

# Programiranje 2

## Rekurzija

### 1 ZADACI SA ČASA

**Zadatak 1** Napisati rekurzivnu funkciju koja prikazuje sve cifre datog celog pozitivnog broja  $i$  to:

a) void ispis1(int x) - koja ispisuje s leva na desno;

b) void ispis2(int x) - koja ispisuje s desna na levo.

<b>Primer 1:</b> a) 34021  3 4 0 2 1	<b>Primer 2:</b> b) 56219  9 1 2 6 5
--	--

**Zadatak 2** Napisati rekurzivnu funkciju **int broj\_parnih(int x)** koja računa broj parnih cifara datog celog broja  $x$ .

**Zadatak 3** Napisati rekurzivnu funkciju **int najveca\_cifra(int x)** koja računa najveću cifru datog celog broja  $x$ .

**Zadatak 4** Napisati rekurzivnu funkciju **int ukloni(int x, int c)** koja uklanja sva pojavljivanja date cifre  $c$  iz datog broja  $x$ .

**Zadatak 5** Napisati rekurzivnu funkciju **void napravi\_niz(int\* kraj\_niza, int broj)** koja kreira niz cifara datog celog broja. Napisati rekurzivnu funkciju **void ispisi\_niz(int\* niz, int n)** koja ispisuje elemente niza duzine  $n$ . Testirati obe funkcije pozivom iz glavog programa.

**Zadatak 6** Napisati rekurzivnu funkciju **void heksa\_zapis(int x)** koja određuje heksadekadni zapis datog celog broja.

<b>Primer 1:</b> 2356  934	<b>Primer 2:</b>
-------------------------------------	------------------

**Zadatak 7** Napisati rekurzivnu funkciju **int obrni(int x, int p)** koja obrće cifre datog celog broja  $x$ .

**Zadatak 8** Napisati rekurzivnu funkciju **void obrni\_niz(int\* a, int n)** koja obrće niz brojeva (niz alocirati dinamički).

**Zadatak 9** Napisati rekurzivnu funkciju `int palindrom(int* a, int n)` koja ispituje da li su elementi nekog niza brojeva poredani palindromski (isto od napred i od pozadi).

**Zadatak 10** Napisati rekurzivnu funkciju `int dodaj_0(int x)` koja posle svake neparne cifre datog broja dodaje 0.

**Zadatak 11** Napisati rekurzivnu funkciju `int skalarni_proizvod(int* a, int* b, int n)` koja izračunava skalarni proizvod dva data vektora (predstavljena nizovima dužine  $n$ ).

**Zadatak 12** Napisati rekurzivnu funkciju `void trougao(int n, int trenutna_linija)` koja za dato  $n$  iscrta trougao dimenzije  $n$ . Napr za  $n=5$ :

```
+
++
+++
++++
+++++
```

**Zadatak 13 (Kolokvijum 15/16)** Napisati rekurzivnu funkciju `int f3(int x)` koja u početnom zapisu broja  $x$  izbacuje svaku neparnu cifru, ukoliko se ispred te cifre nalazi cifra 2. Napisati program koji testira ovu funkciju, tako što sa standardnog ulaza učitava ceo broj  $x$ , i na standardni izlaz ispisuje vrednost funkcije  $f3(x)$ . U slučaju greške ispisati -1 na standardni izlaz za greške.

**Napomena:** Zadatak mora biti uradjen rekurzivno. Nije dozvoljeno korišćenje statičkih i globalnih promenljivih, menjanje zaglavlja funkcije i pisanje pomoćnih funkcija.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:
325	23523	12345	0	-235524
32	252	1245	0	-25524

**Zadatak 14 (Test 14/15)** Sa standardnog ulaza se učitava ceo broj  $x$ , ceo broj  $n$  ( $n \leq 100$ ), a potom i niz od  $n$  celih brojeva. Napisati rekurzivnu funkciju `void f(int a[], int n, int x)`, koja u nizu  $a$  poslaavlja na nulu sve parove susednih elementa čiji je zbir (u početnom nizu) jednak  $x$ . Rezultujući niz ispisati na standardni izlaz. U slučaju greške ispisati -1 na standardni izlaz za greške.

**Napomena:** Zadatak mora biti uradjen rekurzivno. Nije dozvoljeno korišćenje statičkih i globalnih promenljivih, menjanje zaglavlja funkcije i pisanje pomoćnih funkcija.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
3 9	2 2	5 5	7 8
1 2 7 6 0 3 2 1 4	1 1	8 2 3 2 9	1 5 2 6 3 4 3 9
0 0 7 6 0 0 0 0 4	0 0	8 0 0 0 9	1 0 0 6 0 0 0 9

## 2 DOMAĆI ZADACI

**Zadatak 15** Napisati rekurzivnu funkciju `int izbaci_parne(int x)` koja izbacuje sve parne cifre datog celog broja  $x$ .

**Zadatak 16** Napisati rekurzivnu funkciju `int maksimum_niza(int* a, int n)` koja određuje maksimum niza celih brojeva.

**Zadatak 17 (Kolokvijum 15/16)** Napisati rekurzivnu funkciju `void f3(int *a, int na)` koja u početnom nizu  $a$ , dužine  $na$ , smanjuje vrednost broja za 1 ukoliko je on paran, i nakon njega se nalazi paran broj. Kreirati program koji testira ovu funkciju, tako što sa standardnog ulaza učitava broj  $na$  ( $0 \leq na \leq 1000$ ), zatim vrednosti niza  $a$ , i na standardni izlaz ispisuje vrednosti izmenjenog niza. U slučaju greške ispisati -1 na standardni izlaz za greške.

**Napomena:** Zadatak mora biti urađen rekurzivno. Nije dozvoljeno korišćenje statičkih i globalnih promenljivih, menjanje zaglavlja funkcije i pisanje pomoćnih funkcija.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
10	5	10	-1
2 2 5 7 6 8 4 3 2 1	1 2 2 2 1	4 4 6 6 6 6 8 8 10 10	
1 2 5 7 5 7 4 3 2 1	1 1 1 2 1	3 3 5 5 5 5 7 7 9 10	-1

**Zadatak 18 (Kolokvijum 15/16)** Napisati rekurzivnu funkciju `void f3(int *a, int na, int suma prethodnih)` koja u nizu  $a$ , dužine  $na$ , postavlja vrednost svakog broja na 0 ukoliko je veći od sume prethodnih brojeva u nizu. Napisati program koji testira ovu funkciju, tako što sa standardnog ulaza učitava broj  $na$ , zatim vrednosti niza  $a$ , i na standardni izlaz ispisuje izmenjeni niz. U slučaju greške ispisati -1 na standardni izlaz za greške.

**Napomena:** Zadatak mora biti urađen rekurzivno. Nije dozvoljeno korišćenje statičkih i globalnih promenljivih, menjanje zaglavlja funkcije i pisanje pomoćnih funkcija.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
10	5	2	-1
2 2 5 7 6 25 4 3 2 1	1 2 4 12 9	1 1	
0 2 0 7 6 0 4 3 2 1	0 0 0 0 9	0 1	-1

**Zadatak 19 (Test 14/15)** Sa standardnog ulaza se učitava ceo broj  $n$  ( $n \leq 100$ ), a potom i niz od  $n$  celih brojeva. Napisati rekurzivnu funkciju `void f(int a[], int n)` koja u nizu  $a$  postavlja na nulu sve elemente koji su (u početnom nizu) jednaki zbiru svojih suseda. Rezultujući niz ispisati na standardni izlaz. U slučaju greške ispisati -1 na standardni izlaz za greške.

**Napomena:** Zadatak mora biti urađen rekurzivno. Nije dozvoljeno korišćenje statičkih i globalnih promenljivih, menjanje zaglavlja funkcije i pisanje pomoćnih funkcija.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
6	5	2	6
1 4 3 6 3 3	1 0 3 3 0	1 1	1 2 1 1 3 2
1 0 3 0 3 3	1 0 0 0 0	1 1	1 0 1 1 0 2