

Rekurzija

1. Napisati rekurzivnu funkciju **int evenDigits(int n)** koja izbacuje sve parne cifre datog cijelog broja n.
2. Napisati rekurzivnu funkciju **int oddDigits(int n)**, koja vraća broj koji se dobije tako što se iza svake neparne cifre broja n doda 0.
3. Napisati rekurzivnu funkciju **int digitalRoot(int target)** koja izračunava digitalni korijen prirodnog broja target. Ako je suma cifara s(n) broja n jednocifren broj, tada je s(n) digitalni korijen. Ako s(n) nije jednocifren, tada se postupak ponavlja sa s(n) sve dok se ne dobije jednocifren broj.
4. Napisati funkciju **int pow_rek(int a, int b)** koja rekurzivno izračunava b-ti stepen broja a. Kasnije u glavnoj funkciji za dva uneta prirodna broja ispisati rezultat dobijen pozivom te funkcije.
5. Napisati funkciju **int zbir_cif_rek(int a)** koja rekurzivno izračunava zbir cifara broja a. Kasnije u glavnoj funkciji za uneti celi broj ispisati rezultat dobijen pozivom te funkcije.
6. Napisati funkciju **int dec2bin(int n)** koja rekurzivno prebacuje broj iz sistema sa osnovom 10, u odgovarajući broj sistema sa osnovom 2. U glavnoj funkciji testirati napisanu funkciju, podrazumevati da korisnik unosi broj za koji se treba odraditi konverzija.
7. Napisati rekurzivnu funkciju **double f1(int n)** i iterativnu funkciju **double f1_i(int n)** koja za uneto **n = 6** izracunava sledeci razlomacki izraz:

$$6 + \frac{1}{5 + \frac{1}{4 + \frac{1}{3 + \frac{1}{2 + \frac{1}{1}}}}}$$

8. Napisati rekurzivnu funkciju **double f4(int i, int n)** i iterativnu funkciju **double f4_i(int n)** koja za uneto **n = 6** izracunava sledeci potkoreni izraz:

$$\sqrt{1 + \sqrt{2 + \sqrt{3 + \sqrt{4 + \sqrt{5 + \sqrt{6}}}}}}$$

U glavnoj funkciji, za uneto **n**, u dva reda ispisati rezultat, u prvom iterativno dobijen, u drugom rekurzivno.