

1. Napisati funkciju **SUSJEDNI** koja za argument ima niz cijelih brojeva **X**, dužine **N**, pri čemu je niz je sortiran u opadajući poredak (ne provjeravati). Funkcija treba da pronade i odštampa dva susjedna elementa niza koja su najbliža po vrijednosti (najmanja razlika između njih). Ako postoji više parova susjednih elemenata sa najmanjom razlikom, odštampati par po želji. Napisati i glavni dio algoritma koji poziva kreiranu funkciju i štampa njen rezultat.

Primjer: Za niz $X=[16\ 12\ 10\ 5\ 2]$, funkcija treba da odštampa brojeve 12 i 10.

2. Napisati funkciju **CIFRE** koja za argument ima string **S** i niz **X** od **N** cijelih brojeva. Funkcija treba da provjeri da li svaka cifra stringa **S** ima odgovarajući broj u nizu **X**, i u tom slučaju funkcija vraća 1. U suprotnom, vraća 0. Napisati i glavni dio algoritma koji poziva kreiranu funkciju i štampa njen rezultat.
3. Napisati program koji učitava niz cijelih brojeva **X**, dužine **N**. U datom nizu potrebno je pronaći i odštampati element za koji važi da je suma svih elemenata sa njegove lijeve strane jednaka sumi svih elemenata sa njegove desne strane. Ukoliko takav element ne postoji potrebno je odštampati obavještenje "Ne postoji".

Primjer: Za niz $X=[1\ 12\ 5\ 6\ 4\ 3]$ treba da bude odštampan broj 5, dok za niz $X=[-1\ 7\ -3\ -5\ 9\ -2]$ treba da bude odštampan broj 9.

4. Napisati funkciju **JEDINICE** koja za argument ima niz cjelobrojnih vrijednosti **A**. Funkcija treba da zamijeni element niza sa brojem desetica tog elementa. Ukoliko je element jednocifren upisati -1. Napisati i glavni program koji učitava niz **A**, poziva napisanu funkciju i štampa rezultat.
5. Napisati program koji učitava niz cijelih brojeva **X**, dužine **N** i cio broj **T**. Pretpostaviti da je niz **X** sortiran u rastući poredak (ne treba provjeravati). Program treba da pronade da li u nizu **X** postoje dva elementa čiji je količnik jednak broju **T**. Ukoliko postoje, odštampati te brojeve. Ukoliko postoji više parova takvih brojeva odštampati jedan par po izboru. Ukoliko ne postoje takva dva broja, odštampati odgovarajuću poruku.

Primjer: Za niz $X=[3\ 7\ 12\ 15\ 28]$ i $T=4$ program treba da odštampa brojeve 3 i 12 ili 7 i 28. Za niz $X=[3\ 4\ 13]$ i $T=4$, program treba da odštampa poruku da nema takih brojeva.

6. Kreirati strukturu **NEPAR**, koja kao podatke članove ima niz cijelih brojeva **X** i njegovu dužinu **N**. Kreirati funkciju koja za ulazni argument ima dvije strukture tipa **NEPAR** i koja kao rezultat vraća broj 1 ako prva struktura ima sve elemente niza kao neparne ili 2 ako druga struktura ima sve elemente neparne. Ukoliko nijedna od struktura ne zadovoljava postavljene uslove vratiti -1.
7. Elementi liste su strukture koje sadrže jedan string. Kreirati rekurzivnu funkciju koja kao argument ima glavu liste i određuje da li su stringovi u listi leksikografski uređeni u rastući poredak.
8. Kreirati strukturu **VELIKO**, koja za podatke ima dva cijela broja **X** i **Y** i string **B**. Napisati funkciju **poredi** koja za ulazni argument ima dvije strukture tipa **VELIKO** i koja kao rezultat vraća prvo veliko slovo stringa one strukture koja ima veću vrijednost podatka **X**.

9. Napisati program kojim se učitava niz studenata definisanih strukturom **STUDENT**, koja sadrži ime studenta i niz njegovih ocjena. Potrebno je realizovati funkciju **SREDNJA_OCJENA**, kojoj se kao argument proslijeđuje formiran niz struktura tipa **STUDENT**, a kao rezultat je potrebno odrediti i odštampati pojedinačne prosječne ocjene studenata i njihovu ukupnu prosječnu ocjenu.
10. * Kreirati strukturu **KOLOKVIJUM**, koja za podatke ima string **SALA**, niz cijelih brojeva **BROJ_INDEKSA** koji predstavlja broj indeksa studenata koji su se prijavili da rade kolokvijuma, niz cijeli brojeva **GODINA_UPISA** koji predstavlja godine upisa studenata (l-ti element niza **BROJ_INDEKSA** i l-ti element niza **GODINA_UPISA** određuju broj indeksa i godinu upisa l-tog studenta), dužinu pomenutih nizova **BROJ_STUDENATA**. Napisati funkciju koja za ulazni argument ima jednu strukturu tipa **KOLOKVIJUM**. Funkcija treba da odštampa sve brojeve indeksa i godine upisa studenata sa najvećom godinom upisa.
11. Kreirati klasu **AUTOMOBIL** koja za podatke ima marku (string), starost u godinama (cio broj), broj pređenih kilometara (cio broj) i potrošnju goriva na 100 km (realan broj). Klasa treba da ima bar po dva konstruktora, mutatora i inspektora. Realizovati ove funkcije članice, kao i funkciju članicu **NAJVECI** koja kao argument ima niz automobila a određuje i štampa najvećeg od njih. Od dva automobila, veći je onaj koji je stariji. Ako su jednako stari, veći je onaj koji ima veći broj pređenih kilometara. Ako je i kilometražna ista, veći je onaj koji ima veću potrošnju.
12. * Kreirati klasu **PREGLED** koja ima podatke članove **NAZIV** (string – naziv pacijenta), **BROJ_UZORAKA** (cio broj koje predstavlja broj uzetih uzoraka u izvještaju), **MJERENJA** (niz realnih brojeva gdje l-ti element vrijednost l-tog uzorka). Klasa treba da ima bar po dva konstruktora, mutatora i inspektora. Realizovati ove funkcije članice, kao i funkciju članicu **MIN** koja za argument ima cio broj K. Ukoliko je K=1 funkcija računa minimalno mjerenje. Ukoliko je K=2, funkcija računa drugo najmanje mjerenje, itd...
13. Kreirati klasu **RADNIK** koja ima podatke članove **KOEFICIJENT** (cijeli broj – koeficijent za platu), **BROJ** (cijeli broj – identifikacioni broj radnika), **IME** (string – ime radnika). Realizovati funkciju članicu koja od dva radnika vraća ime radnika sa većim koeficijentom za platu.
14. Kreirati klasu **TACKA** koja ima podatke članove **X, Y** (realni brojevi – koordinate tačke) i funkciju članicu za računanje rastojanja između dvije tačke. Nakon toga kreirati klasu **KRUG** koja kao podatak član sadrži **CENTAR** (**TACKA** – predstavlja centar kruga) i funkcije članice za izračunavanje obima i površine kruga.