



UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
KATEDRA ZA PRIMENJENE RAČUNARSKE NAUKE

Osnovi programiranja i programskih jezika

Računarske vežbe – vežba 3

Zimski semestar 2024/2025.

Studijski program: Informacioni inženjering

Znakovni tip, kontrola toka i nizovi

Znakovni tip

Karakter (znakovi):

'A', 'B', 'c', 'd', '1', '4', '#', '?', '\n'

char promenljiva se koristi za čuvanje znakova:

- slova
- cifara
- specijalnih znakova
- neštampanih (belih) znakova
- može se koristiti i za čuvanje malih celobrojnih vrednosti (0 do 255 ili -128 do 127).

char je zapravo broj

- Svakom karakteru se pridružuje numerički kod
- Postoje različiti skupovi kodova:
 - **ASCII** (American Standard Code for Information Interchange) – najčešći
 - EBCDIC – zastareo, danas se retko koristi
 - Noviji skupovi karaktera (Unicode)
- U jeziku C koristi se ASCII

Primer 1

```
// upotreba char kao znakovnog tipa
// i za malu numericku vrednost

#include <stdio.h>

int main() {
    char znak;
    printf("Unesite znak: ");
    scanf("%c", &znak);
    printf("Znak kao karakter je: '%c'\n", znak);
    printf("Numericka vrednost znak-a je: %hhd\n", znak);
    printf("Karakter posle '%c' je '%c'\n", znak, znak+1);
    return 0;
}
```

Selekcija u jeziku C

- Omogućava da se određene naredbe izvrše uslovno u zavisnosti od vrednosti određenog iskaza
- Iskazi selekcije (grananja):
 - **if - else**
 - **switch - case**

if-else iskaz

```
if (izraz)
    naredba1
else
    naredba2
```

- Ako je izraz tačan (vrednost izraza različita od nule), izvršava se *naredba1*.
- Ako je izraz netačan (vrednost izraza jednaka nula) izvršava se *naredba2*.
- Napomena: *naredba#* može da predstavlja i blok naredbi.
- Nepotpuni **if** iskaz – bez odgovarajućeg *else*

Ternarni operator

- *uslov* ? *izraz1* : *izraz2*
- Funkcioniše kao **if-else**

```
min = ( x < y ) ? x : y;  ⇔  if (x < y)
                               min = x;
                               else
                               min = y;
```

– ? : je operator, **if-else** je iskaz

Zadatak I.

Napisati C program kojim se za zadati broj x , određuje i prikazuje vrednost broja x i njegova apsolutna vrednost $|x|$.

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

Zadatak 2.

Napisati C program koji za dve ulazne celobrojne promenljive, određuje i ispisuje vrednost manje na standardni izlaz.

Zadatak za domaći I.

Napisati C program koji za tri ulazne celobrojne promenljive, određuje i ispisuje vrednost najmanje na standardni izlaz.

Zadatak 3.

Napisati C program koji za godinu unesenu sa standardnog ulaza određuje i ispisuje na standardni izlaz da li je godina prestupna. Godina se računa prestupnom ako je:

- deljiva sa 400, a ako nije onda ako je
- deljiva sa 4, ali ne i sa 100.

Zadatak za domaći 2.

Napisati C program koji za broj unesen sa standardnog ulaza određuje i ispisuje na standardni izlaz:

- broj uvećan za 3 ako je paran,
- poruku "Neparan broj..." ako je neparan.

switch iskaz

- Višestruki uslovni iskaz, sličan **if-else if-else ...** konstrukciji
- Omogućava izbor proizvoljnog broja opcija. Izbor je zasnovan na celobrojnoj vrednosti.

```
switch (izraz) {  
  case const-izraz-1:  
    naredbe-1  
  case const-izraz-2:  
    naredbe-2  
  
  ...  
  default:  
    naredbe-podrazumevane  
}
```

Primer 2

```
switch (ocena) {  
    case 5: printf("Odlican!\n\n");  
        break;  
    case 4: printf("Vrlo dobar!\n\n");  
        break;  
    case 3: printf("Dobar!\n\n");  
        break;  
    case 2: printf("Dovoljan!\n\n");  
        break;  
    case 1: printf("Nedovoljan!\n\n");  
        break;  
    default: printf("Ocena mora biti izmedju 1 i 5.");  
}
```

Naredba **break**

- Kada se naiđe na **break**, blok naredbi se napušta bez obzira na trenutnu vrednost uslova
- Program nastavlja sa izvršavanjem prve naredbe iza bloka
- Ukoliko se pozove u ugnježenim blokovima, **break** izlazi samo iz unutrašnjeg bloka

Zadatak 4.

Realizovati *Primer 2* uz pomoć **if** selekcije.

Zadatak 5.

Nacrtati dijagram toka algoritma i napisati C program kojim se za zadato x izračunava $f(x)$ po formuli:

$$f(x) = \begin{cases} -7, & x < 0 \\ x^{1/4} + 4, & 0 \leq x < 1 \\ 2\sqrt{x} - 5, & 1 \leq x < 13 \\ x^2/9, & x \geq 13 \end{cases}$$

Zadatak za domaći 3.

Nacrtati dijagram toka algoritma i napisati C program koji učitava vrednosti promenljivih a i b , a zatim rešava jednačinu $ax + b = 0$.

Pokriti sve slučajeve za različite kombinacije vrednosti a i b .

Zadatak za domaći 4.

Tri tačke su zadate koordinatama u ravni. Nacrtati dijagram toka algoritma i napisati C program kojim se izračunava površina paralelograma čiji su vrhovi date tačke.

Zadatak za domaći 5.

Napisati C program koji za tri ulazne pozitivne celobrojne vrednosti određuju i na ekran ispisuju srednju vrednost po veličini.

Zadatak 6.

Napisati C program kojim se unosi brojevi a , b i c i ispituje da li je moguće formirati trougao sa dužinama stranica koje su a , b i c .

Zadatak 7.

Bankomat: Napisati C program kojim se sa tastature unose trenutni iznos (u RSD) na računu korisnika u banci, kao i iznos koji korisnik želi da podigne sa bankomata. Ukoliko je iznos deljiv sa 500 onda je transakciju moguće uspešno obaviti i sa provizijom banke od 15 RSD. U suprotnom, transakcija je neuspešna i stanje na računu ostaje nepromenjeno. Ipisati poruku o uspešnosti transakcije kao i preostali iznos na računu.

Primer:

```
Ulaz: 77200 2500  
Izlaz: USPESNO 74685
```

```
Ulaz: 45900 1400  
Izlaz: NEUSPESNO 45900
```

Iteracija `while`

```
while (izraz) {  
    naredbe;  
}
```

Naredbe se izvršavaju sve dok je vrednost izraza tačna (različita od 0), kada to više nije slučaj izlazi se iz petlje.

Primer 3

```
/* Stampa farenhajt-celzijus tabelu za far = 0, 20, ..., 300; */
#include <stdio.h>

int main() {
    float far, cel;
    float donja, gornja, korak;

    donja = 0;    // donja granica temperaturnog opsega
    gornja = 300; // gornja granica
    korak = 20;  // velicina koraka

    far = donja;
    while (far <= gornja) {
        cel = (5.0/9.0) * (far-32.0);
        printf("%3.0f %6.1f\n", far, cel);
        far = far + korak;
    }
    return 0;
}
```

Brojačka iteracija **for**

- **for** petlje se kontrolišu brojačkom promenljivom

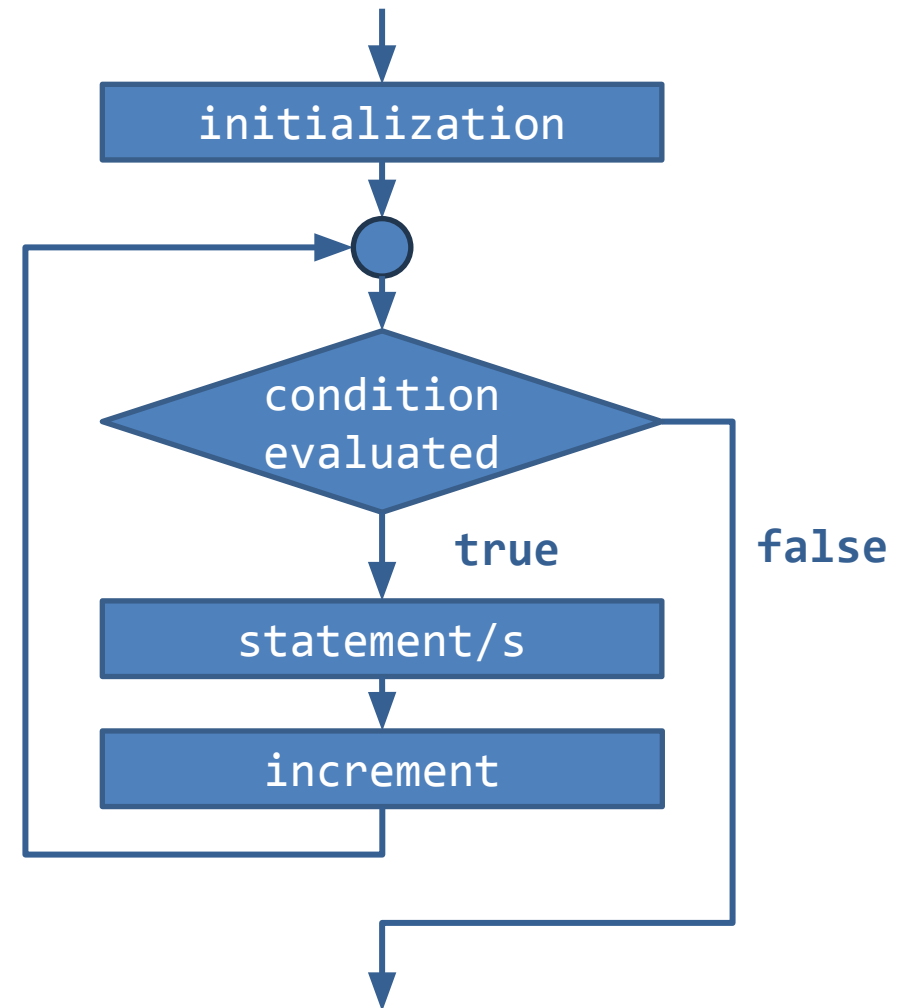
- Primer:

```
for (i = 0; i <= 50; i++) {  
    printf("%d\n", i);  
}
```

for vs while

```
for (initialization; expression; increment)  
    statement;
```

```
initialization;  
while (expression) {  
    statement;  
    increment;  
}
```



Zadatak 8.

Napisati C program koji računa sumu prvih n prirodnih brojeva, pri čemu se n zadaje na početku programa.

Zadatak 9.

Napisati C program za sumiranje brojeva koji se unose na ulazu i čuvaju se u promenljivoj b . Unos brojeva i njihovo sumiranje se vrše sve dok suma ne postane veća od vrednosti zapamćene u promenljivoj g koja se takođe zadaje na ulazu. Na izlazu se prikazuje izračunata suma.

Zadatak za domaći 6.

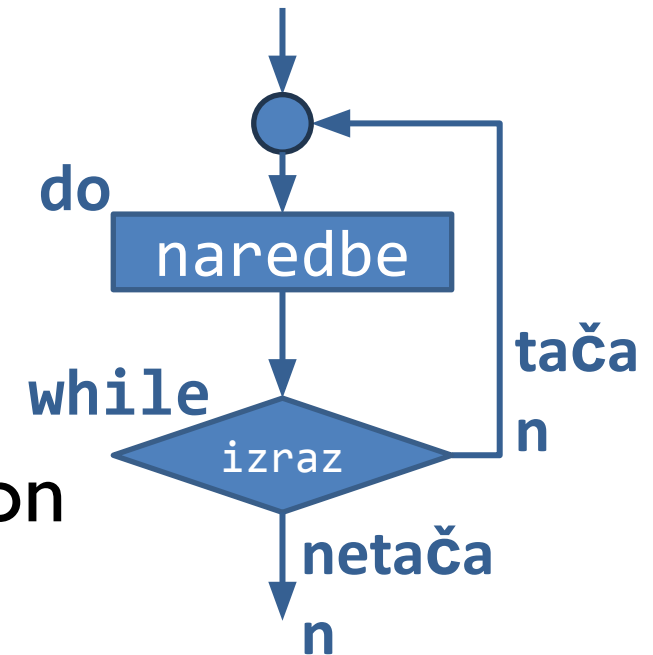
Napisati C program koji proverava da li korisnik zna da sabira. Nasumično se generišu 2 cela broja iz $[0, 100]$ i potom se od korisnika traži da unese tačan zbir brojeva. Svaki put kada ne unese tačan zbir, program ispisuje "Smotan si... Budi više kao FTN moto!". Ako korisnik iz prvog pokušaja upiše rezultat kako treba, ispisuje se poruka "Odlicno!", ako pogreši 1-2 "Nije strašno." a u suprotnom "Krajnje vreme je bilo...".

Iteracija **do-while**

```
do {  
    naredbe  
} while (izraz);
```

Sličan **while** iteraciji:

- Vrednost izraza se izračunava nakon tela ciklusa
- Telo ciklusa se izvršava **NAJMANJE** jednom, čak i kada je izraz odmah netačan (jednak nuli)



Iteracija do-while

```
int broj;  
  
do {  
    printf("Unesite pozitivan ceo broj:");  
    scanf("%d", &broj);  
} while (broj <= 0);
```

Iteracija `do-while`

```
int broj;  
  
do {  
    printf("Unesite negativan paran broj:");  
    scanf("%d", &broj);  
} while ( ##### );  
// ##### = ?
```

Iteracija `do-while`

```
int broj;  
  
do {  
    printf("Unesite negativan broj deljiv sa 3:");  
    scanf("%d", &broj);  
} while ( ##### );  
// ##### = ?
```

Naredbe **break** i **continue**

- **break** omogućava da se prekine izvršavanje petlje
- **continue** omogućava da se (odmah) pređe na narednu iteraciju

Zadatak 10.

Napisati C program kojim se računa specifična suma prvih n prirodnih brojeva. Vrednost n se unosi sa standardnog ulaza. Specifična suma se računa na sledeći način:

- Brojevi se redom dodaju na sumu,
- Ukoliko je broj jednak trenutnoj sumi, broj se preskače,
- Ukoliko u bilo kom momentu suma bude jednaka vrednosti n sumiranje se zaustavlja.

Na kraju programa ispisati vrednost sume.

Zadatak I I.

Nacrtati dijagram toka algoritma i napisati C program kojim se za dato n izračunava faktorijel prirodnog broja n . Koristeći **do-while** napraviti zaštitu za unos broja n :

$$\begin{aligned} \text{faktorijel}(n) &= n! = 1 * 2 * 3 * \dots * n, \\ 0! &= 1 \end{aligned}$$

Zadatak 12.

Nacrtati dijagram toka algoritma i napisati C program kojim se sa standardnog ulaza učitavaju vrednosti prirodnih brojeva n i q . Potom, koristeći iteraciju **while** ispisati sve brojeve od 2 do n koji su deljivi sa q .

Zadatak za domaći 7.

Napisati C program za generisanje i prikazivanje prvih n članova Fibonačijevog niza definisanog sledećom formulom:

$$f_n = f_{n-1} + f_{n-2}, n \geq 2$$

gde su $f_0 = 0$ i $f_1 = 1$. Vrednost n se zadaje kao ulaz.

Zadatak za domaći 8.

Napisati C program koji ispituje da li je dati prirodan broj n prost. Broj je prost ako je deljiv samo sa jedan i sa samim sobom.

Zadatak 13.

Nacrtati dijagram toka algoritma i napisati C program kojim se štampaju svi trocifreni Armstrongovi brojevi. Broj je Armstrongov ako je jednak zbiru kubova svojih cifara.

Zadatak 14.

Nacrtati dijagram toka algoritma i napisati C program koji ispisuje prvih n prostih brojeva.

Zadatak za domaći 9.

Nacrtati dijagram toka algoritma i napisati C program koji ispisuje sve proste brojeve manje od n .

Zadatak za domaći 10.

Nacrtati dijagram toka algoritma i napisati C program koji za dati prirodan broj n izračunava:

$$s = 1 * 2 + 2 * 3 * 4 + \dots + (n + 0) * (n + 1) * \dots * (n + n)$$

Nizovi - deklaracija

Niz predstavlja kolekciju elemenata istog tipa. Primer deklaracije niza je:

```
int niz1[5]; //niz od 5 elemenata tipa int
```

```
float niz2[12]; //niz od 12 elemenata tipa float
```

Pristupanje elementima niza

Pristupanje elementima niza se ostvaruje na sledeći način:

- `niz[0] = 4;`
- `niz[1] = 2 * niz[0]; /*niz[1] = 8*/`
- `niz[2] = niz[0] * niz[1]; /*niz[2] = 32*/`
- `niz[3] = 5;`
- `niz[4] = 7;`
- `a = niz[10];`

Napomena: Indeks niza ide od 0 (a ne od 1!) do $n-1$

Inicijalizacija niza

Primer inicijalizacije vrednosti elemenata niza

```
for (i = 0; i < 5; i++)  
    niz[i] = 0;
```

Inicijalizacija niza

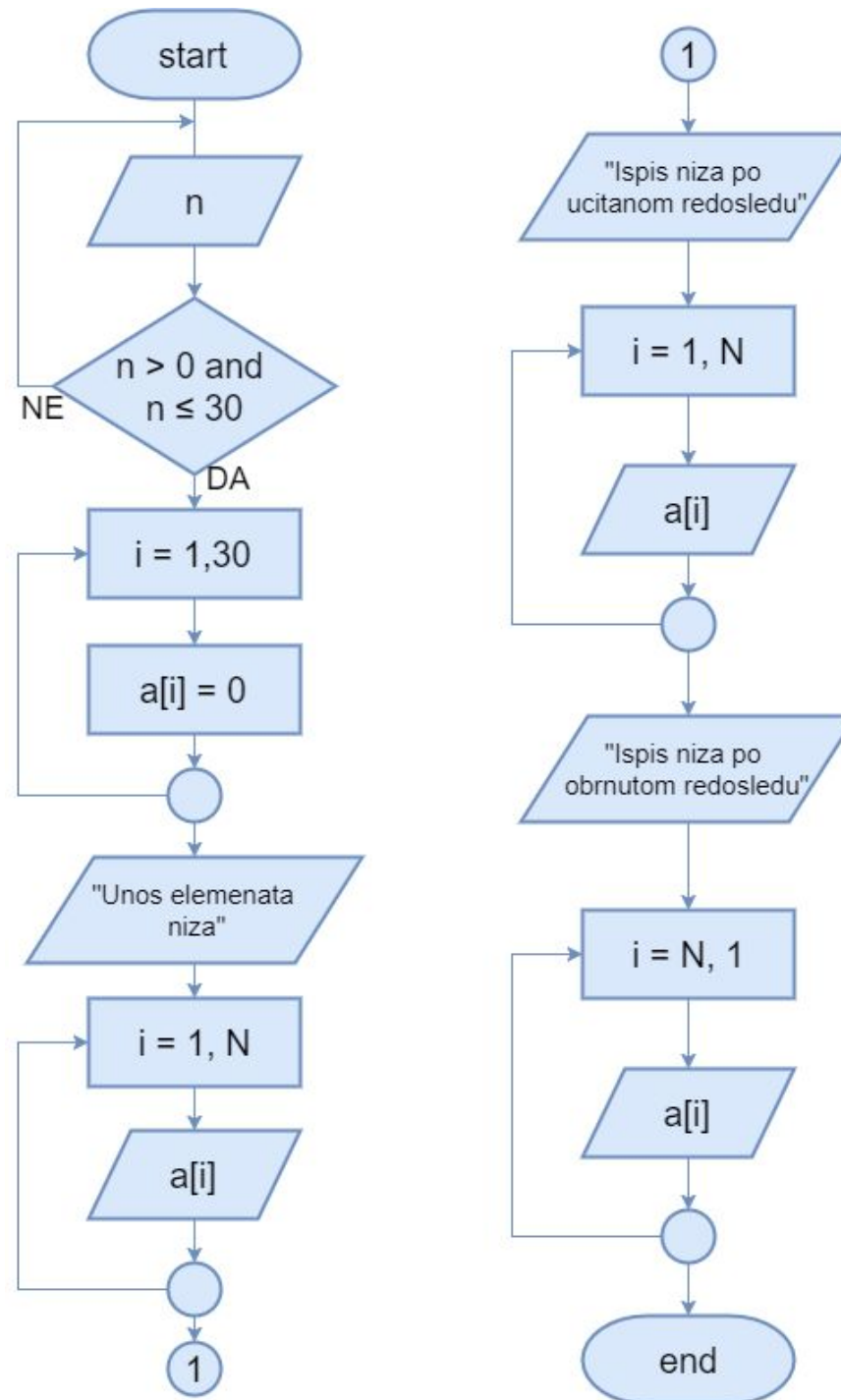
- Opšta sintaksa:
 - `tip_podatka ime[veličina] = {vred0, vred1, ... vredn-1};`
- Primer:
 - `int dani_u_mesecu[12] = {31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};`
- Broj elemenata za inicijalizaciju može da bude manji od veličine niza.
 - ostatak se inicijalizuje nulama
- Primer:
 - `int dani_u_mesecu[12] = {31, 29, 31, 30, 31, 31, 30};`

Inicijalizacija niza

- Dimenzija niza se može izostaviti ako se niz inicijalizuje
 - veličina niza je broj elemenata za inicijalizaciju
- Primer:
 - `int dani_u_mesecu[] = {31, 29, 31, 30, 31, 31, 30, 31, 30, 31};`

Primer 4

Dat je niz od maksimalno 30 celobrojnih elemenata. Nacrtati dijagram toka algoritma i napisati C program kojim se učitava n elemenata i elementi se potom ispisuju po učitanom i obrnutom redosledu.



```
1  #include <stdio.h>
2  #define MAX_SIZE 30
3
4  int main()
5  {
6      int a[MAX_SIZE];
7      int i, n;
8
9      do {
10         printf("Unesite broj elemenata niza (maksimalno %d): ", MAX_SIZE);
11         scanf("%d", &n);
12     } while ( n <= 0 || n>MAX_SIZE);
13
14     for (i = 0; i<MAX_SIZE; i++) {
15         a[i] = 0;
16     }
17
18     for (i = 0; i<n; i++) {
19         printf("a[%d]=", i);
20         scanf("%d", &a[i]);
21     }
22
23     printf("\n Elementi niza po ucitanom redosledu: \n");
24
25     for (i = 0; i<n; i++) {
26         printf("a[%d]=%d\n", i, a[i]);
27     }
28
29     printf("\n Obrnuti redosled: \n");
30
31     for (i = n-1; i>=0; i--) {
32         printf("a[%d]=%d\n", i, a[i]);
33     }
34
35     return 0;
36 }
```

Zadatak 15.

Dat je niz $nizA$ od maksimalno 30 celobrojnih elemenata. Nacrtati dijagram toka algoritma i napisati C program kojim se učitava n elemenata i ceo broj b . Na izlazu ispisati ceo niz i broj pojavljivanja broja b u nizu.

Primer:

$nizA = [2, 5, 6, 2, 8, 9, 2]$

$b = 2$

Rešenje:

"Broj 2 se pojavljuje 3 puta u nizu [2, 5, 6, 2, 8, 9, 2]"

Zadatak 16.

Dat je niz od maksimalno 20 realnih elemenata. Nacrtati dijagram toka algoritma i napisati C program kojim se učitava n elemenata, a zatim nalazi i prikazuje maksimalna i minimalna vrednost u nizu.

Zadatak 17.

Dat je niz $nizC$ od maksimalno 25 celobrojnih elemenata. Nacrtati dijagram toka algoritma i napisati C program kojim se učitava n elemenata u $nizC$ i formiraju nizovi $nizA$ i $nizB$, pri čemu su elementi $nizA$ parni elementi $nizC$, a elementi $nizB$ su negativni elementi $nizC$. Ispisati nizove $nizC$, $nizA$ i $nizB$.

Zadatak 18.

Dat je niz *nizC* od maksimalno 80 karaktera. Nacrtati dijagram toka algoritma i napisati C program kojim se učitava *n* karaktera u *nizC* i formira *nizA*, pri čemu su elementi *nizA* samo slova iz *nizC*.

Zadatak za domaći I I.

Nacrtati dijagram toka algoritma i napisati C program kojim se pronalazi prvi element niza koji je najbliži srednjoj vrednosti niza celih brojeva. Niz može imati najviše 20 elemenata.

Zadatak za domaći 12.

Nacrtati dijagram toka algoritma i napisati C program koji za (sa standardnog ulaza) zadati prirodni broj N (ne veći od 4 milijarde) nalazi i ispisuje sumu prve i poslednje cifre.

Zadatak za domaći 13.

Nacrtati dijagram tok algoritma i napisati C program koji za zadati niz brojeva A (maksimalne dužine 100) određuje dužinu najdužeg neopadajućeg podniza uzastopnih članova.

Na primer:

$A = [1 \ 5 \ 2 \ 4 \ 4 \ 8 \ 2 \ 6 \ 1]$

Najduži neopadajući podniz ima 4 elementa.

Zadatak za domaći 14.

Nacrtati dijagram tok algoritma i napisati C program koji u zadatom nizu celobrojnih vrednosti A (maksimalne dužine 100) zamenjuje mesta prvom parnom i poslednjem neparnom broju.

Zadatak za domaći 15.

Nacrtati dijagram tok algoritma i napisati C program koji od zadatog niza brojeva A (maksimalne dužine 100) formira niz B sačinjen od suma svih susednih parova elemenata niza A.

Na primer:

$$A = [2 \ 6 \ 3 \ 1 \ 8 \ 9 \ 4 \ 6 \ 8]$$

$$B = [8 \ 9 \ 4 \ 9 \ 17 \ 13 \ 10 \ 14]$$