

Priprema za prvi kolokvijum iz Programiranja I

1. Napisati program koji od korisnika zahtijeva unos prirodnog broja **N** i koji pronalazi sumu svih prirodnih brojeva **n** ne većih od **N**, takvih da je $n^2 + n$ djeljivo sa 4.
2. Napisati program koji za ulazni podatak ima prirodni broj **N** i koji određuje i štampa koliko puta se u broju **N** pojavljuje cifra **7**. Na primjer, ukoliko se unese **N=1772** onda je broj pojavljivanja cifre 7 jednak **2**.
3. Napisati program koji računa sumu svih prirodnih brojeva manjih od **1000** čiji je zbir cifara jednak **6**. Na izlazu štampati dobijenu sumu.
4. Napisati program koji određuje i ispisuje sve trocifrene prirodne brojeve koji su djeljivi dvocifrenim brojem koji se dobija izostavljanjem cifre stotina. Na primjer, broj **325** je djeljiv brojem **25**.
5. Napisati program koji ispisuje sve trocifrene Armstrong-ove brojeve. Broj je Armstrong-ov ukoliko je zbir kubova njegovih cifara jednak tom broju.
6. Napisati program koji za zadat prirodan broj **n** računa i ispisuje sumu:

$$s = 1 - \frac{1+2}{2!} + \frac{1+2+3}{3!} - \dots + (-1)^{n-1} \frac{1+2+3+\dots+n}{n!}$$

7. Napisati program koji na osnovu unešenog proizvoljnog prirodnog broja pronalazi i ispisuje broj kome je redosljed cifara obrnut u odnosu na unešeni broj. Na primjer, za unešeni broj 4916 program treba da ispiše broj 6194.
8. Napisati program koji učitava prirodan broj **N** i koji formira string **S** koji predstavlja razvoj broja **N** u obliku proizvoda njegovih prostih činilaca. Na primjer, za unešeni broj **420** string **S** bi bio **"2*2*3*5*7"**.
9. Napisati program koji učitava string **S** i koji provjerava da li taj string može predstavljati binarni (oktalni, heksadecimalni) zapis nekog broja. Štampati odgovarajuću poruku.
10. Napisati program koji od korisnika traži unos prirodnog broja, a na izlazu ispisuje string koji predstavlja binarni (oktalni, heksadecimalni) zapis toga broja.
11. Napisati program koji na ulaznom stringu **S** implemetira tzv. **RLE (Run Length Encoding)** kompresiju i ispisuje novodobijeni string. *Napomena:* RLE kompresija se sastoji u zamjeni niza sukcesivnih vrijednosti (elementi koji se uzastopno ponavljaju) tom vrijednošću i brojem uzastopnih ponavljanja te vrijednosti.
Primjer: **"AAAAABBBECCCC" → "A5B3E1C4"**.
12. Napisati program koji od korisnika traži unos stringa i koji na izlazu ispisuje dužinu najduže riječi u tom stringu, pri čemu pod riječju podrazumijevamo najduži niz susednih karaktera koji ne sadrži razmak (Space).
Primjer: Najduža riječ u stringu **"Programski jezik C"** ima dužinu 10.
13. Napisati program koji učitava string **VRIJEME** od 8 karaktera oblika **"HH:MM:SS"**, pri čemu par karaktera **HH** predstavlja broj sati (cijeli broj između 0 i 23), drugi par karaktera **MM** predstavlja broj minuta (cijeli broj između 0 i 59) i treći par karaktera predstavlja broj sekundi (cijeli broj između 0 i 59). Dvotačke služe kao separatori ovih brojeva. Program treba da vrati ukupan broj sekundi određen datim stringom, tj. ukoliko string **VRIJEME** tumačimo kao trenutno vrijeme tokom dana onda program treba da vrati ukupan broj sekundi proteklih u tekućem danu. Ukoliko je **HH>23** ili **MM>59** ili **SS>59** program treba da javi grešku. Nije potrebno provjeravati ispravnost stringa što se tiče drugih nepravilnosti unosa. Pri izradi zadatka koristiti funkciju **atoi**.
Primjer: Ukoliko korisnik unese string **"10:03:54"** onda program treba da vrati broj **36234=10*3600+3*60+54**. Ukoliko korisnik unese string **"57:03:54"** program treba da ispiše **Pogresan unos**.
14. Napisati program koji učitava niz cijelih brojeva **X** i koji ispisuje najduži neopadajući podniz tog niza.
15. Napisati program koji učitava prirodan broj **N**, ne veći od **10**, i koji formira kvadratnu matricu **A**, dimenzija **NxN**, sa vrijednostima elemenata **1** na glavnoj dijagonali, **2** iznad i **3** ispod glavne dijagonale.
16. Napisati program kojim se učitava niz **A=[a₀, a₁, ..., a_{N-1}]** i na osnovu njega se formira kvadratna matrica **C(NxN)** kojoj je prva vrsta niz **A**, druga vrsta se dobija tako što cirkularno rotiramo elemente niza **A** za jedno mjesto ulijevo, tako da prvi element dođe na posljednje mjesto, drugi na prvo mjesto, a posljednji element dođe na pretposljednje mjesto. Treća vrsta se dobija tako što cirkularno rotiramo elemente druge vrste za jedno mjesto ulijevo, itd. Znači, matrica **C** bi trebala izgledati:

$$C = \begin{pmatrix} a_0 & a_1 & \dots & a_{N-1} \\ a_1 & a_2 & \dots & a_0 \\ \dots & \dots & \dots & \dots \\ a_{N-1} & a_0 & \dots & a_{N-2} \end{pmatrix}$$

17. Napisati program koji za zadati prirodan broj **M** ispisuje sve proste brojeve koji su manji ili jednaki **M**, koristeći algoritam **Eratostenovo sito**. Algoritam je sljedeći: Formiramo niz brojeva od **2** do **M**. Polazimo od početka niza, tj. uzimamo element 2 i prolazimo kroz ostatak niza "precrtavajući" sve elemente do kraja niza koji su djeljivi sa 2. Kao "precrtavanje" se može uzeti postavljanje vrijednosti na 0. Kada smo to odradili uzimamo sljedeći neprecrtani element, tj. 3 i ponavljamo postupak, tj. precrtavamo sve elemente koji su veći od 3 i manji ili jednaki **M** i koji su djeljivi sa 3. Postupak se nastavlja za sledeći neprecrtani element i završava kada nema neprecrtanih elemenata u nizu koji su veći od tekućeg elementa. Brojevi koji su ostali neprecrtani u početnom nizu su prosti brojevi manji ili jednaki **M**.