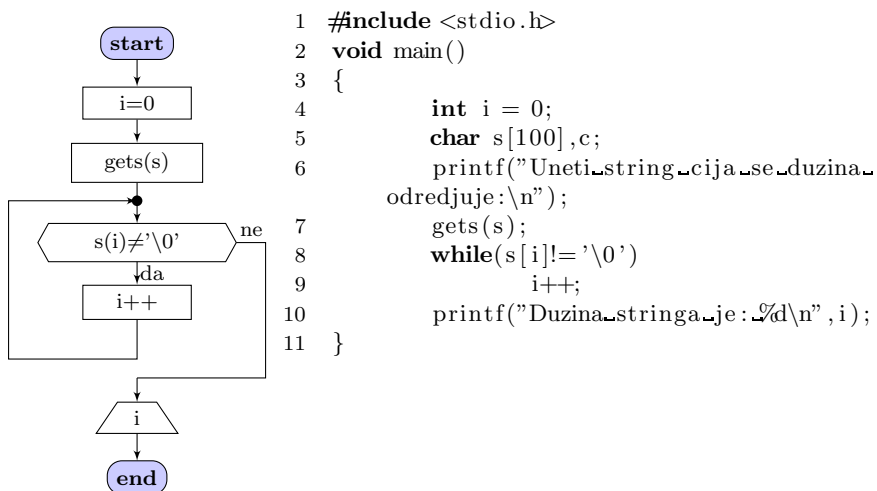


### 2.1.2 Znakovni nizovi - stringovi

#### Zadatak 2.57

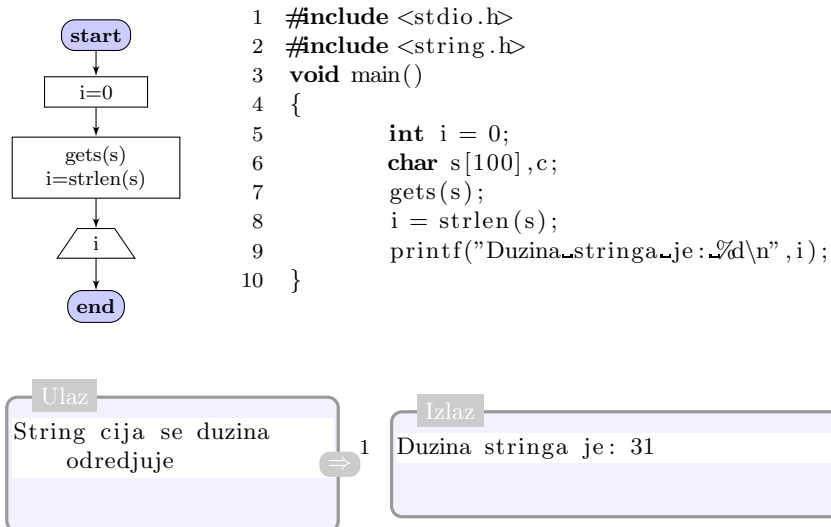
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje dužinu unetog stringa bez korišćenja bibliotekskih funkcija.

#### Rešenje



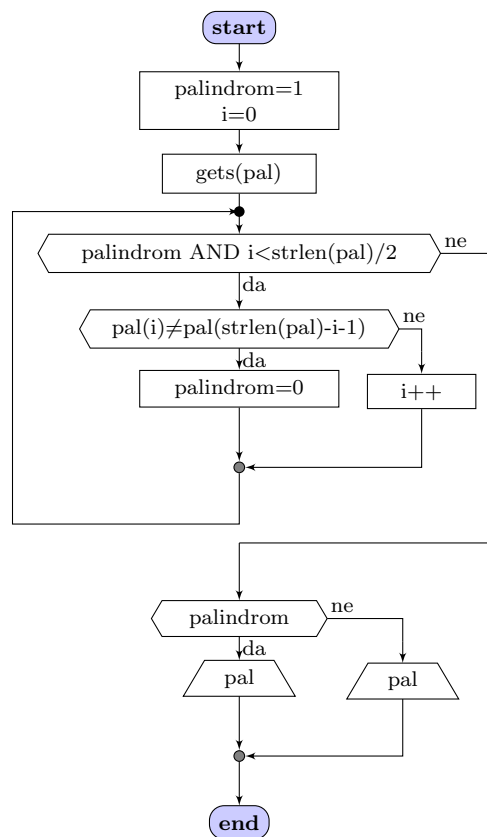
**Zadatak 2.58**

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

**Rešenje****Zadatak 2.59**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni 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**

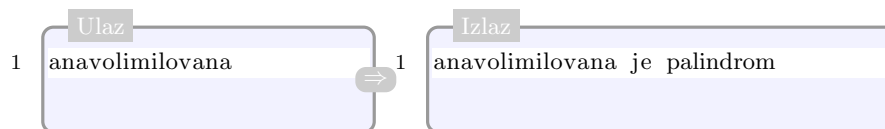


```

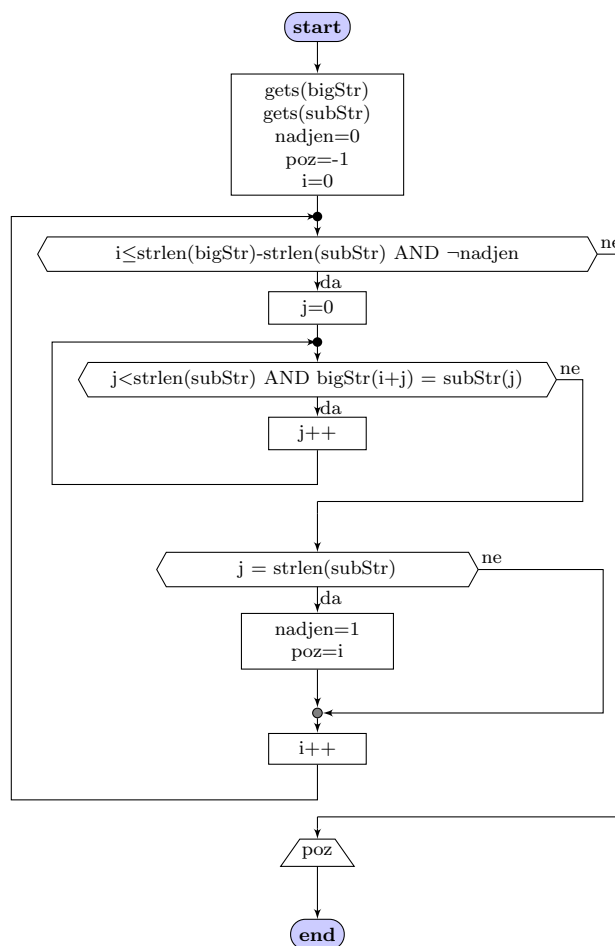
1  #include <stdio.h>
2  #include <string.h>
3  void main()
4  {
5      char pal[50];
6      int palindrom = 1, i = 0;
7      gets(pal);
8      while ( palindrom && i < strlen(pal) / 2 )
9      {
10         if ( pal[i] != pal[strlen(pal)-i-1] )
11             palindrom = 0;
12         else
13             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**

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

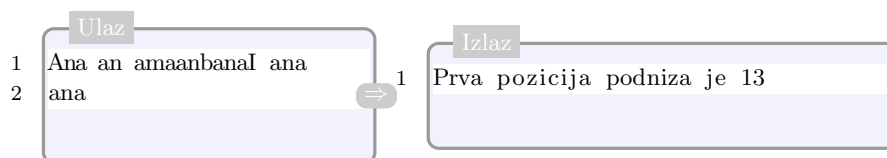
**Rešenje**



```

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 }

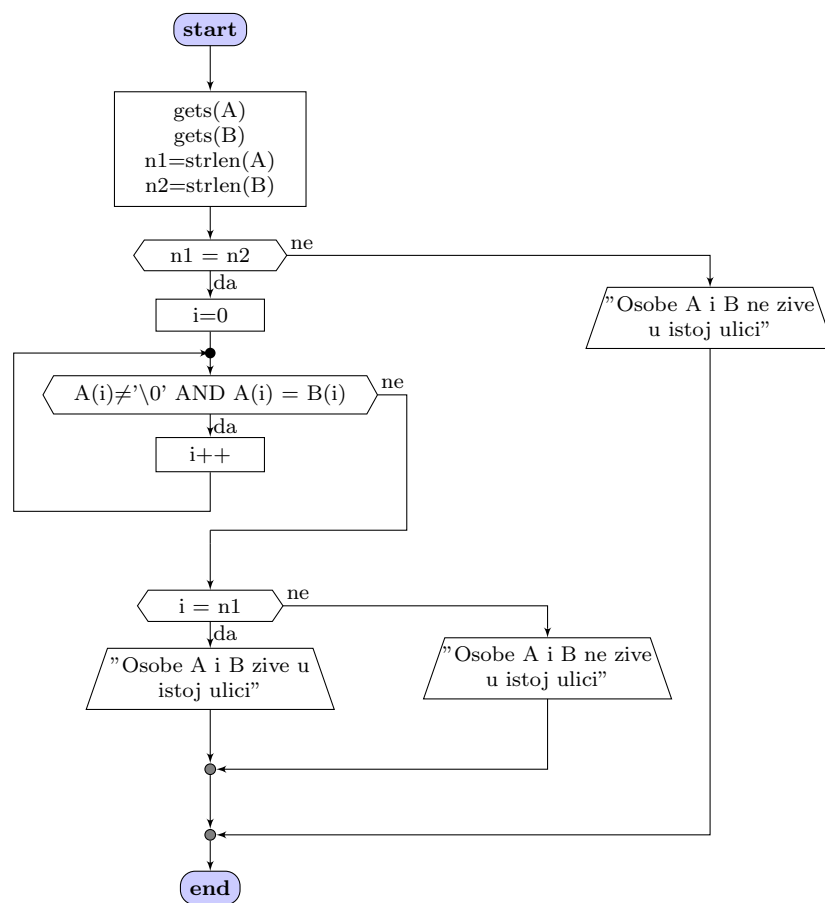
```

**Zadatak 2.61**

Nacrtaji 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



```

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

```

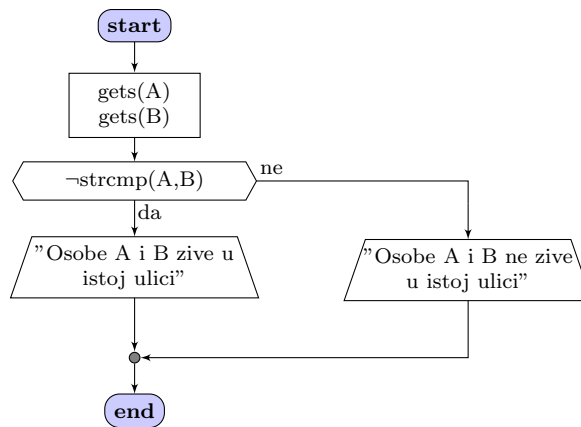
Zbirka zadataka

```

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

```

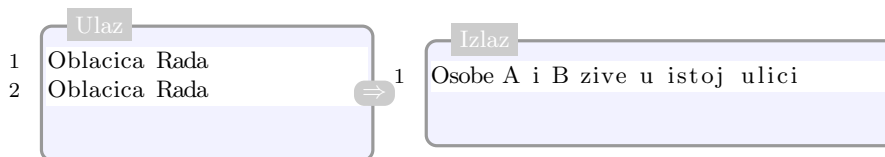
drugi način



```

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 }

```



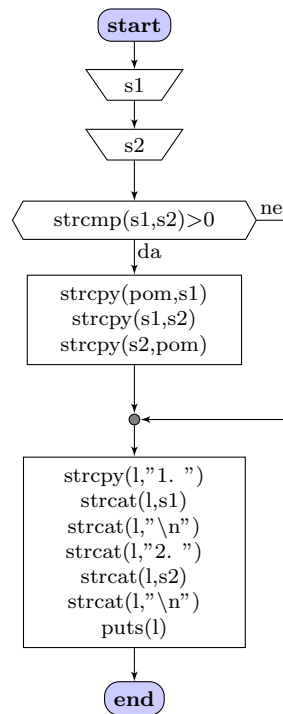
#### Zadatak 2.62

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



```

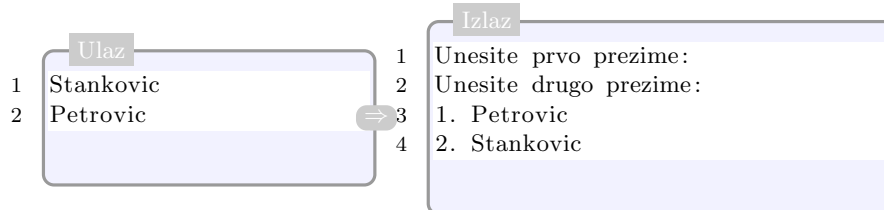
1  #include <stdio.h>
2  #include <string.h>
3
4  void main()
5  {
6      char s1[25], s2[25], l[100], pom[25];
7      printf("Unesite prvo prezime:\n");
8      gets(s1);
9      printf("Unesite drugo prezime:\n");
10     gets(s2);
11     if ( strcmp(s1, s2) > 0 )
12     {
13         strcpy(pom, s1);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

Zbirka zadataka

```

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 više). Prikazati redni broj najduže reči.

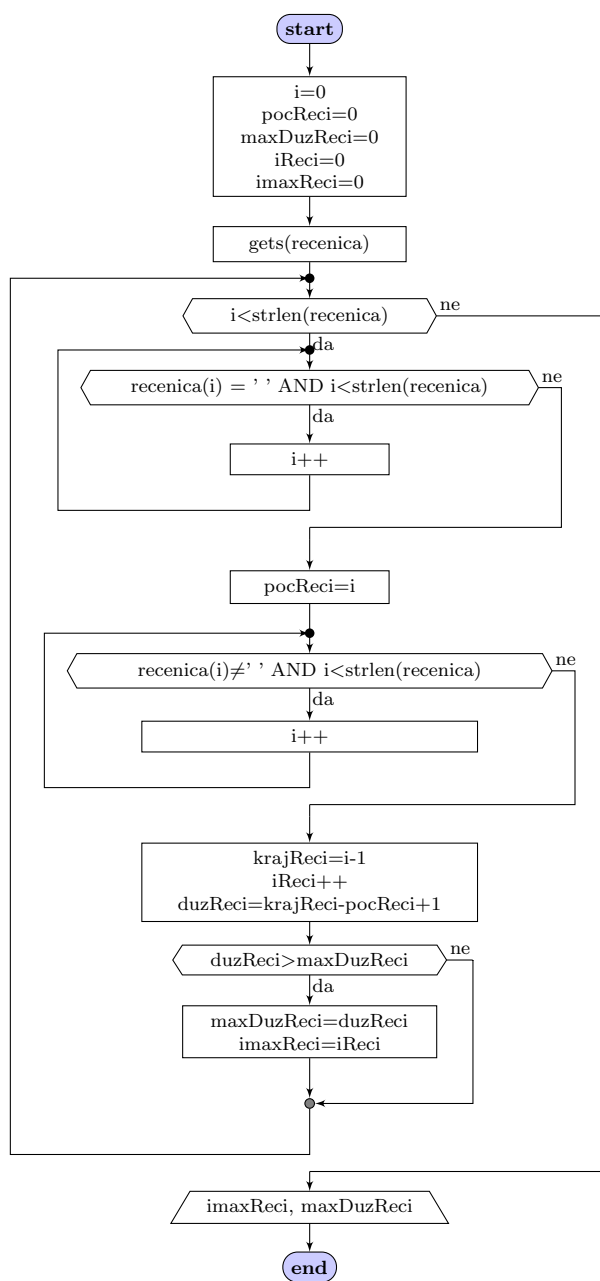
### Rešenje

```

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
    = 0, imaxReci = 0;
7      gets (recenica);
8      while ( i < strlen ( recenica ) )
9      {
10         while ( recenica[i] == ' ' && i < strlen ( recenica ) )
11             i++;
12         pocReci = i;
13         while ( recenica[i] != ' ' && i < strlen ( recenica ) )
14             i++;
15         krajReci = i - 1;
16         iReci++;
17         duzReci = krajReci - pocReci + 1;
18         if ( duzReci > maxDuzReci )
19         {
20             maxDuzReci = duzReci;
21             imaxReci = iReci;
22         }

```

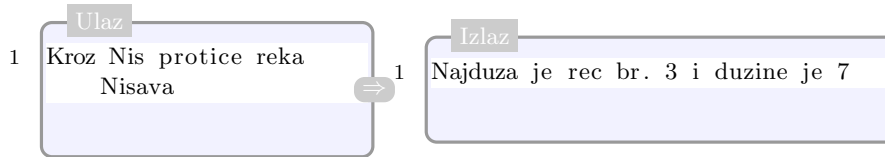
Zbirka zadataka



```

24     printf ("Najduza_je_rec_br. %d i duzine_je %d",
25     imaxReci, maxDuzReci);

```



## 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

3. vrste,
2. kolone,
- na glavnoj dijagonali,
- na sporednoj dijagonali,
- iznad glavne dijagonale,
- ispod sporedne dijagonale.

### Rešenje

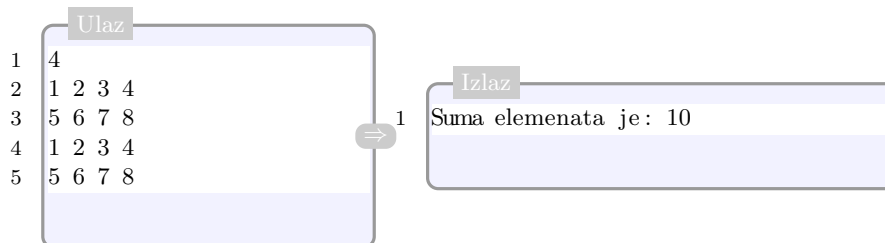
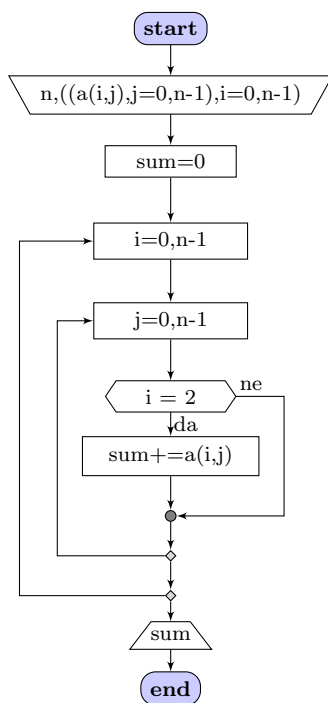
a)

```

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 == j )
13                 sum += a[i][j];
14     printf("Suma elemenata je: %d\n", sum);
15 }

```

Zbirka zadataka



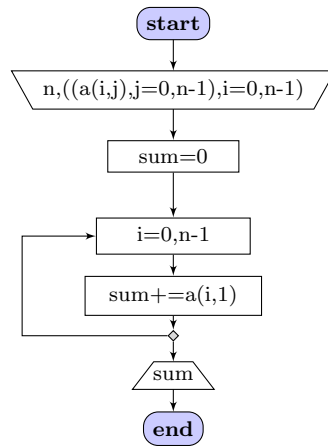
b)

```

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

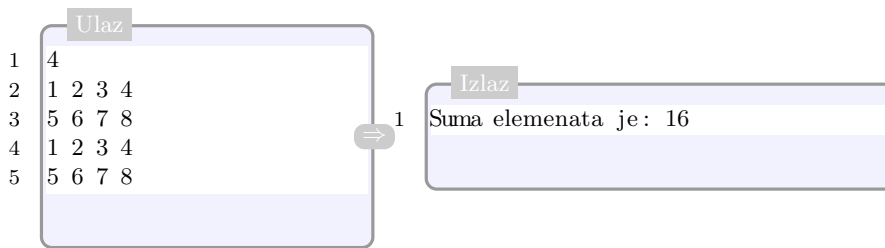




```

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

```



c)

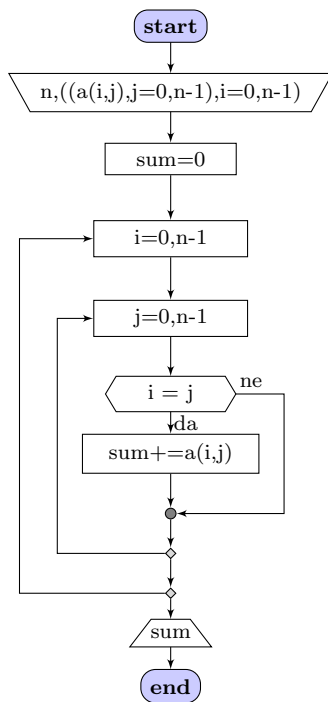
```

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];

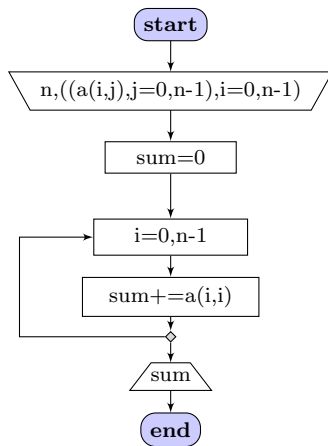
```

Zbirka zadataka

varijanta 1



varijanta 2

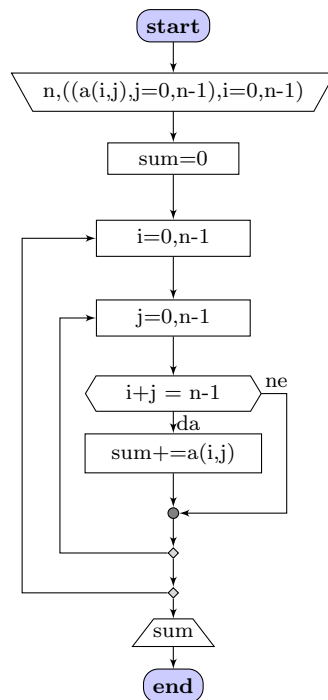


```

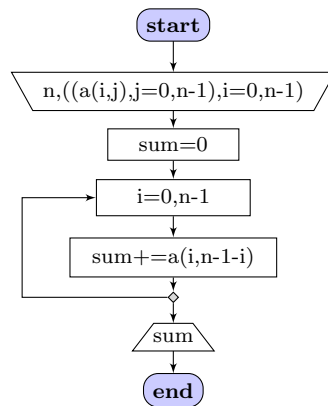
14     printf("Suma elemenata je: %d\n", sum);
15 }
  
```

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);
  
```

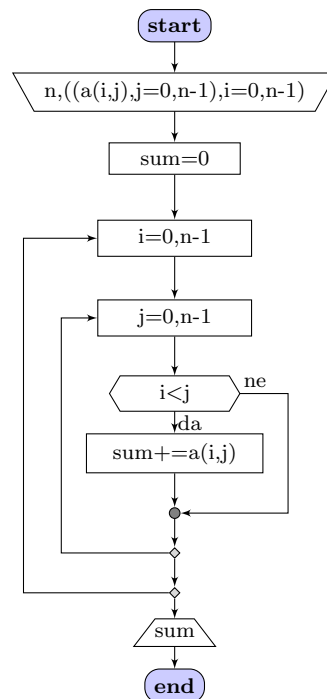
Zbirka zadataka

```

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]);

```

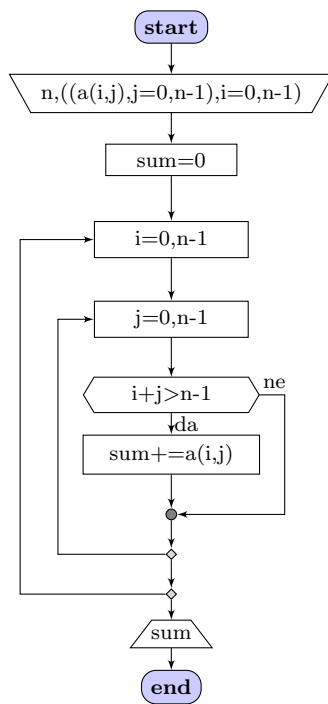
Zbirka zadataka

```

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

```

f)



```

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++)

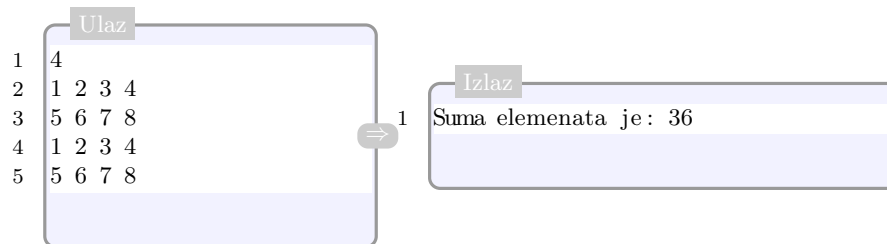
```

Zbirka zadataka

```

12         if( i+j > n-1)
13             sum+=a[i][j];
14     printf("Suma elemenata je: %d\n", sum);
15 }

```



### Zadatak 2.65

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje najveći element u celobrojnoj matrici dimenzija  $m \times n$ . Predvideti unos vrednosti dimenzija matrice  $m$  i  $n$ , i vrednosti elemenata matrice. Prikazati maksimalni element, kao i indeks vrste i indeks kolone kojoj on pripada.

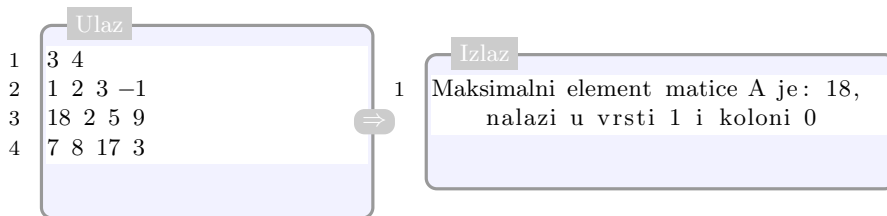
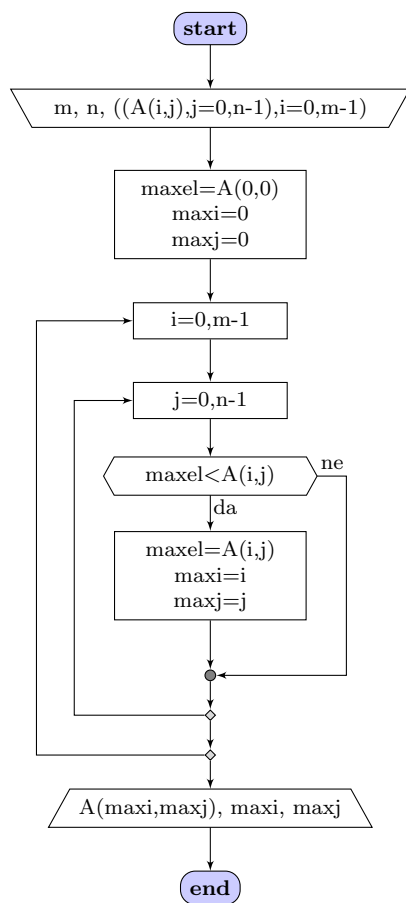
### Rešenje

```

1  #include <stdio.h>
2  main()
3  {
4      int A[100][100], m, n, maxel, maxi, maxj, i, j;
5      scanf("%d %d", &m, &n);
6      for ( i=0; i<n; i++ )
7          for ( j=0; j<n; j++ )
8              scanf( "%d", &A[i][j] );
9
10     maxel=A[0][0];
11     maxi=0;
12     maxj=0;
13     for ( i=0; i<n; i++ )
14         for ( j=0; j<n; j++ )
15             if (maxel < A[i][j])
16                 {
17                     maxel=A[i][j];
18                     maxi=i;
19                     maxj=j;
20                 }
21
22     printf( "Maksimalni element matrice A je: %d, nalazi u vrsti %d i u koloni %d\n", A[maxi][maxj], maxi, maxj );
23 }

```

Zbirka zadataka

**Zadatak 2.66**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program za sabiranje matrica

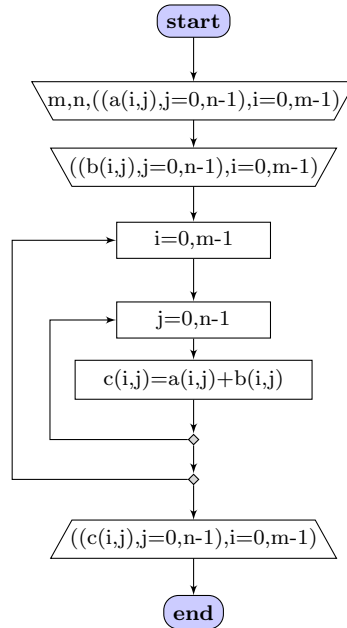
$$C_{N \times M} = A_{N \times M} + B_{N \times M}.$$

Zbirka zadataka

Dimenzije matrica i vrednosti elemenata matrica  $A$  i  $B$  zadaje korisnik.

Napomena: zbir matrica definisan je kao zbir odgovarajućih elemenata, t.j.  $c_{i,j} = a_{i,j} + b_{i,j}$ , za svako  $i = \{0, 1, 2, \dots, N-1\}$  i  $j = \{0, 1, 2, \dots, M-1\}$ .

### Rešenje



```

1  #include<stdio.h>
2  void main()
3  {
4      int i,j,n,m,a[20][20],b[20][20],c[20][20];
5      printf("Unesite broj vrsta matrice:\n");
6      scanf("%d",&m);
7      printf("Unesite broj kolona matrice:\n");
8      scanf("%d",&n);
9      printf("Unesite elemente prve matrice:\n");
10     for(i=0;i<m;i++)
11         for(j=0;j<n;j++)
12             scanf("%d",&a[i][j]);
13     printf("Unesite elemente druge matrice:\n");
14     for(i=0;i<m;i++)
15         for(j=0;j<n;j++)
16             scanf("%d",&b[i][j]);
17     for(i=0;i<m;i++)
18         for(j=0;j<n;j++)
19             c[i][j] = a[i][j] + b[i][j];
20     printf("Prikaz rezultujuće matrice:\n");

```

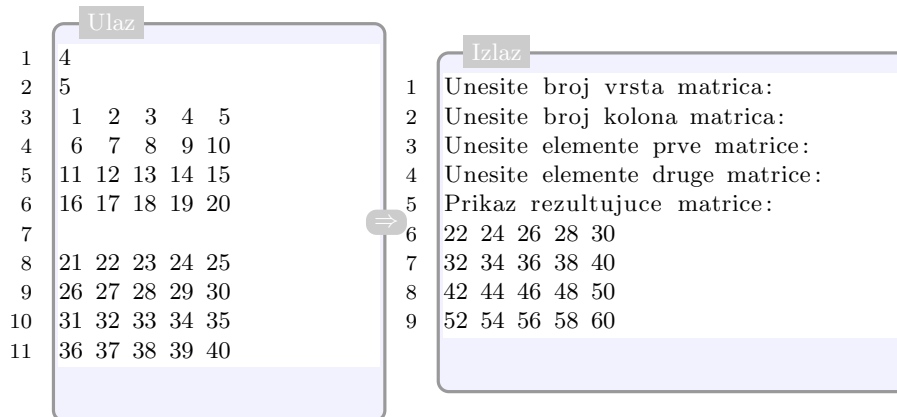
Zbirka zadataka



```

21     for(i=0;i<n; i++)
22     {
23         for(j=0;j<n; j++)
24             printf("%2d ",c[i][j]);
25         printf("\n");
26     }
27 }

```

**Zadatak 2.67**

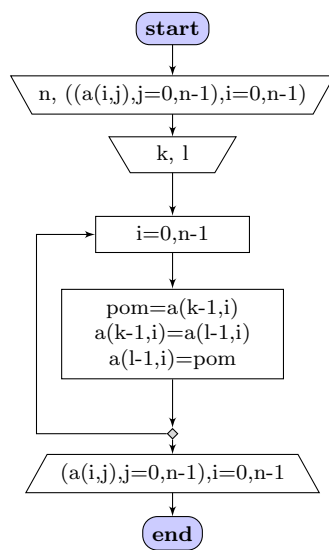
Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji u matrici  $A_{N \times N}$  vrši zamenu mesta elementima

- $k$ -te i  $l$ -te vrste,
- $k$ -te i  $l$ -te kolone,
- $k$ -te vrste i  $k$ -te kolone.

Dimenziju matrice  $N$ , vrednosti elemenata matrice  $a_{i,j}$  i parametre transformacije  $k$  i  $l$  zadaje korisnik. Prikazati matricu nakon transformacije.

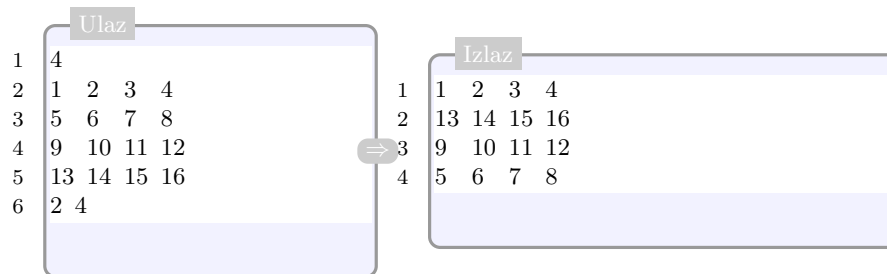
**Rešenje**

a)



```

1 #include <stdio.h>
2 void main()
3 {
4     int i,j,n,a[30][30],k,l,pom;
5     scanf ("%d",&n);
6     for ( i = 0; i < n; i++ )
7         for ( j = 0; j < n; j++ )
8             scanf
9             ("%d",&a[i][j]);
10    scanf ("%d%d",&k,&l);
11    for ( i = 0; i < n; i++ )
12    {
13        pom = a[k-1][i];
14        a[k-1][i] = a[l-1][i];
15        a[l-1][i] = pom;
16    }
17    for ( i = 0; i < n; i++ )
18    {
19        for ( j = 0; j < n; j++ )
20            printf
21            ("%d",a[i][j]);
22    }
  
```

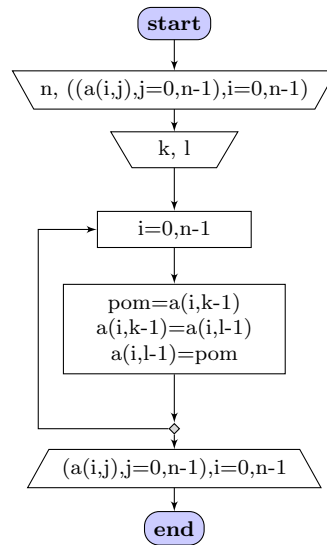


b)

```

1 #include <stdio.h>
2 void main()
3 {
4     int i,j,n,a[30][30],k,l,pom;
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     scanf ("%d%d",&k,&l);
10    for ( i = 0; i < n; i++ )
11    {
  
```

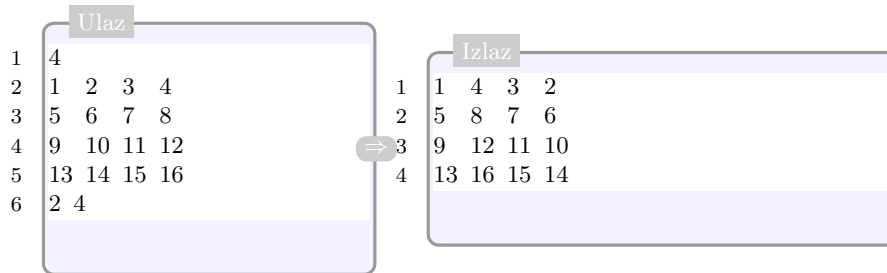
Zbirka zadataka



```

12         pom = a[i][k-1];
13         a[i][k-1] = a[i][l-1];
14         a[i][l-1] = pom;
15     }
16     for ( i = 0; i < n; i++ )
17     {
18         for ( j = 0; j < n; j++ )
19             printf ("%d\\", a[i][j]);
20         printf ("\\n");
21     }
22 }

```



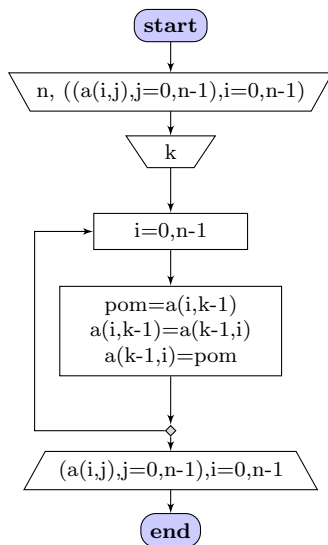
c)

```

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

```

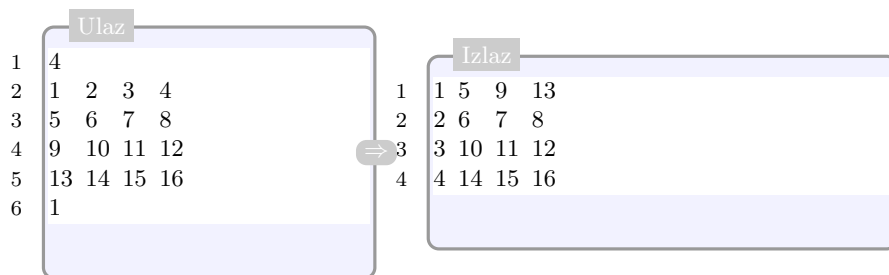
Zbirka zadataka



```

4      int i,j,n,a[30][30],k,pom;
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      scanf ("%d",&k);
10     for ( i = 0; i < n; i++ )
11     {
12         pom = a[i][k-1];
13         a[i][k-1] = a[k-1][i];
14         a[k-1][i] = pom;
15     }
16     for ( i = 0; i < n; i++ )
17     {
18         for ( j = 0; j < n; j++ )
19             printf ("%d\t",a[i][j]);
20         printf ("\n");
21     }
22 }

```

**Zadatak 2.68**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji vrši sortiranje elemenata dela kvadratne matrice  $B_{N \times N}$ , označenog na slici 2.1, u neopadajući redosled. Dimeniju matrice i vrednosti elemenata matrice zadaje korisnik. Prikazati matricu nakon transformacije.

$$B = \begin{bmatrix} 5 & 1 & 7 & 3 \\ 1 & 8 & 2 & 6 \\ 2 & 9 & 4 & 5 \\ 6 & 0 & 1 & 10 \end{bmatrix} \rightarrow B = \begin{bmatrix} 1 & 2 & 3 & 5 \\ 1 & 8 & 6 & 6 \\ 2 & 7 & 4 & 5 \\ 6 & 0 & 1 & 10 \end{bmatrix}$$

Slika 2.1

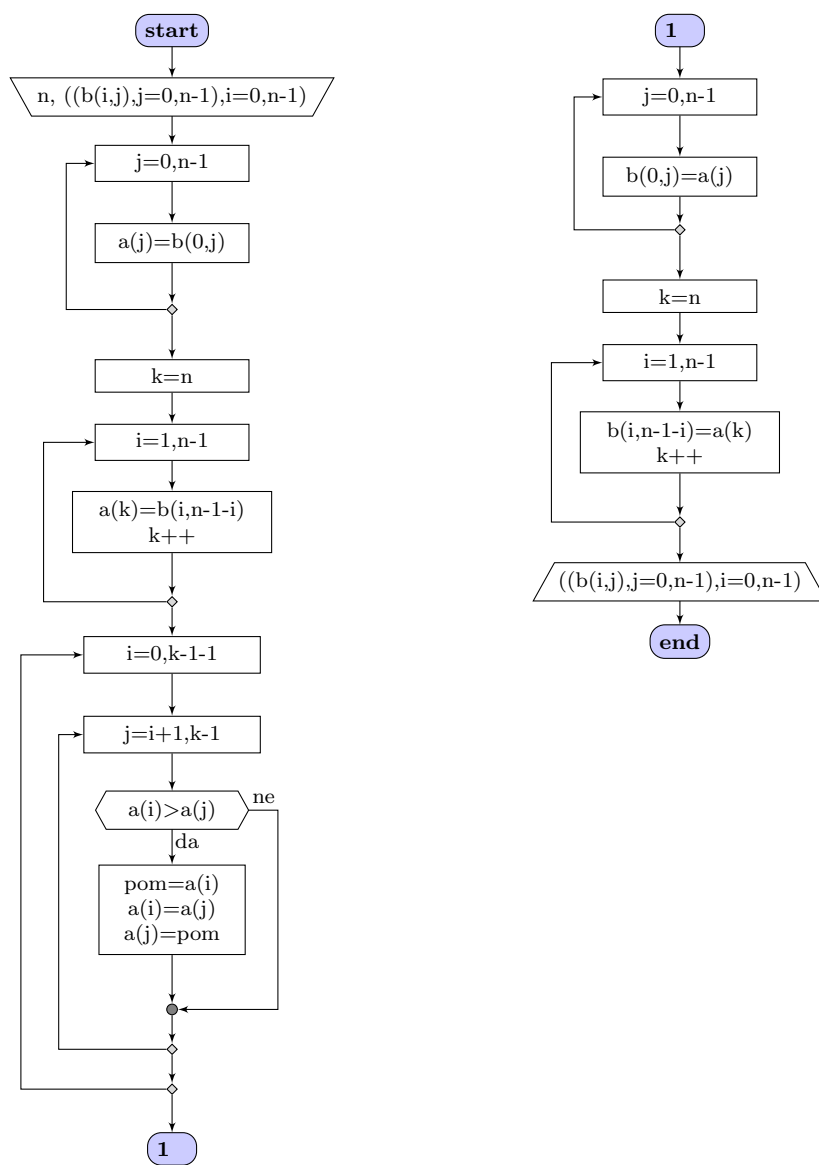
**Rešenje**

```

1  #include "stdio.h"
2  int main()
3  {
4      int i,j,pom,a[30],b[15][15],n,k;
5      scanf ("%d", &n);
6      for (i = 0; i<n; i++)
7          for (j = 0; j<n; j++)
8              scanf ("%d", &b[i][j]);
9      // prvu vrstu i sporednu dijagonalu u pomocni niz
10     for (j = 0; j<n; j++)
11         a[j]=b[0][j];
12     k=n;
13     for (i = 1; i<n; i++)
14     {
15         a[k]=b[i][n-1-i];
16         k++;
17     }
18     // sortiranje pomocnog niza
19     for (i = 0; i<k-1; i++)
20         for (j = i+1; j<k; j++)

```

Zbirka zadataka



```

21         if (a[i]>a[j])
22         {
23             pom=a[i];
24             a[i]=a[j];
25             a[j]=pom;
26         }
27         //pomocni niz u prvu vrstu i sporednu dijagonalu

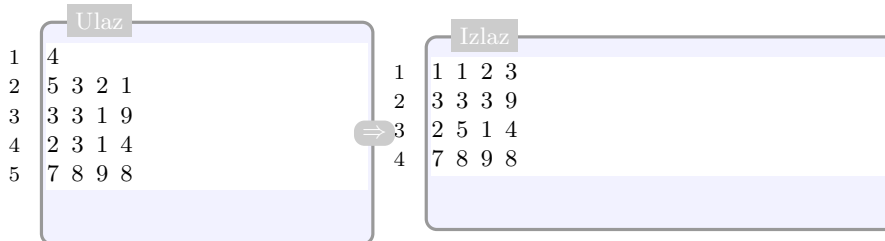
```

Zbirka zadataka

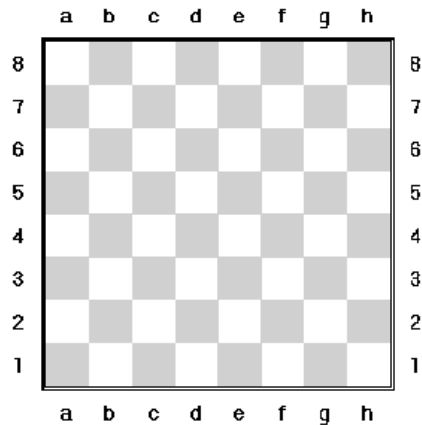
```

28     for (j = 0; j<n; j++)
29         b[0][j]=a[j];
30     k=n;
31     for (i = 1; i<n; i++)
32     {
33         b[i][n-1-i]=a[k];
34         k++;
35     }
36     for (i = 0; i<n; i++)
37     {
38         for (j = 0; j<n; j++)
39             printf ("%d", b[i][j]);
40         printf ("\n");
41     }
42 }

```

**Zadatak 2.69**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji elemente matrice dimenzija 8x8 popunjava brojevima 0 i 1, tako da matrica predstavlja šahovsku tablu (slika 2.2). Vrednost 0 odgovara belom polju, vrednost 1 odgovara crnom polju. Smatrati da element matrice (1,1) odgovara polju a8, element (1,2) polju b8, itd. Korisnik unosi pozicije bele i crne kraljice unoseći indekse polja (npr. ako je bela kraljica na d2, unosi se 4 i 7). Nakon formiranja matrice, u matricu upisati pozicije kraljica tako što se za belu kraljicu upisuje 6 a za crnu kraljicu 9 u odgovarajući element. Prikazati matricu nakon upisivanja i prikazati poruku o tome da li se kraljice napadaju ili ne.



Slika 2.2

**Rešenje**

Matricu je moguće popuniti pomoću dve strategije:

1. Polja su naizmenično bela pa crna, ako se obilazi po vrstama. Pri prelasku s jedne vrste na početak sledeće, polje zadržava boju.
2. Polja čiji je zbir indeksa paran su bela, polja čiji je zbir indeksa neparan su crna.

Kraljice se napadaju ukoliko je ispunjen neki od sledećih uslova:

- a) ako se nalaze u istoj vrsti ili u istoj koloni (prvi ili drugi indeks im je isti),
- b) nalaze se na istom dijagonalnom pravcu: apsolutna vrednost razlike prvih indeksa je jednaka apsolutnoj vrednosti razlike drugih indeksa.

**varijanta 1:**

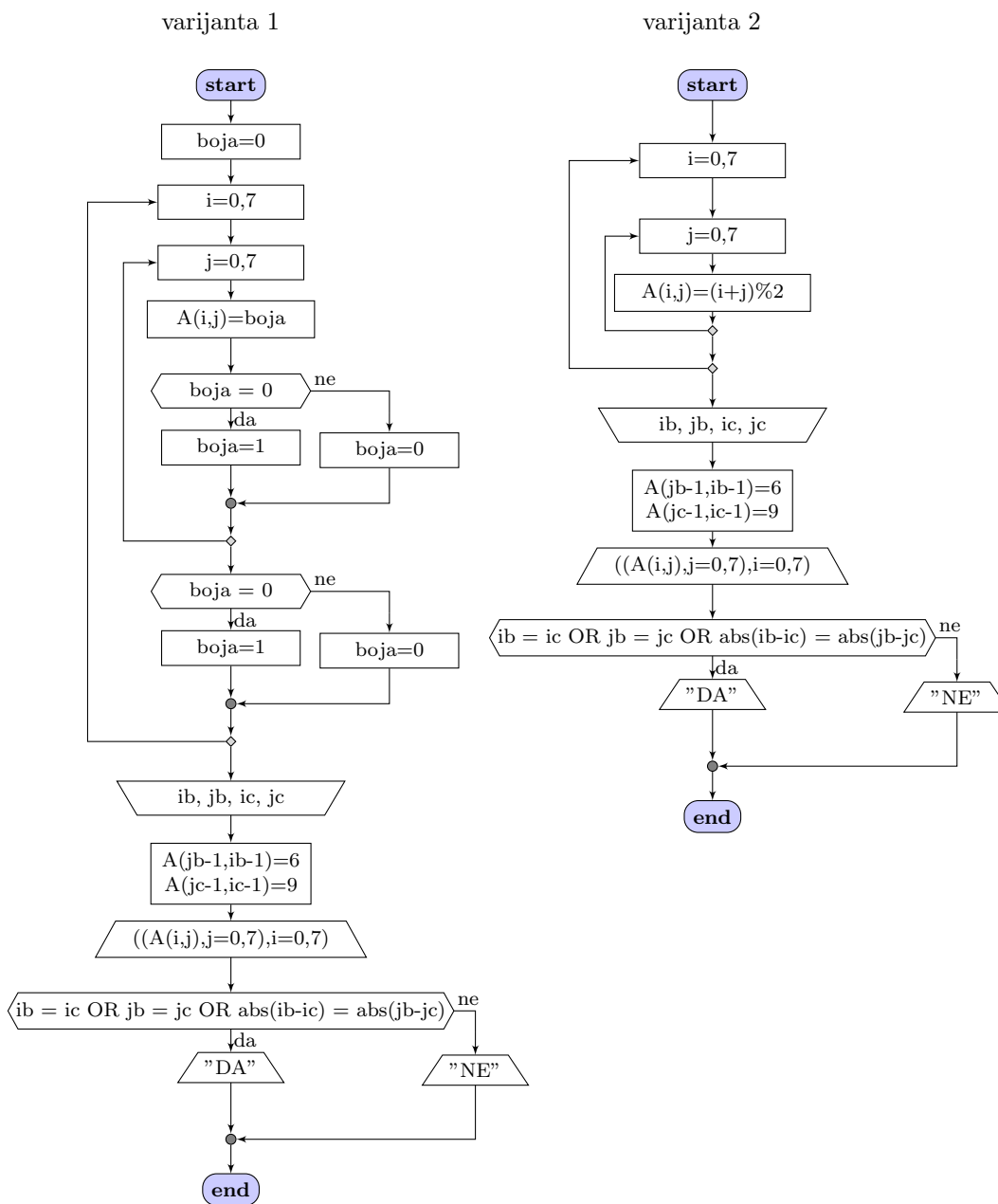
```

1  #include <stdio.h>
2  #include <math.h>
3  void main()
4  {
5      char A[8][8];
6      int i, j, ib, ic, jb, jc;
7      int boja=0;
8      for (i=0;i<8;i++)
9      {
10         for (j=0;j<8;j++)
11         {
12             A[i][j]=boja;
13             if (boja==0)
14                 boja=1;
15             else
16                 boja=0;

```

*Zbirka zadataka*





```

17     }
18     if (boja==0)
19         boja=1;
20     else
21         boja=0;

```

Zbirka zadataka

```

22     }
23     scanf("%d%d%d", &ib, &jb, &ic, &jc);
24     A[jb-1][ib-1]=6;
25     A[jc-1][ic-1]=9;
26     for (i=0; i<8; i++)
27     {
28         for (j=0; j<8; j++)
29             printf("%d ", A[i][j]);
30         printf("\n");
31     }
32     if (ib==ic || jb==jc || abs(ib-ic)==abs(jb-jc))
33         printf("Kraljice se napadaju.\n");
34     else
35         printf("Kraljice se NE napadaju.\n");
36 }

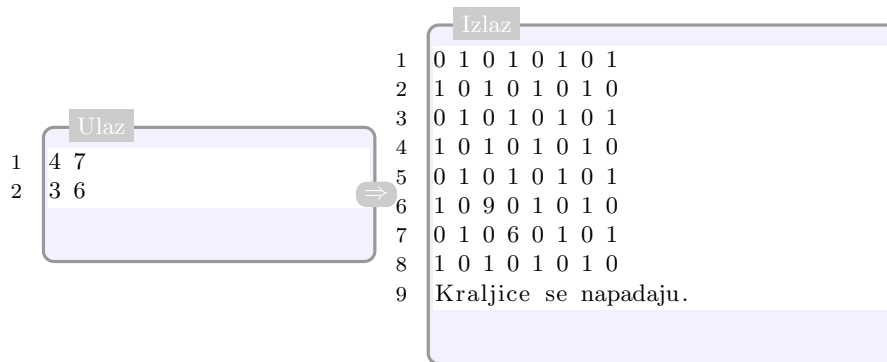
```

#### varijanta 2:

```

1  #include <stdio.h>
2  #include <math.h>
3  void main()
4  {
5      char A[8][8];
6      int i, j, ib, ic, jb, jc;
7      for (i=0; i<8; i++)
8          for (j=0; j<8; j++)
9              A[i][j]=(i+j)%2;
10     scanf("%d%d%d", &ib, &jb, &ic, &jc);
11     A[jb-1][ib-1]=6;
12     A[jc-1][ic-1]=9;
13     for (i=0; i<8; i++)
14     {
15         for (j=0; j<8; j++)
16             printf("%d ", A[i][j]);
17         printf("\n");
18     }
19     if (ib==ic || jb==jc || abs(ib-ic)==abs(jb-jc))
20         printf("Kraljice se napadaju.\n");
21     else
22         printf("Kraljice se NE napadaju.\n");
23 }

```

**Zadatak 2.70**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji generiše kvadratnu matricu  $A$  reda  $N$  oblika

$$A = \begin{cases} [1], & N = 1 \\ \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}, & N = 2 \\ \begin{bmatrix} 0 & 0 & 1 \\ 0 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, & N = 3 \\ \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 3 \\ 0 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 \end{bmatrix}, & N = 4 \end{cases}.$$

Red matrice  $N$  zadaje korisnik. Prikazati generisanu matricu.

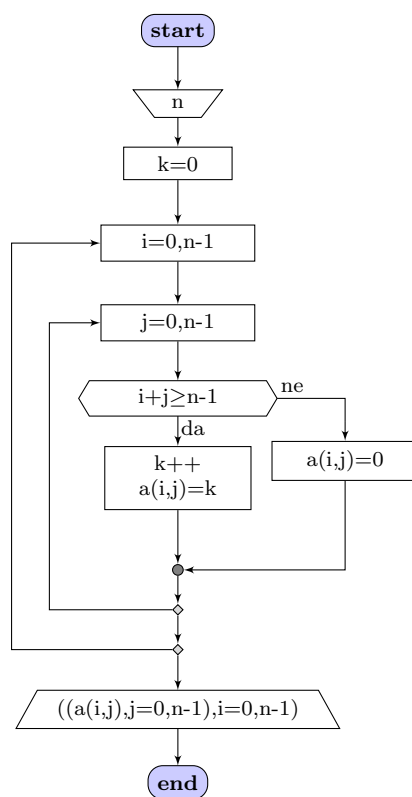
**Rešenje**

```

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

```

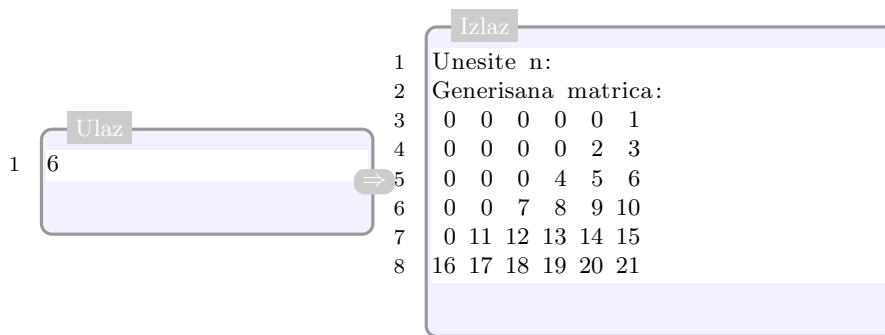
Zbirka zadataka



```

16         else
17             a[i][j] = 0;
18         }
19     printf("Generisana matrica:\n");
20     for(i=0; i<n; i++)
21     {
22         for(j=0; j<n; j++)
23             printf("%2d ", a[i][j]);
24         printf("\n");
25     }
26 }

```

**Zadatak 2.71**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program za množenje matrica

$$C_{N \times M} = A_{N \times K} \cdot B_{K \times M}.$$

Dimenzije matrica  $N$ ,  $M$  i  $K$ , i vrednosti elemenata matrica  $A$  i  $B$  zadaje korisnik.

Napomena: element rezultujuće matrice  $c_{i,j}$  definisan je kao skalarni proizvod  $i$ -te vrste matrice  $A$  i  $j$ -te kolone matrice  $B$ , t.j.

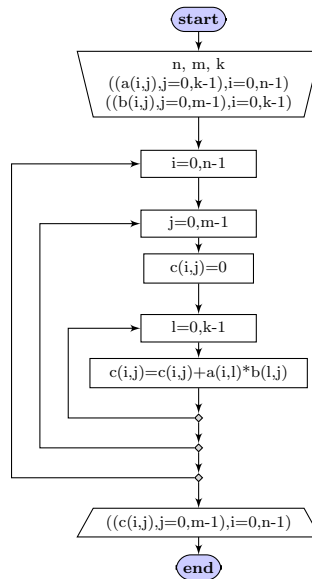
$$c_{i,j} = \sum_{l=0}^{K-1} a_{i,l} \cdot b_{l,j}.$$

**Rešenje**

```

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

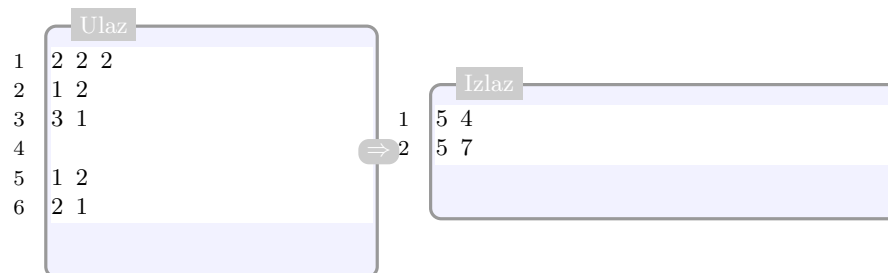
Zbirka zadataka



```

21         for ( l = 0; l < k; l++ )
22             c[i][j] = c[i][j] + a[i][l] * b[l][j];
23         }
24     }
25     for ( i = 0; i < n; i++ )
26     {
27         for ( j = 0; j < m; j++ )
28             printf ("%d\\n", c[i][j]);
29         printf ("\\n");
30     }
31 }

```

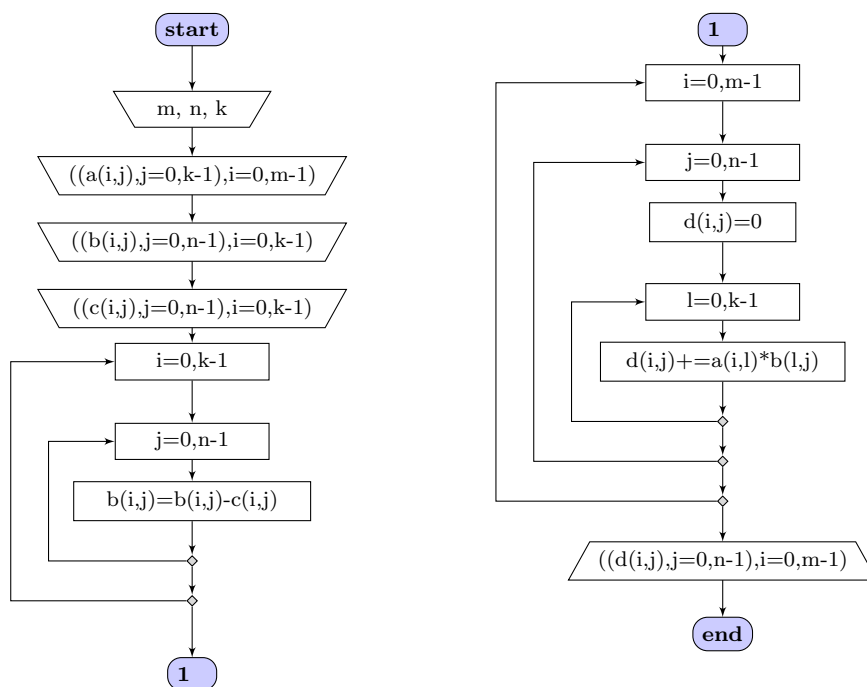


**Zadatak 2.72**

Nacrtati strukturni dijagram toka algoritma i napisati strukturni program kojim se određuje i prikazuje matrica

$$D = A \cdot (B - C),$$

gde je matrica  $A$  dimenzija  $m \times k$ , a matrice  $B$  i  $C$  dimenzija  $k \times n$ . Predvideti unos vrednosti dimenzija  $m$ ,  $n$  i  $k$ , kao i vrednosti elemenata matrica  $A$ ,  $B$  i  $C$ .

**Rešenje**

```

1  #include <stdio.h>
2  void main()
3  {
4      int i,j,l,n,m,k,a[20][20],b[20][20],c[20][20],d[20][20];
5      printf("Unesite m,n,i,k dimenzije matrica:\n");
6      scanf("%d%d%d",&m,&n,&k);
7      printf("Unesite matricu a:\n");
8      for(i=0;i<m;i++)
9          for(j=0;j<k;j++)
10             scanf("%d",&a[i][j]);
11     printf("Unesite matricu b:\n");
12     for(i=0;i<k;i++)
13         for(j=0;j<n;j++)

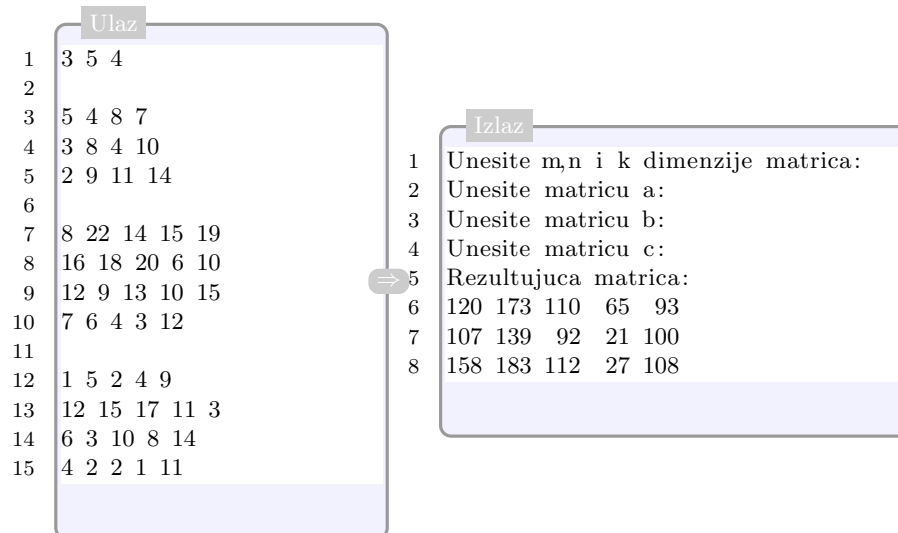
```

Zbirka zadataka

```

14         scanf("%d", &b[i][j]);
15     printf("Unesite matricu c:\n");
16     for(i=0;i<k;i++)
17         for(j=0;j<n;j++)
18             scanf("%d", &c[i][j]);
19     for(i=0;i<k;i++)
20         for(j=0;j<n;j++)
21             b[i][j] = b[i][j] - c[i][j];
22     for(i=0;i<n;i++)
23         for(j=0;j<n;j++)
24         {
25             d[i][j] = 0;
26             for(l=0;l<k;l++)
27                 d[i][j] += a[i][l]*b[l][j];
28         }
29     printf("Rezultujuca matrica:\n");
30     for(i=0;i<n;i++)
31     {
32         for(j=0;j<n;j++)
33             printf("%3d", d[i][j]);
34         printf("\n");
35     }
36 }

```





## 3

# Funkcije i rad sa fajlovima

## 3.1 Funkcije sa skalarnim parametrima

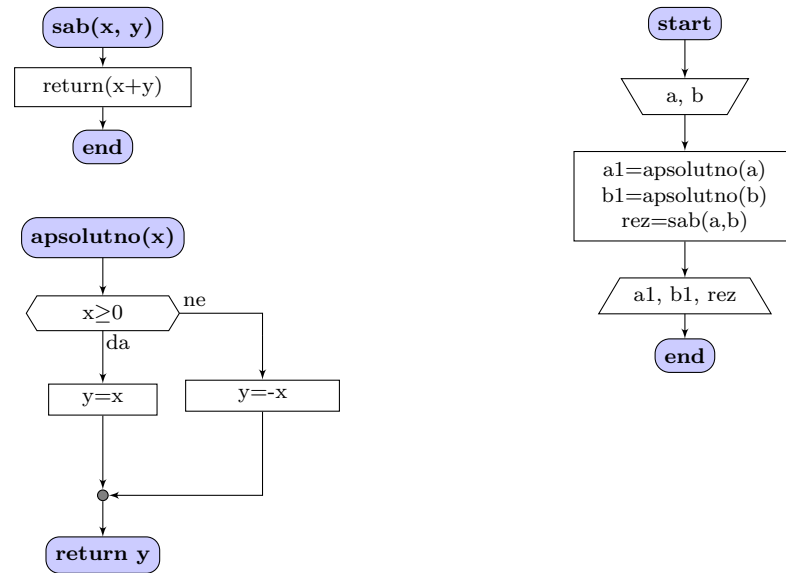
### Zadatak 3.73

---

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturnu funkciju za računanje zbira dva cela broja preneti preko parametara i funkciju za određivanje apsolutne vrednosti celog broja prenetog preko parametara. U glavnom programu korišćenjem formiranih funkcija odrediti i prikazati apsolutne vrednosti i zbir brojeva  $a$  i  $b$ , čije vrednosti zadaje korisnik.

### Rešenje

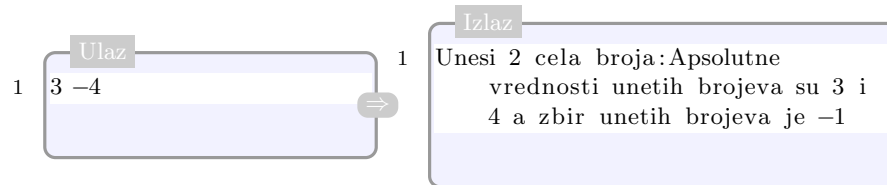
```
1  #include "stdio.h"
2  int sab(int x, int y)
3  {
4      return (x+y);
5  }
6  int apsolutno(int x)
7  {
8      int y;
9      if (x>=0)
10         y=x;
11     else
12         y=-x;
13     return y;
14 }
15 main()
16 {
17     int a,b,a1,b1,rez;
18     printf("Unesi 2 cela broja:");
19     scanf("%d%d",&a,&b);
```



```

20     a1=apsolutno(a);
21     b1=apsolutno(b);
22     rez=sab(a,b);
23
24     printf("Apsolutne_vrednosti_unetih_brojeva_su_%d_i_%d_a_zbir_
unetih_brojeva_je_%d\n", a1, b1, rez);
25     /* ili , bez promenljivih, odmah nakon unosa brojeva:
26     printf("Apsolutne vrednosti unetih brojeva su %d i %d a zbir
unetih brojeva je %d\n", apsolutno(a), apsolutno(b), sab(a,b));
27     */
28 }

```



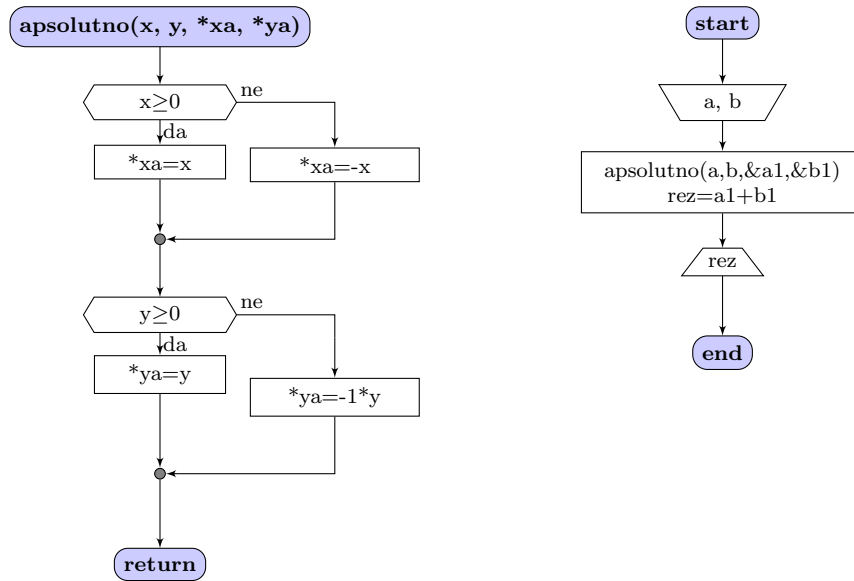
### Zadatak 3.74

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturnu funkciju koja određuje i vraća apsolutne vrednosti dva cela broja preneti preko parametara. U glavnom programu korišćenjem formirane funkcije odrediti apsolutne

Zbirka zadataka

vrednosti brojeva  $a$  i  $b$ , čije vrednosti zadaje korisnik, i prikazati zbir dobijenih apsolutnih vrednosti.

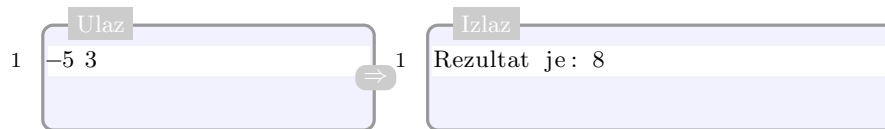
### Rešenje



```

1  #include "stdio.h"
2  void apsolutno(int x, int y, int* xa, int* ya)
3  {
4
5      if (x>=0)
6          *xa=x;
7      else
8          *xa=-x;
9      if (y>=0)
10         *ya=y;
11     else
12         *ya=-y;
13     return;
14 }
15 void main()
16 {
17     int a,b,a1,b1,rez;
18     scanf("%d%d",&a,&b);
19     apsolutno(a,b,&a1,&b1);
20     rez=a1+b1;
21     printf("Rezultat je: %d\n",rez);
22 }
  
```

Zbirka zadataka

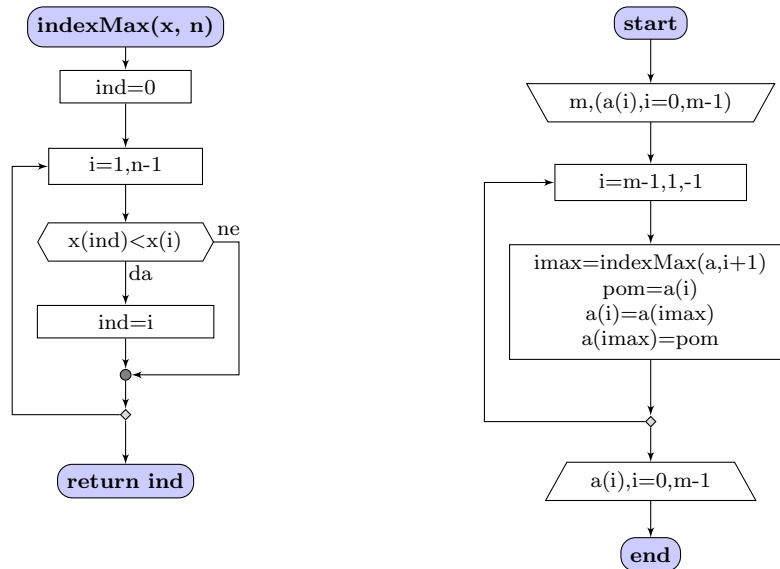


## 3.2 Prenos nizova preko parametara funkcije

### Zadatak 3.75

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati funkciju koja određuje indeks maksimalnog elementa u nizu  $X$  sa  $N$  elemenata. U glavnom programu učitati niz  $A$  sa  $M$  elemenata i korišćenjem formirane funkcije urediti elemente niza u rastući redosled. Prikazati niz nakon uređenja.

### Rešenje



```

1 #include <stdio.h>
2
3 int indexMax(int* x, int n)
4 {
5     int i, ind = 0;
6     for(i=1; i<n; i++)
7         if(x[ind] < x[i])
8             ind = i;

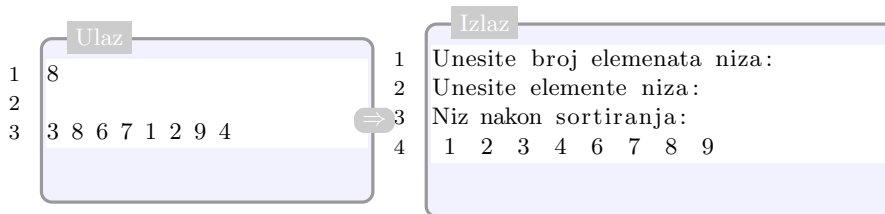
```

Zbirka zadataka

```

9         return ind;
10    }
11
12    void main()
13    {
14        int m, i, a[20], imax, pom;
15        printf("Unesite broj elemenata niza:\n");
16        scanf("%d", &m);
17        printf("Unesite elemente niza:\n");
18        for(i=0; i<m; i++)
19            scanf("%d", &a[i]);
20        for(i=m-1; i>0; i--)
21        {
22            imax = indexMax(a, i+1);
23            pom = a[i];
24            a[i] = a[imax];
25            a[imax] = pom;
26        }
27        printf("Niz nakon sortiranja:\n");
28        for(i=0; i<m; i++)
29            printf("%2d", a[i]);
30        printf("\n");
31    }

```



### Zadatak 3.76

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati funkciju koja rotira elemente niza  $X$  sa  $N$  elemenata za  $k$  mesta ulevo. U glavnom programu učitati matricu  $A$  dimenzija  $M \times M$  i korišćenjem formirane funkcije rotirati elemente prve vrste za jedno mesto ulevo, druge za dva, itd. Elemente poslednje vrste matrice rotirati za  $M$  mesta ulevo. Prikazati matricu nakon transformacije.

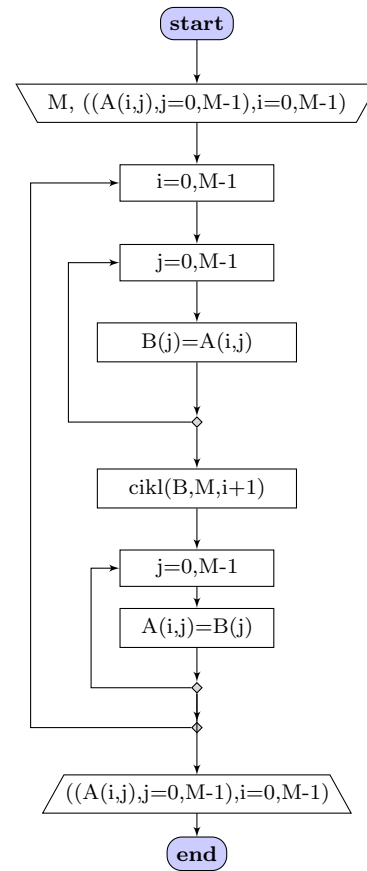
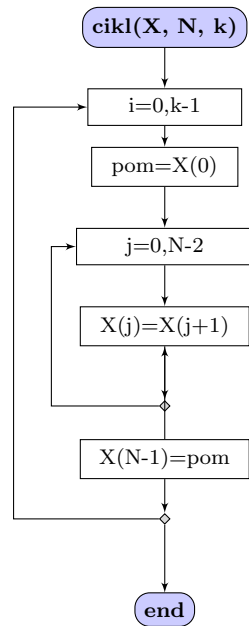
### Rešenje

```

1  #include <stdio.h>
2  void cikl(int X[], int N, int k)
3  {
4      int i, j, pom;
5      for ( i = 0; i < k; i++ )

```

Zbirka zadataka



```

6      {
7          pom = X[0];
8          for ( j = 0; j < N-1; j++ )
9              X[j] = X[j+1];
10         X[N-1] = pom;
11     }
12 }
13 void main()
14 {
15     int A[30][30], MB[30], i, j;
16     scanf ("%d", &M);
17     for ( i = 0; i < M; i++ )
18         for ( j = 0; j < M; j++ )
19             scanf ("%d", &A[i][j]);
20     for ( i = 0; i < M; i++ )
21     {
22         for ( j = 0; j < M; j++ )

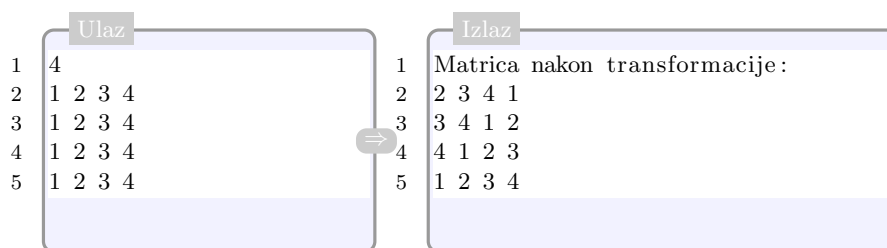
```

Zbirka zadataka

```

23         B[j] = A[i][j];
24     cikl ( B,M,i+1);
25     for ( j = 0; j < M; j++ )
26         A[i][j] = B[j];
27 }
28 printf ("Matrica nakon transformacije:\n");
29 for ( i = 0; i < M; i++ )
30 {
31     for ( j = 0; j < M; j++ )
32         printf ("%d ", A[i][j]);
33     printf ("\n");
34 }
35 }

```

**Zadatak 3.77**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturnu funkciju koja u celobrojnom nizu  $A$  dužine  $N$  pronalazi prvi par indeksa  $(p, k)$  koji zadovoljavaju uslov da je zbir članova niza sa indeksima od  $p$  do  $k$  jednak zadatom broju  $M$ . Ukoliko takav par indeksa ne postoji funkcija vraća  $(-1, -1)$ . U glavnom programu učitati niz i parametar  $M$ , izvršiti funkciju nad unetim nizom i prikazati vrednosti indeksa  $(p, k)$ .

Primer:  $A = \{1, 2, 8, 3, 4\}$ ,  $M = 13 \Rightarrow (p, k) = (1, 3)$ , jer je  $2+8+3=13$ .

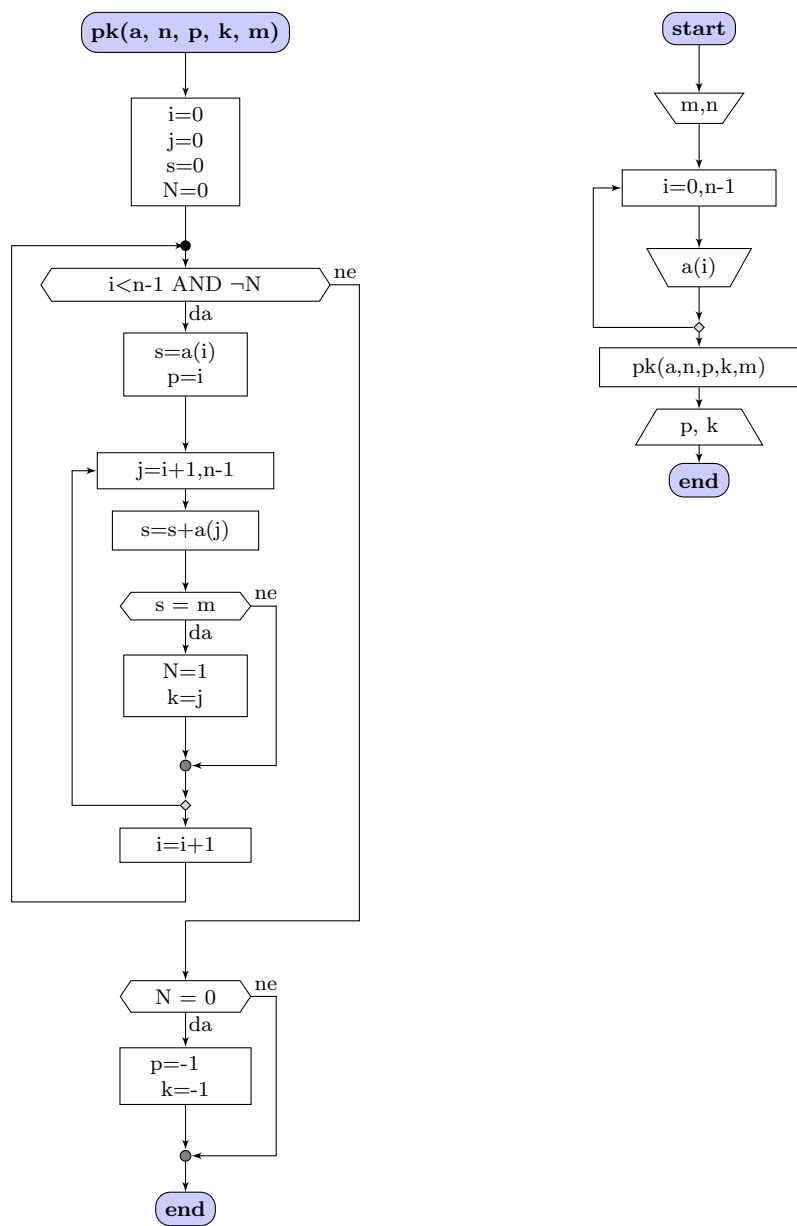
**Rešenje**

```

1  #include <stdio.h>
2  void pk (int a[], int n, int *p, int *k, int m)
3  {
4      int i=0, j=0, s=0, N=n;
5      while (i < n-1 && !N)
6      {
7          s=a[i]; *p=i;
8          for (j=i+1; j < n; j++)
9              {
10                 s=s+a[j];
11                 if (s==m)
12                     {

```

Zbirka zadataka



```

13                                     N=1;
14                                     *k=j ;
15                                     }
16                                     }
17                                     i=i+1;
  
```

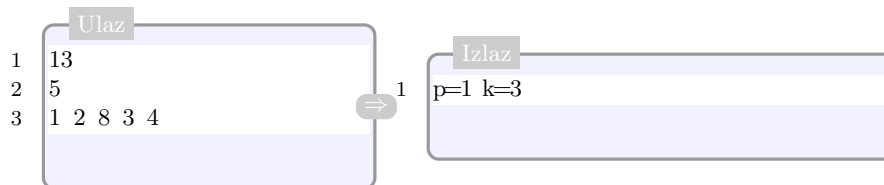
Zbirka zadataka



```

18     }
19     if (N==0)
20     {
21         *p=-1;
22         *k=-1;
23     }
24 }
25 void main ()
26 {
27     int a[50],n,i,m,p,k;
28     scanf("%d",&m);
29     scanf("%d",&n);
30     for (i=0; i<n; i++)
31         scanf("%d",&a[i]);
32     pk (a,n,&p,&k,m);
33     printf ("p=%d k=%d",p,k);
34 }

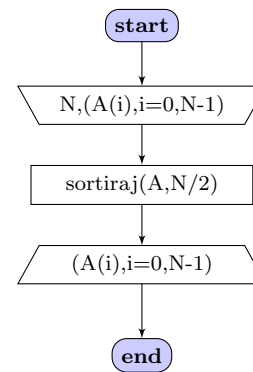
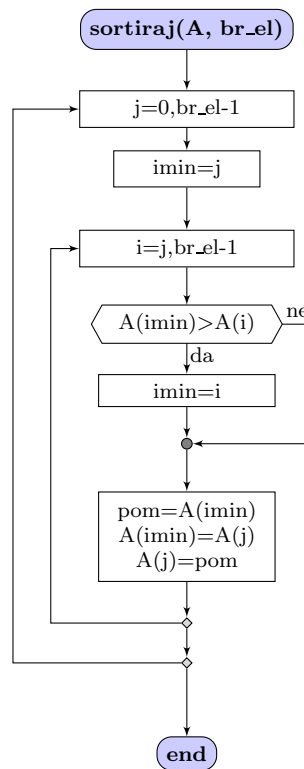
```

**Zadatak 3.78**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati funkciju koja niz prenet preko parametara funkcije uređuje u rastući redosled. U glavnom programu sa tastature uneti niz  $A$  sa  $N$  elemenata i korišćenjem formirane funkcije urediti prvu polovinu niza u rastući redosled. Drugu polovinu ostaviti nepromenjenju. Prikazati niz nakon transformacije.

**Rešenje**

*Zbirka zadataka*



```

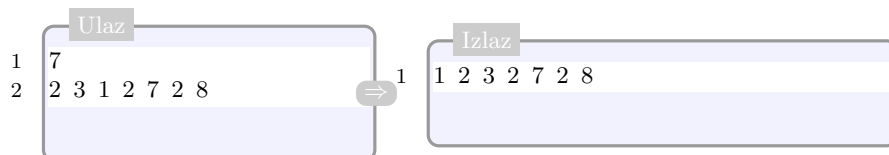
1  #include "stdio.h"
2  void sortiraj(int *A, int br_el)
3  {
4      int i, j, imin, pom;
5      for (j=0; j<br_el; j++)
6      {
7          imin = j;
8          for (i=j; i<br_el; i++)
9              if (A[imin] > A[i])
10                 imin = i;
11          pom = A[imin];
12          A[imin] = A[j];
13          A[j] = pom;
14      }
15  }
16  void main()
17  {
18      int A[100], N, i;
19      scanf("%d", &N);
20      for(i = 0; i < N; i++)
21          scanf("%d", &A[i]);
  
```

Zbirka zadataka

```

22
23     sortiraj(A,N/2);
24
25     for(i=0; i<N; i++)
26         printf("%d ", A[i]);
27 }

```

**Zadatak 3.79**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati funkciju koja u zavisnosti od prenetog parametra funkcije izračunava presek ili uniju dva skupa. U glavnom programu korišćenjem formirane funkcije odrediti elemente skupa

$$\mathbb{E} = (\mathbb{A} \cap \mathbb{B}) \cup \mathbb{C}$$

i prikazati ih na ekranu. Skup  $\mathbb{A}$  predstavljen je nizom sa  $m$  elemenata, skup  $\mathbb{B}$  nizom sa  $n$  elemenata, a skup  $\mathbb{C}$  nizom sa  $p$  elemenata. Elementi skupova  $\mathbb{A}$ ,  $\mathbb{B}$  i  $\mathbb{C}$  su celi brojevi čije vrednosti zadaje korisnik.

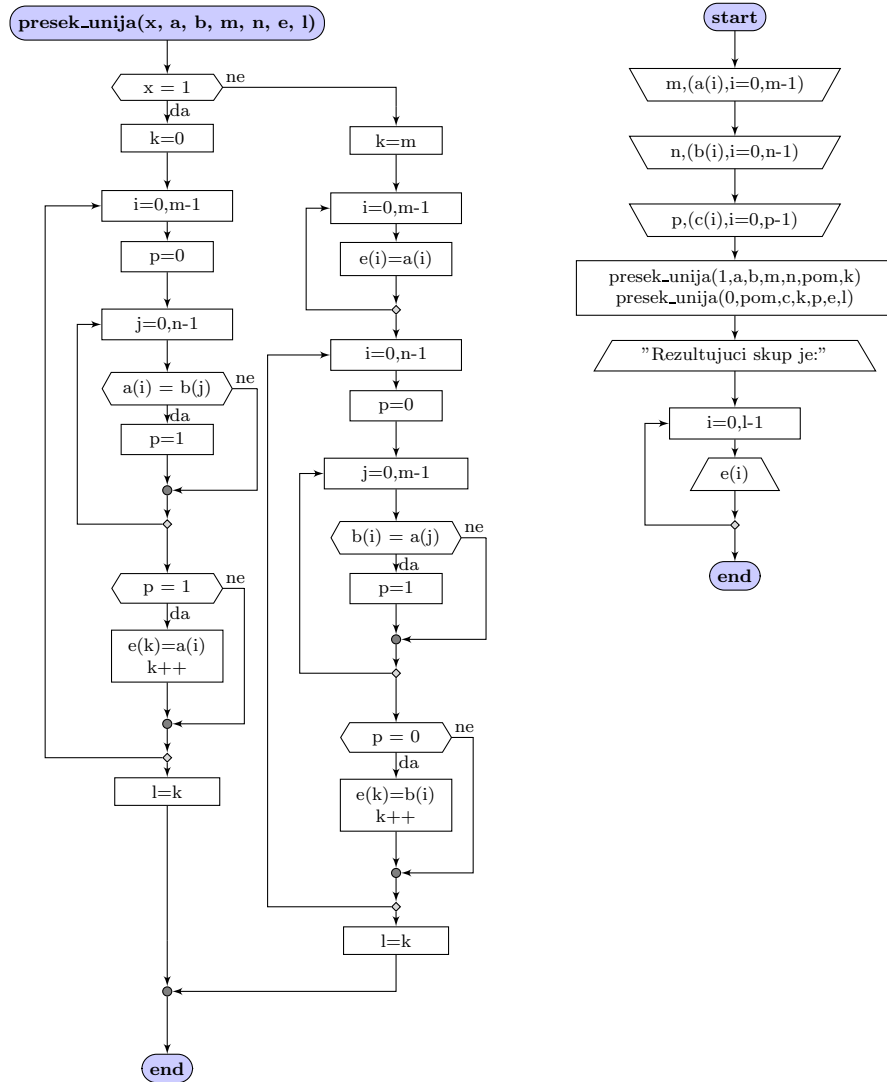
**Rešenje**

```

1  #include <stdio.h>
2  void presek_unija(int x, int a[], int b[], int m, int n, int e[], int *l)
3  {
4      int i,j,p,k;
5      if ( x == 1 ) //presek
6      {
7          k = 0;
8          for ( i = 0; i < m; i++ )
9          {
10             p=0;
11             for ( j = 0; j < n; j++ )
12                 if ( a[i] == b[j] )
13                     p = 1;
14             if( p == 1 )
15                 {
16                     e[k] = a[i];
17                     k++;
18                 }
19             }
20             *l = k;
21     }

```

Zbirka zadataka



```

22     else // unija
23     {
24         k = m;
25         for ( i = 0; i < m; i++ )
26             e[i] = a[i];
27         for ( i = 0; i < n; i++ )
28         {
29             p = 0;
30             for ( j = 0; j < m; j++ )
31                 if ( b[i] == a[j] )

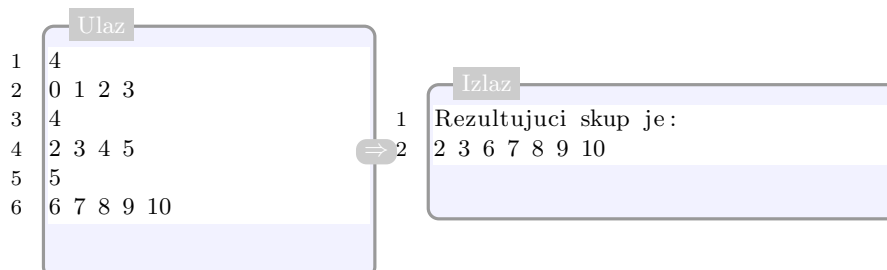
```

Zbirka zadataka

```

32                                     p = 1;
33             if ( p == 0)
34             {
35                 e[k] = b[i];
36                 k++;
37             }
38         }
39         *l = k;
40     }
41 }
42 void main()
43 {
44     int i, j, n, m, p, l, k, a[20], b[20], c[20], e[40], pom[40];
45     scanf ("%d", &m);
46     for ( i = 0; i < m; i++ )
47         scanf ("%d", &a[i]);
48     scanf ("%d", &n);
49     for ( i = 0; i < n; i++ )
50         scanf ("%d", &b[i]);
51     scanf ("%d", &p);
52     for ( i = 0; i < p; i++ )
53         scanf ("%d", &c[i]);
54     presek_unija(1, a, b, m, n, pom, &k);
55     presek_unija(0, pom, c, k, p, e, &l);
56     printf ("Rezultujuci skup je:\n");
57     for ( i = 0; i < l; i++ )
58         printf ("%d ", e[i]);
59 }

```



### 3.3 Matrice u funkciji

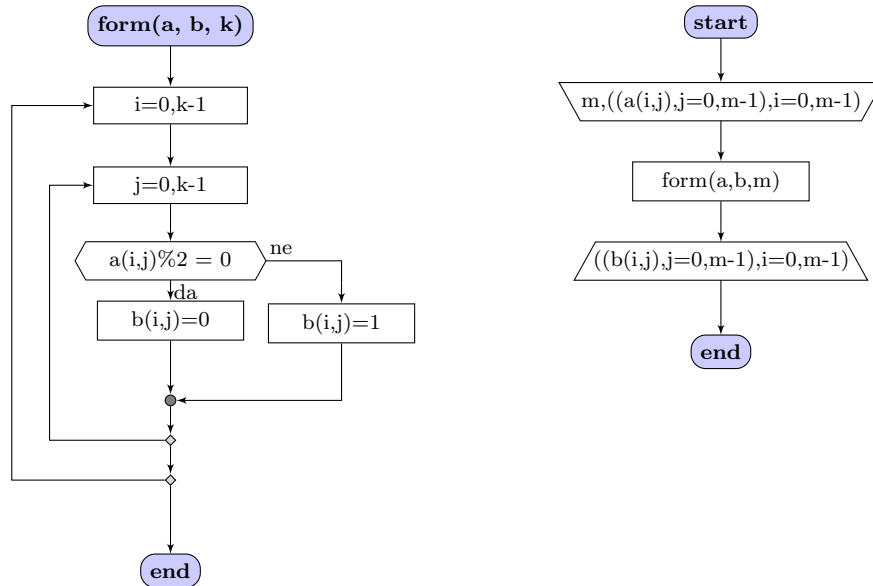
#### Zadatak 3.80

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati funkciju kojom se od matrice  $A_{M \times M}$  formira matrica  $B_{M \times M}$ . Elementi matrice  $B$  dobijaju se tako što se elementi sa parnim vrednostima matrice  $A$  zamenjuju nulom,

*Zbirka zadataka*

a elementi sa neparnim vrednostima jedinicom. U glavnom programu uneti matricu A i korišćenjem funkcije formirati matricu B. Prikazati matricu B.

### Rešenje



```

1  #include <stdio.h>
2  #define dim 30
3  void form(int a[dim][dim], int b[dim][dim], int k)
4  {
5      int i, j;
6      for (i = 0; i < k; i++)
7          for (j = 0; j < k; j++)
8              if (a[i][j] % 2 == 0)
9                  b[i][j] = 0;
10             else
11                 b[i][j] = 1;
12  }
13
14 void main()
15 {
16     int a[dim][dim], b[dim][dim], i, j, m;
17     scanf ("%d", &m);
18     for (i = 0; i < m; i++)
19         for (j = 0; j < m; j++)
20             scanf ("%d", &a[i][j]);
21

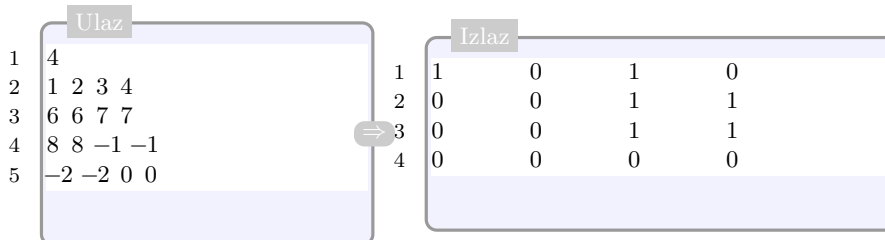
```

Zbirka zadataka

```

22     form(a,b,m);
23
24     for (i = 0; i<m; i++)
25     {
26         for (j=0; j<n; j++)
27             printf ("%4d", b[i][j]);
28         printf("\n");
29     }
30 }

```

**Zadatak 3.81**

Nacrtati stuktorni dijagram toka algoritma i napisati strukturnu funkciju na programskom jeziku C kojom se kvadratna matrica  $A_{N \times N}$  modifikuje tako da elementi ispod sporedne dijagonale u svakoj koloni budu ciklično pomereni za  $k$  mesta nadole ( $k$  je parametar funkcije). U glavnom programu uneti dimenziju  $M$  i elemente matrice  $B_{M \times M}$ , i korišćenjem formirane funkcije izvršiti transformaciju matrice. Prikazati matricu  $B$  pre i posle transformacije. Napomena: nije dozvoljeno korišćenje pomoćne matrice ili niza.

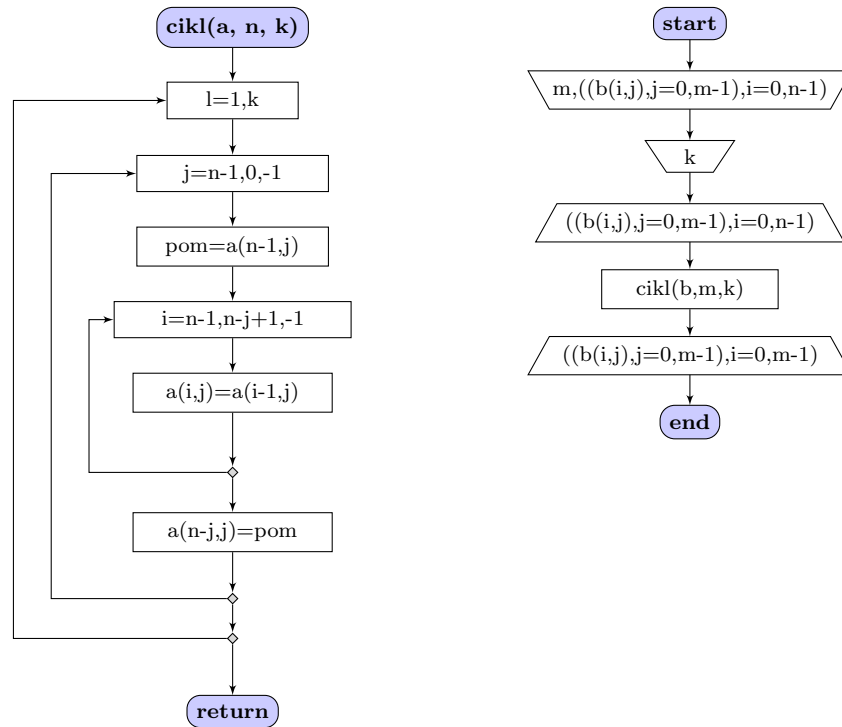
**Rešenje**

```

1  #include <stdio.h>
2  void cikl(int a[][20],int n,int k)
3  {
4      int pom,l,i,j;
5      for (l = 1; l<=k; l++)
6          for (j = n-1; j>0; j--)
7              {
8                  pom=a[n-1][j];
9                  for (i=n-1; i>=n-j+1;i--)
10                     a[i][j]=a[i-1][j];
11                 a[n-j][j]=pom;
12             }
13 }
14 void main()
15 {
16     int b[20][20],i,j,m,k;

```

*Zbirka zadataka*



```

17     printf("Unesite dimenziju matrice: \n");
18     scanf ("%d", &m);
19     printf("Unesite elemente matrice: \n");
20     for (i = 0; i < m; i++)
21         for (j = 0; j < m; j++)
22             scanf ("%d", &b[i][j]);
23     printf("Unesite k: \n");
24     scanf ("%d", &k);
25     printf("Matrica pre ciklicnog pomeranja: \n");
26     for (i = 0; i < m; i++)
27     {
28         for (j = 0; j < m; j++)
29             printf ("%2d", b[i][j]);
30         printf("\n");
31     }
32     cikl(b,m,k);
33     printf("Matrica nakon ciklicnog pomeranja: \n");
34     for (i = 0; i < m; i++)
35     {
36         for (j = 0; j < m; j++)
37             printf ("%2d", b[i][j]);
38         printf("\n");

```

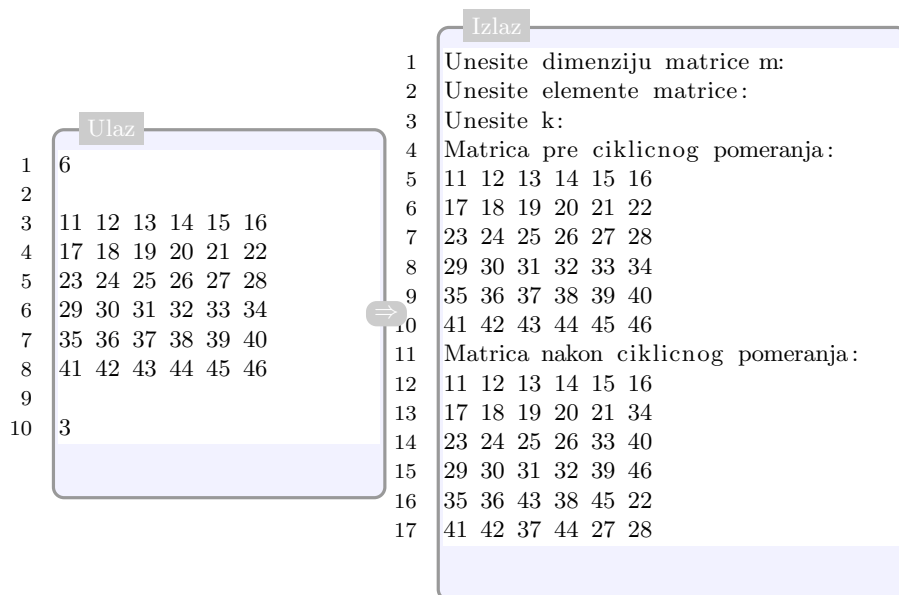
Zbirka zadataka



```

39     }
40 }

```

**Zadatak 3.82**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati funkciju kojim se na osnovu zadate celobrojne kvadratne matrice  $A_{N \times N}$  formiraju matrice  $B_{N \times N}$  i  $C_{N \times N}$  oblika

$$B_{N \times N} = \begin{bmatrix} a_{1,1} & 0 & 0 & 0 & 0 & a_{1,6} \\ a_{2,1} & a_{2,2} & 0 & 0 & a_{2,5} & a_{2,6} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} & a_{3,6} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & a_{4,5} & a_{4,6} \\ a_{5,1} & a_{5,2} & 0 & 0 & a_{5,5} & a_{5,6} \\ a_{6,1} & 0 & 0 & 0 & 0 & a_{6,6} \end{bmatrix}$$

$$C_{N \times N} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} & a_{1,6} \\ 0 & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} & 0 \\ 0 & 0 & a_{3,3} & a_{3,4} & 0 & 0 \\ 0 & 0 & a_{4,3} & a_{4,4} & 0 & 0 \\ 0 & a_{5,2} & a_{5,3} & a_{5,4} & a_{5,5} & 0 \\ a_{6,1} & a_{6,2} & a_{6,3} & a_{6,4} & a_{6,5} & a_{6,6} \end{bmatrix}$$

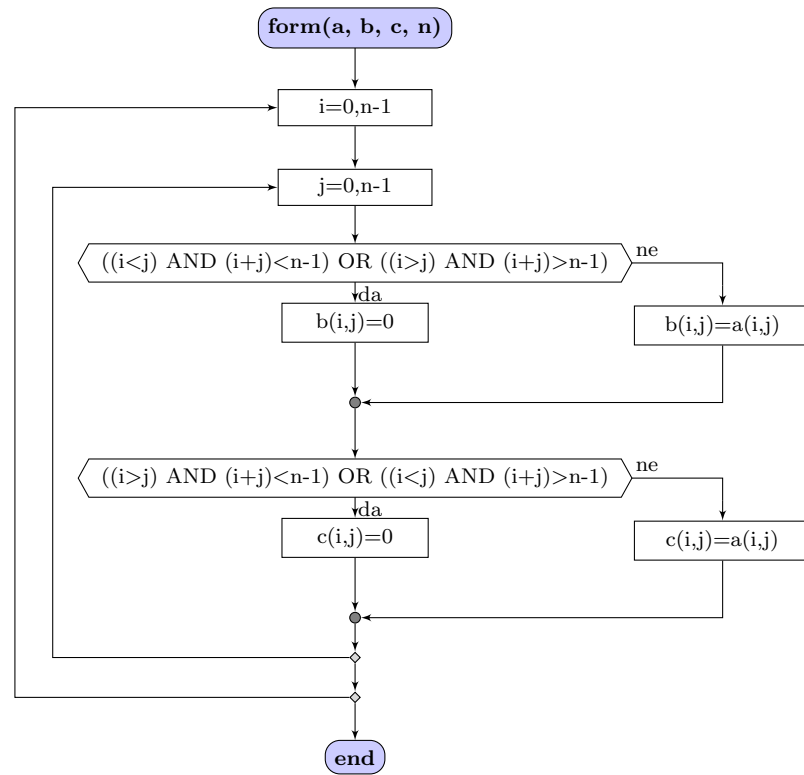
U glavnom programu učitati matricu  $A_{N \times N}$ , pozivom funkcije generisati matrice  $B_{N \times N}$  i  $C_{N \times N}$ , a zatim odrediti matricu

$$D = B - 5 \cdot C.$$

Prikazati na ekranu matrice  $B$ ,  $C$  i  $D$ .

*Zbirka zadataka*

## Rešenje

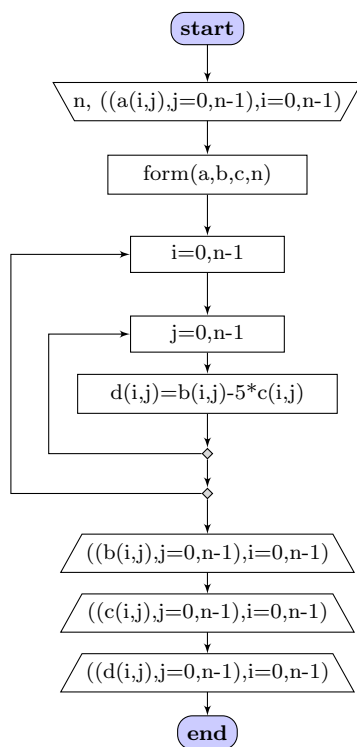


```

1 #include <stdio.h>
2 #define dim 20
3 void form(int a[dim][dim], int b[dim][dim], int c[dim][dim], int n)
4 {
5     int i, j;
6     for(i=0; i<n; i++)
7         for(j=0; j<n; j++)
8             {
9                 if (((i<j) && (i+j)<n-1) || ((i>j) && (i+j)>n-1))
10                    b[i][j]=0;
11                 else
12                    b[i][j]=a[i][j];
13
14                 if (((i>j) && (i+j)<n-1) || ((i<j) && (i+j)>n-1))
15                    c[i][j]=0;
16                 else
17                    c[i][j]=a[i][j];
18             }
19 }

```

Zbirka zadataka



```

20
21 void main()
22 {
23     int a[dim][dim], b[dim][dim], c[dim][dim], d[dim][dim], i, j, n;
24     scanf ("%d", &n);
25     for (i = 0; i < n; i++)
26         for (j = 0; j < n; j++)
27             scanf ("%d", &a[i][j]);
28     form(a,b,c,n);
29     for (i = 0; i < n; i++)
30         for (j = 0; j < n; j++)
31             d[i][j] = b[i][j] - 5*c[i][j];
32     printf("Matrica B:\n");
33     for (i = 0; i < n; i++)
34     {
35         for (j = 0; j < n; j++)
36             printf ("%d\t", b[i][j]);
37         printf("\n");
38     }
39     printf("\nMatrica C:\n");

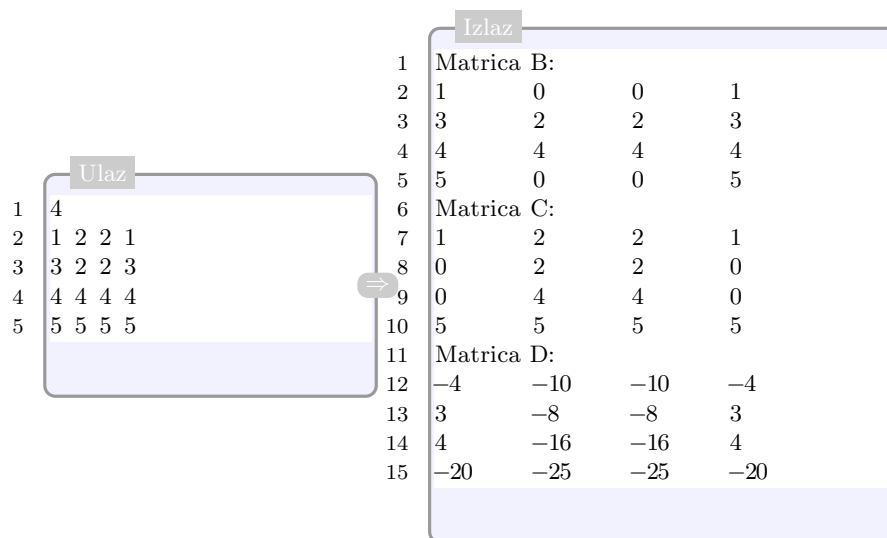
```

Zbirka zadataka

```

40     for (i = 0; i < n; i++)
41     {
42         for (j = 0; j < n; j++)
43             printf ("%d\t", c[i][j]);
44         printf("\n");
45     }
46     printf("\nMatrica D:\n");
47     for (i = 0; i < n; i++)
48     {
49         for (j = 0; j < n; j++)
50             printf ("%d\t", d[i][j]);
51         printf("\n");
52     }
53 }

```



### Zadatak 3.83

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturnu funkciju koja u nizu stringova pronalazi najduži string i vraća ga po referenci pozivajućem programu. U glavnom programu uneti niz stringova i korišćenjem formirane funkcije odrediti i prikazati najduži string.

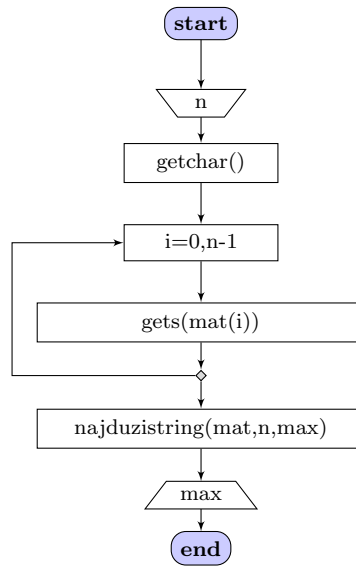
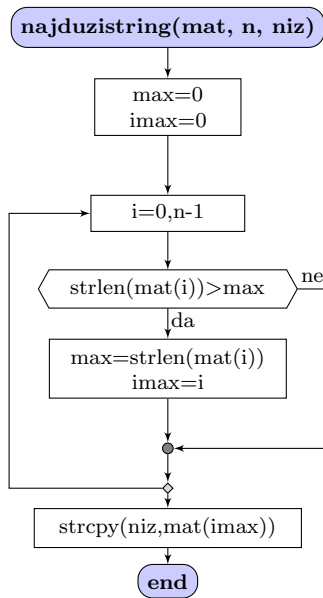
### Rešenje

```

1 #include <stdio.h>
2 #include <string.h>
3 void najduzistring (char mat[][50], int n, char niz[])
4 {

```

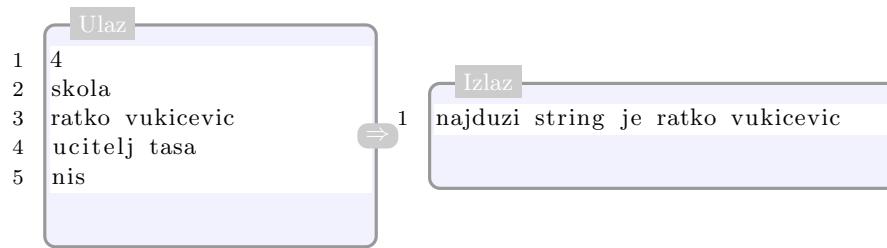
Zbirka zadataka



```

5      int max=0,imax=0,i;
6      for (i=0;i<n;i++)
7          if (strlen(mat[i])>max)
8              {
9                  max=strlen(mat[i]);
10                 imax=i;
11             }
12      strcpy(niz,mat[imax]);
13  }
14
15  void main()
16  {
17      char mat[30][50];
18      char max[50];
19      int n,i;
20      scanf("%d",&n);
21      getchar();
22      for (i=0;i<n;i++)
23          gets(mat[i]);
24      najduzistring(mat, n, max);
25      printf ("najduzi string je %s",max);
26  }
  
```

Zbirka zadataka

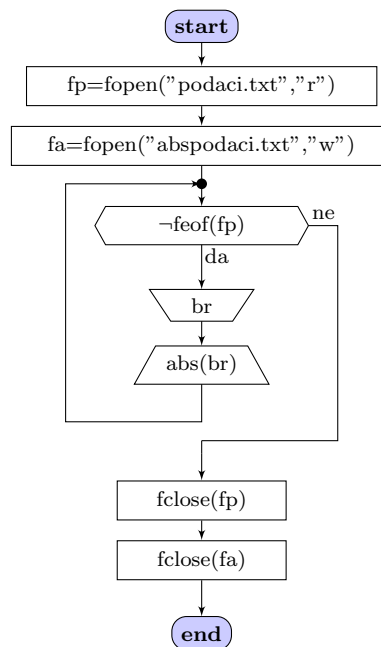


### 3.4 Fajlovi

#### Zadatak 3.84

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji u fajl *"abspodaci.txt"* upisuje apsolutne vrednosti celih brojeva iz fajla *"podaci.txt"*. Prikazati odgovarajuće poruke nakon otvaranja fajlova, pre učitavanja i nakon upisa podataka fajl.

#### Rešenje



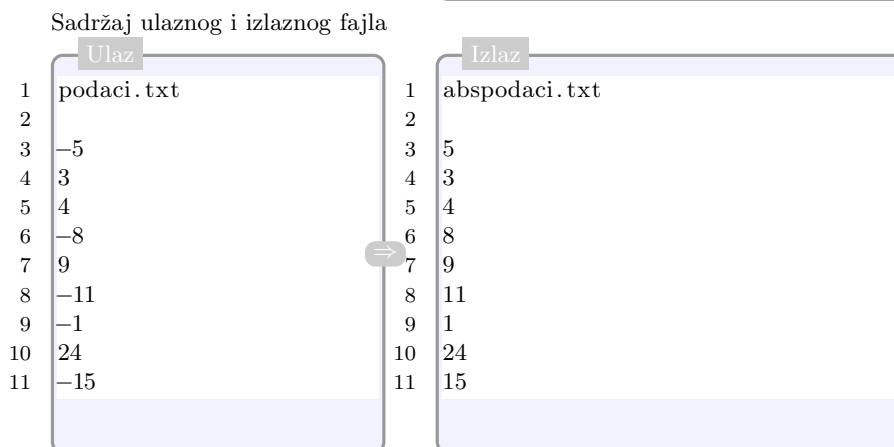
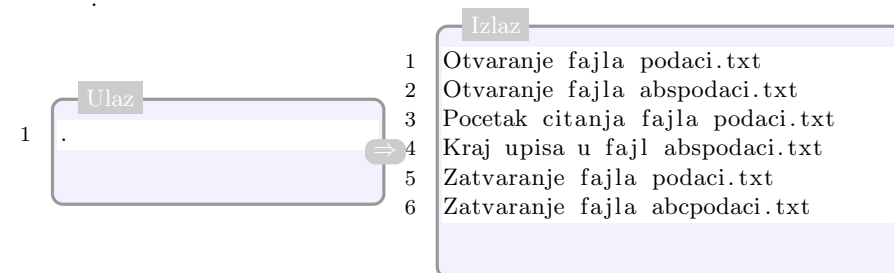
```
1 #include <stdio.h>
```

Zbirka zadataka

```

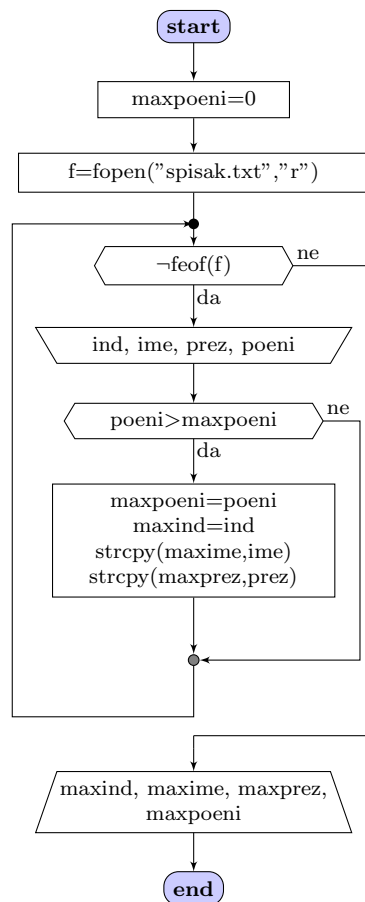
2  #include <math.h>
3  void main()
4  {
5      int br;
6      FILE *fp,*fa;
7      printf("Otvaranje fajla podaci.txt\n");
8      fp = fopen("podaci.txt", "r");
9      printf("Otvaranje fajla abspodaci.txt\n");
10     fa = fopen("abspodaci.txt", "w");
11     printf("Pocetak citanja fajla podaci.txt\n");
12     while (!feof(fp))
13     {
14         fscanf(fp, "%d", &br);
15         fprintf(fa, "%d\n", abs(br));
16     }
17     printf("Kraj upisa u fajl abspodaci.txt\n");
18     printf("Zatvaranje fajla podaci.txt\n");
19     fclose(fp);
20     printf("Zatvaranje fajla podaci.txt\n");
21     fclose(fa);
22 }

```



**Zadatak 3.85**

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje i prikazuje podatke o studentu sa najviše poena na pisanom delu ispita iz predmeta Algoritmi i programiranje. Spisak studenata se nalazi u tekstualnom fajlu *"spisak.txt"*, a formatiran je tako da svaka linija teksta sadrži podatke o jednom studentu i to: broj indeksa, ime, prezime, i broj poena u opsegu 0 do 100. Podaci su razdvojeni sa po jednim blanko znakom.

**Rešenje**

```

1 #include <stdio.h>
2 #include <string.h>
3 void main()
4 {

```

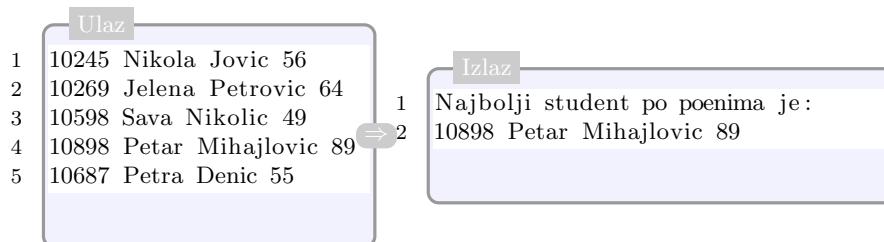
Zbirka zadataka



```

5      char ime[25], prez[25], maxime[25], maxprez[25];
6      int ind, poeni, maxind, maxpoeni = 0;
7      FILE *f = fopen("spisak.txt", "r");
8      while ( !feof(f) )
9      {
10         fscanf ( f, "%d%s%s%d", &ind, ime, prez, &poeni );
11         if ( poeni > maxpoeni )
12         {
13             maxpoeni = poeni;
14             maxind = ind;
15             strcpy (maxime, ime);
16             strcpy (maxprez, prez);
17         }
18     }
19     printf("Najbolji student po poenima je: \n%d%s%s%d", maxind,
20         maxime, maxprez, maxpoeni);

```



### 3.5 Fajlovi u funkciji

#### Zadatak 3.86

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati funkciju koja u tekstualnom fajlu čije se ime prenosi preko parametara određuje koliko puta se javlja slovo takode preneto preko parametara funkcije. U glavnom programu korišćenjem formirane finkcije odrediti i prikazati koji se od karaktera 'a', 'b' i 'c' najčešće javlja u fajlu *"ispit.txt"*. Nije dozvoljeno korišćenje indeksiranih promenljivih.

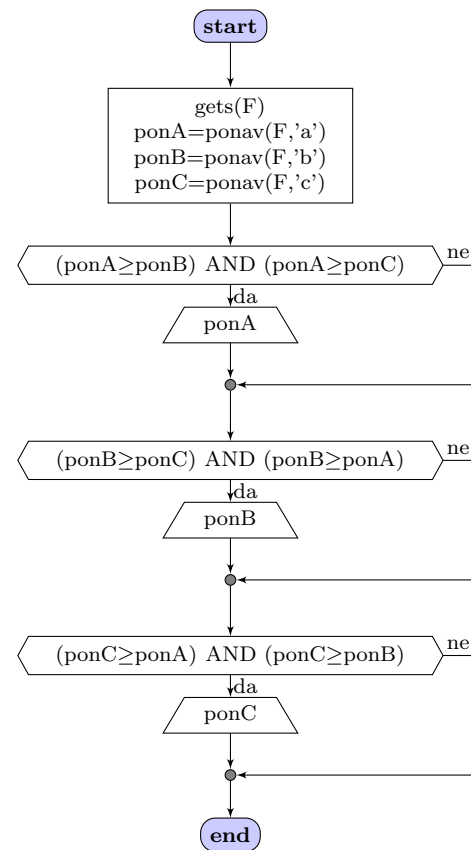
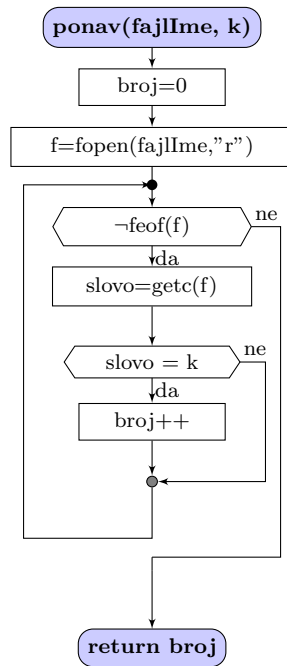
#### Rešenje

```

1  #include <stdio.h>
2  int ponav(char* fajlIme, char k)
3  {
4      int broj = 0;
5      FILE* f = fopen(fajlIme, "r");
6      while (!feof(f))

```

Zbirka zadataka



```

7      {
8          char slovo = getc(f);
9          if (slovo == k)
10             broj++;
11     }
12     return broj;
13 }
14
15 void main()
16 {
17     char F[25];
18     int ponA, ponB, ponC;
19     gets(F);
20     ponA = ponav(F, 'a');
21     ponB = ponav(F, 'b');
22     ponC = ponav(F, 'c');
23     if ((ponA >= ponB) && (ponA >= ponC))

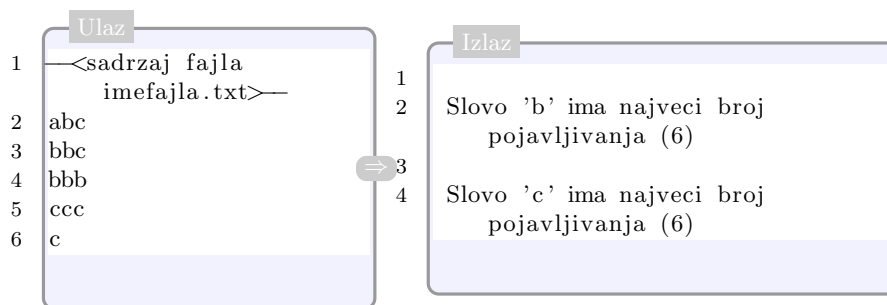
```

Zbirka zadataka

```

24     // >= umesto > da bi se pokrio i slucaj kada vise slova ima isti
    maksimalni broj pojavljivanja
25     printf("\nSlovo 'a' ima najveći broj pojavljivanja
    (%d)\n", ponA);
26     if ((ponB >= ponC) && (ponB >= ponA))
27     printf("\nSlovo 'b' ima najveći broj pojavljivanja
    (%d)\n", ponB);
28     if ((ponC >= ponA) && (ponC >= ponB))
29     printf("\nSlovo 'c' ima najveći broj pojavljivanja
    (%d)\n", ponC);
30 }

```



### Zadatak 3.87

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturnu funkciju koja kopira sadržaj tekstualnog fajla u novi fajl i u toku kopiranja sva pojavljivanja velikog ili malog slova 'Z' zamjenjuje velikim ili malim slovom 'Y' i obrnuto. Ulazni parametri funkcije su imena fajlova. U glavnom programu uneti imena fajlova i izvršiti funkciju. Po obavljenoj obradi ispisati odgovarajuću poruku.

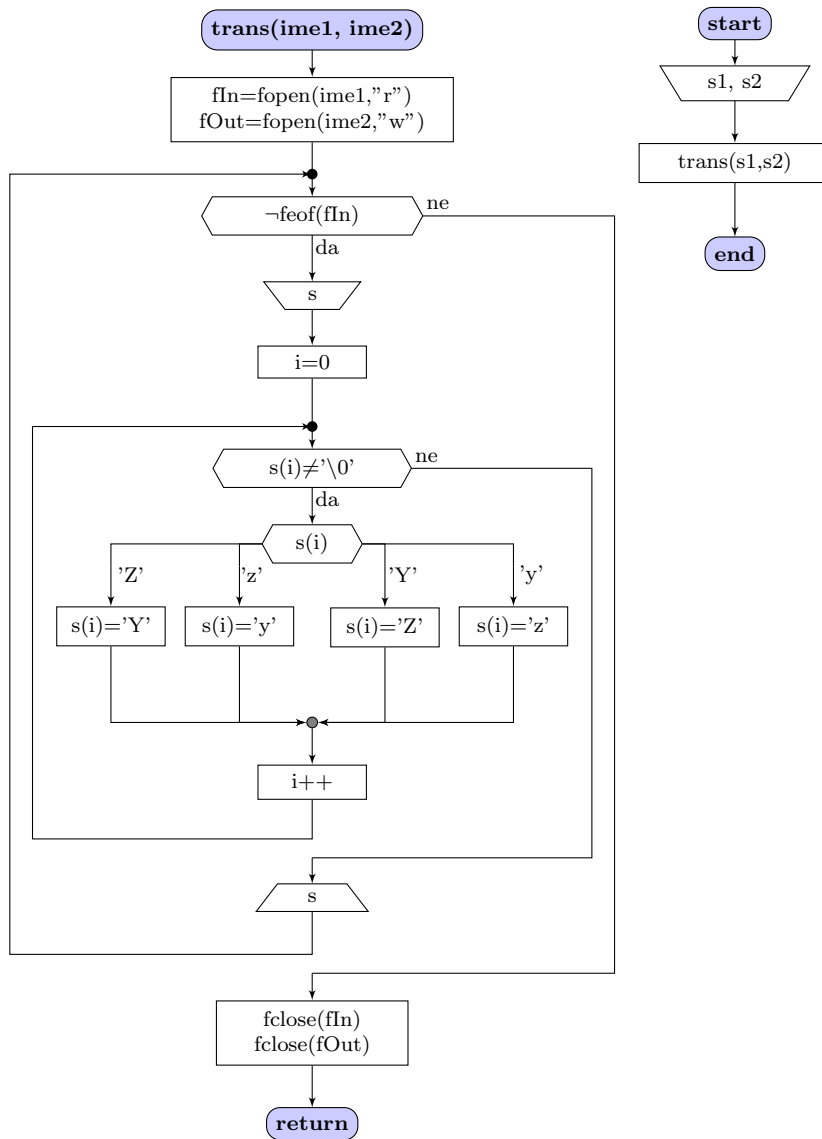
### Rešenje

```

1  #include <stdio.h>
2  #include <string.h>
3  void trans(char* ime1, char* ime2)
4  {
5      int i;
6      char s[100];
7      FILE* fIn = fopen(ime1, "r");
8      FILE* fOut = fopen(ime2, "w");
9      while(!feof(fIn))
10     {
11         fgets(s, 100, fIn);
12         i=0;
13         while(s[i] != '\0')
14         {

```

Zbirka zadataka



```

15      switch(s[i])
16      {
17          case 'Z': s[i] = 'Y'; break;
18          case 'z': s[i] = 'y'; break;
19          case 'Y': s[i] = 'Z'; break;
20          case 'y': s[i] = 'z'; break;
21      }
22      i++;

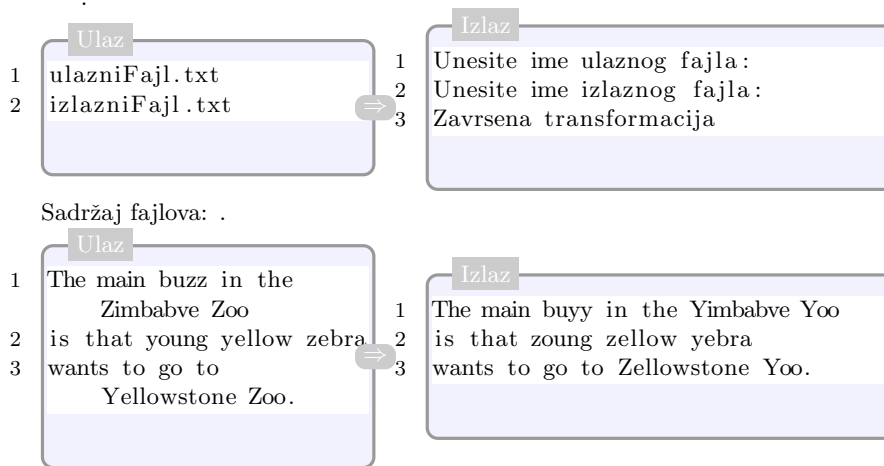
```

Zbirka zadataka

```

23     }
24     fputs(s, fOut);
25 }
26 fclose(fIn);
27 fclose(fOut);
28 }
29 void main()
30 {
31     char s1[20], s2[20];
32     printf("Unesite ime ulaznog fajla:\n");
33     gets(s1);
34     printf("Unesite ime izlaznog fajla:\n");
35     gets(s2);
36     trans(s1, s2);
37     printf("Završena transformacija\n");
38 }

```



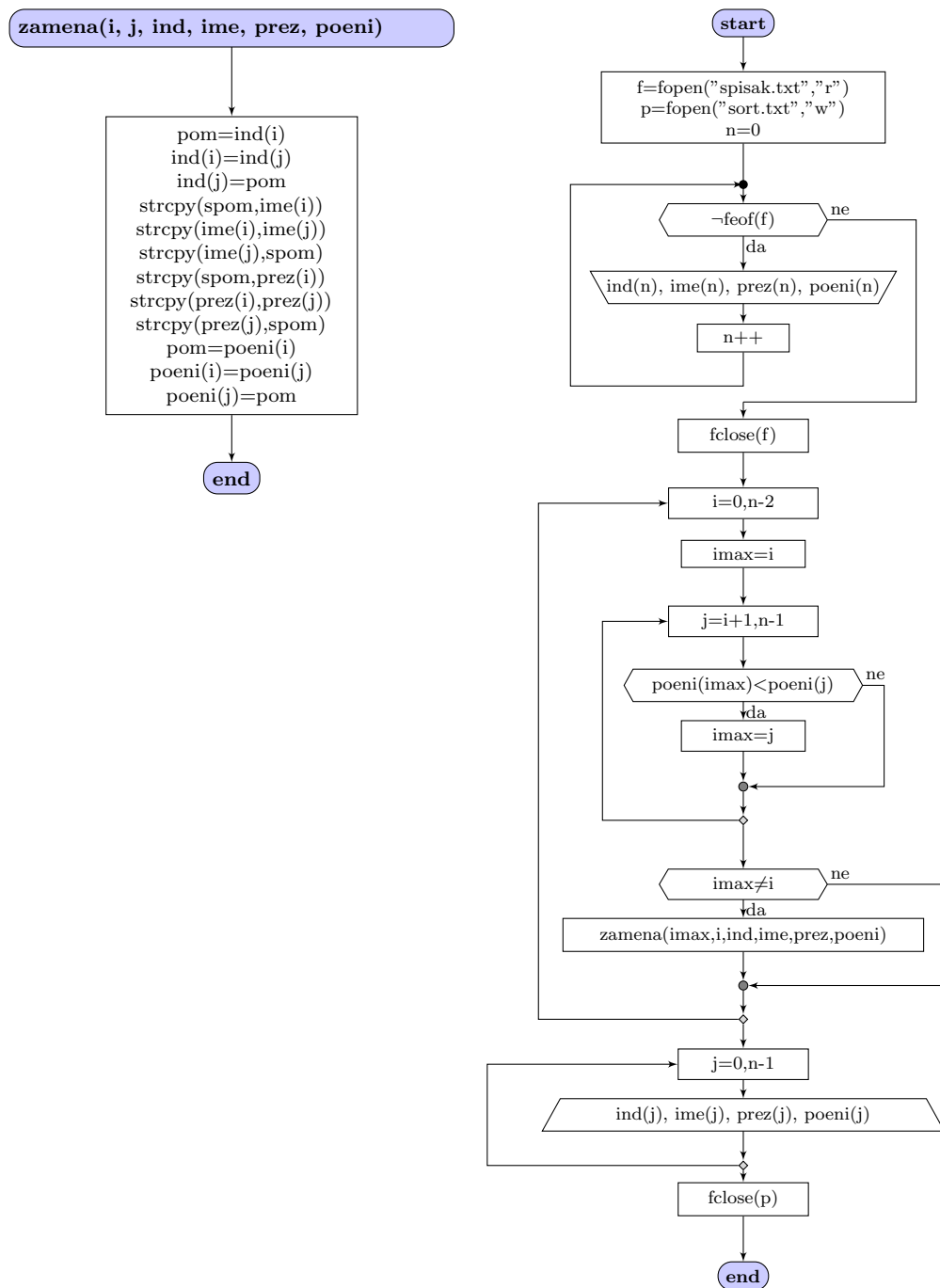
#### Zadatak 3.88

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji spisak studenata uređuje u opadajući redosled po poenima osvojenim na pisanom delu ispita iz predmeta Algoritmi i programiranje. Spisak se nalazi u tekstualnom fajlu *"spisak.txt"*. Na početku svake linije fajla nalazi se broj indeksa, a za njim ime, prezime i broj poena. Podaci su međusobno razdvojeni sa po jednim blanko znakom. Rezultujući spisak snimiti u fajl *"sort.txt"*. Za zamenu mesta podacima napisati funkciju za zamenu.

#### Rešenje

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

Zbirka zadataka

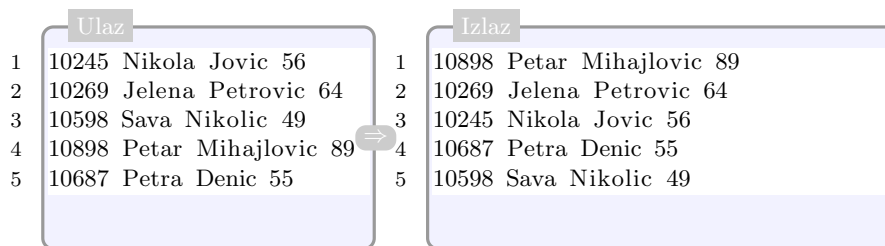


```

2  #include <string.h>
3  void zamena(int i, int j, int ind[100], char ime[100][20], char
    prez[100][20], int poeni[100])
4  {
5      int pom;
6      char spom[20];
7      pom = ind[i];
8      ind[i] = ind[j];
9      ind[j] = pom;
10     strcpy(spom, ime[i]);
11     strcpy(ime[i], ime[j]);
12     strcpy(ime[j], spom);
13     strcpy(spom, prez[i]);
14     strcpy(prez[i], prez[j]);
15     strcpy(prez[j], spom);
16     pom = poeni[i];
17     poeni[i] = poeni[j];
18     poeni[j] = pom;
19 }
20 void main()
21 {
22     int ind[100], poeni[100], read, i, j, n, imax;
23     char ime[100][20], prez[100][20];
24     FILE *f, *p;
25     f = fopen("spisak.txt", "r");
26     p = fopen("sort.txt", "w");
27     n=0;
28     while( !feof( f ))
29     {
30         fscanf( f, "%d %s %s %d", &ind[n], ime[n], prez[n], &poeni[n] );
31         n++;
32     }
33     fclose(f);
34     for ( i = 0; i < n-1; i++ )
35     {
36         imax = i;
37         for( j = i + 1; j < n; j++ )
38             if( poeni[imax] < poeni[j] )
39                 imax = j;
40         if( imax != i )
41             zamena (imax, i, ind, ime, prez, poeni);
42     }
43     for ( j = 0; j < n; j++ )
44         fprintf (p, "%d %s %s %d\n", ind[j], ime[j], prez[j], poeni[j]);
45     fclose(p);
46 }

```

*Zbirka zadataka*





# Literatura

- [1] Brian W. Kernighan, Dennis M. Ritchie, *The C Programming Language*, Prentice Hall, Englewood Cliffs, New Jersey, 1988.
- [2] Vladan Vujičić, *Uvod u C jezik*, Institut za nuklearne nauke "Boris Kidrič", Vinča, 1991.
- [3] *MS-C Reference*, Microsoft Corporation, V6.0, 2012, <http://msdn.microsoft.com/en-us/library/ms950410.aspx>.