

Programski jezici

<http://www.programskijezici.matf.bg.ac.rs/>

**Univerzitet u Beogradu
Matematički fakultet**

Dizajn programskih jezika

Materijali za vežbe

**Nastavnik: Milena Vujošević Janičić
Asistent: Marjana Gligorjević**

**Beograd
2021.**

Priprema materijala:

dr Milena Vujošević Jančić, docent na Matematičkom fakultetu u Beogradu

Marjana Gligorijević, asistent na Matematičkom fakultetu u Beogradu

Branislava Živković

Nemanja Mićović, asistent na Matematičkom fakultetu u Beogradu

Milica Selaković, asistent na Matematičkom fakultetu u Beogradu

Milan Čugurović, asistent na Matematičkom fakultetu u Beogradu

Ivan Ristović, asistent na Matematičkom fakultetu u Beogradu

Sadržaj

1 Skript programiranje	3
1.0.1 Različiti formati podataka. Datum.	3
1.0.2 Argumenti komandne linije. Sortiranje. Obilazak direktorijuma.	4
1.0.3 Zadaci za vežbu	4
1.1 Tabelačno predstavljeni podaci i njihova vizualizacija	6
1.1.1 Matplotlib i NumPy	6
1.1.2 Pandas	8
2 Komponentno programiranje	9
2.1 PyQt5	9
2.1.1 Uvod	9
2.1.2 Zadaci za samostalni rad	9
3 Konkurentno programiranje	11
3.1 Go	11
3.1.1 Uvod u programski jezik Go	11
3.1.2 Gorutine	11
3.1.3 Uslovni redovi čekanja	13
4 Generičko programiranje	15
4.1 Osnove programskog jezika C++	15
4.1.1 Uvodni primeri	15
4.1.2 Šabloni funkcija. Opcioni argumenti šablona funkcija.	15
4.1.3 Zadaci za vežbu	16
4.2 Šabloni klasa	17
4.2.1 Uvodni primeri	17
4.2.2 Zadaci za vežbu	17
4.3 STL. Funkcionalni. Lambda funkcije.	17
4.3.1 Uvodni primeri	17
4.3.2 Razni zadaci sa rešenjima	18
4.3.3 Zadaci za vežbu	20

1

Skript programiranje

Potrebno je imati instaliran Python 3.7 na računaru. Uputstvo za instaliranje na različitim operativnim sistemima možete pogledati na <https://realpython.com/installing-python/>.

Literatura:

- (a) <https://www.python.org/>
- (b) <http://www.tutorialspoint.com/python>
- (c) <https://wiki.python.org/moin/>

1.0.1 Različiti formati podataka. Datum.

JSON

JSON (<https://www.json.org/>) reprezentacija objekata predstavlja jednostavni i pregledan način za serijalizaciju objekata. Koristi se svuda, pre svega u web programiranju kada je potrebno proslediti objekte preko mreže ali i lokalno kada je na primer potrebno sačuvati neki objekat u bazi podataka. JSON je nezamenjiv kod takozvanih *RESTful servisa* (servisa zasnovanih na REST-u, videti https://en.wikipedia.org/wiki/Representational_state_transfer). Korisnici obično pošalju zahtev preko HTTP protokola, a servis odgovara JSON reprezentacijom objekta koji ima informacije koje je korisnik tražio.

Zadatak 1.1 Korisnik na standardni ulaz unosi podatke o imenu, prezimenu i godinama. Program potom kreira JSON objekat *junak*, koji ima podatke *Ime*, *Prezime* i *Godine*, i ispisuje ga na standardni izlaz, a potom i u datoteku *datoteka.txt*.

Zadatak 1.2 Napisati program koji iz datoteke *datoteka.txt* učitava JSON objekat, a potom na standardni izlaz ispisuje podatke o *imenu*, *prezimenu* i *godinama*.

CSV

CSV (https://en.wikipedia.org/wiki/Comma-separated_values) datoteka predstavlja specijalan tip datoteke u kojoj se skladište informacije razdvojene zarezima ili nekim drugim specijalnim karakterom kao delimiterom podataka. Ovaj format datoteka se često koristi za razmenu podataka između različitih aplikacija. Na primer, baze podataka i kontakt menadžeri često podržavaju CSV datoteke.

Zadatak 1.3 Napisati program koji učitava csv datoteku *automobili.csv* sa podacima o različitim modelima i na standardni izlaz ispisuje nazive kolona i podatke o prvih 5 automobila. Dodatno, učitati datoteku u rečnik i prikazati podatke o prvih 5 automobila.

Zadatak 1.4 Formatirati ispis podataka datoteke iz prethodnog zadatka, tako da se sadržaj kolona prikazuje na 10 pozicija sa desnim poravnanjem. Dodatno, realne brojeve iz poslednje dve kolone ispisati na dve decimale. Više o formatiranju možete pogledati na <https://pyformat.info/>.

Zadatak 1.5 Napisati program koji kreira csv datoteku *zaposleni.csv* sa podacima o zaposlenima u firmi. Datoteka treba da sadrži ime, prezime i naziv odseka u kom radi zaposleni redom po kolonama. Dodati odgovarajuće zaglavlje datoteke.

Zadatak 1.6 Napisati program koji kreira csv datoteku *zaposleni.csv* sa podacima o zaposlenima u firmi na osnovu zadatih rečnika za svakog zaposlenog. Pretpostaviti da rečnici imaju odgovarajuće ključeve za ime, prezime i naziv odseka u kom radi zaposleni. Dodati odgovarajuće zaglavlje datoteke.

Zadatak 1.7 Napisati program koji ispisuje tekući dan u nedelji, dan, mesec i vreme u formatu *hh:mm:ss*.

1.0.2 Argumenti komandne linije. Sortiranje. Obilazak direktorijuma.

Zadatak 1.8 Napisati program koji na standardni izlaz ispisuje argumente komandne linije.

Zadatak 1.9 Napisati program koji na standardni izlaz ispisuje oznaku za tekući i roditeljski direktorijum, kao i separator koji se koristi za pravljenje putanje.

Zadatak 1.10 Napisati program koji imitira rad komande *ls*. Program na standardni izlaz ispisuje sadržaj tekućeg direktorijuma.

Zadatak 1.11 Napisati program koji na standardni izlaz ispisuje sve apsolutne putanje regularnih fajlova koji se nalaze u tekućem direktorijumu.

Zadatak 1.12 U datoteci *tacke.json* se nalaze podaci o tačkama u sledećem formatu.

Listing 1.1: *tacke.json*

```
1 [ {"teme": "A" , "coordinate": [10.0, 1.1]},
2   {"teme": "B" , "coordinate": [1.0, 15.0]},
3   {"teme": "C" , "coordinate": [-1.0, 5.0]} ]
```

Napisati program koji učitava podatke o tačkama iz datoteke *tacke.json* i sortira i po udaljenosti od koordinatnog početka. Na standardni izlaz ispisati podatke pre i posle sortiranja.

1.0.3 Zadaci za vežbu

Zadatak 1.13 Napisati program koji iz datoteke *ispiti.json* učitava podatke o ispitima i njihovim datumima. Ispisati na standardni izlaz za svaki ispit njegovo ime i status "Prosao" ukoliko je ispit prosao, odnosno "Ostalo je jos n dana.", gde je n broj dana od trenutnog datuma do datuma ispita.

Primer 1

```
POKRETANJE: python ispiti.py
INTERAKCIJA SA PROGRAMOM:
Relacione baze podataka Prosao
Vestacka inteligencija Prosao
Linearna algebra i analiticka geometrija Prosao
```

Sadržaj ulaznih datoteka koje se koriste u prethodnom primeru upotrebe programa:

Listing 1.2: *ispiti.json*

```
1 [ {'ime': 'Relacione baze podataka',
2   'datum': '21.09.2016.'},
3   {'ime': 'Vestacka inteligencija',
4   'datum': '17.06.2017.'},
5   {'ime': 'Linearna algebra i analiticka geometrija',
6   'datum': '08.02.2017.'} ]
```


Zadatak 1.14 Napisati program koji računa odnos kardinalnosti skupova duže i šire za zadati direktorijum. Datoteka pripada skupu duže ukoliko ima više redova od maksimalnog broja karaktera po redu, u suprotnom pripada skupu šire. Sa standardnog ulaza se unosi putanja do direktorijuma. Potrebno je obići sve datoteke u zadatom direktorijumu i njegovim poddirektorijumima (koristiti funkciju `os.walk()`) i ispisati odnos kardinalnosti skupova duže i šire.

Primer 1

```
POKRETANJE: python duze_sire.py
INTERAKCIJA SA PROGRAMOM:
  Unesite putanju do direktorijuma:
  ..
  Kardinalnost skupa duze : Kardinalnost skupa sire
  10 : 15
```

Zadatak 1.15 Napisati program koji obilazi direktorijume rekurzivno i računa broj datoteka za sve postojeće ekstenzije u tim direktorijumima. Sa standardnog ulaza se unosi putanja do početnog direktorijuma, a rezultat se ispisuje u datoteku `rezultat.json`.

Primer 1

```
POKRETANJE: python ekstenzije.py
INTERAKCIJA SA PROGRAMOM:
  Unesite putanju do direktorijuma:
  .
```

Sadržaj datoteke koja se generise u prethodnom primeru upotrebe programa:

Listing 1.3: `rezultat.txt`

```
1 {
2   'txt' : 14,
3   'py'  : 12,
4   'c'   : 10
5 }
```

Zadatak 1.16 U datoteci `radnici.json` nalaze se podaci o radnom vremenu zaposlenih u preduzeću u sledecem formatu:

```
1 [ { 'ime' : ime\_radnika, 'odmor' : [pocetak, kraj], 'radno_vreme'
   : [pocetak, kraj] }, ... ]
```

Napisati program koji u zavisnosti od unete opcije poslodavcu ispisuje trenutno dostupne radnike odnosno radnike koji su na odmoru. Moguće opcije su 'd' - trenutno dostupni radnici i 'o' - radnici koji su na odmoru. Radnik je dostupan ukoliko nije na odmoru i trenutno vreme je u okviru njegovog radnog vremena.

Primer 1

```
POKRETANJE: python odmor.py
INTERAKCIJA SA PROGRAMOM:
  "Unesite opciju koju zelite
  d - dostupni radnici
  o - radnici na odmoru :
  m
  Uneta opcija nije podrzana
```

Primer 2

```
POKRETANJE: python odmor.py
INTERAKCIJA SA PROGRAMOM:
  "Unesite opciju koju zelite
  d - dostupni radnici
  o - radnici na odmoru :
  d
  Pera Peric
```

Sadržaj ulaznih datoteka koje se koriste u prethodnom primeru upotrebe programa:

Listing 1.4: `radnici.json`

```
1 [ { 'ime' : 'Pera Peric',
2   'odmor' : ['21.08.2016.', '31.08.2016.'],
3   'radno_vreme' : ['08:30', '15:30'] } ]
```

Zadatak 1.17 Napisati program koji učitava ime datoteke sa standardnog ulaza i na standardni izlaz ispisuje putanje do svih direktorijuma u kojima se nalazi ta datoteka.

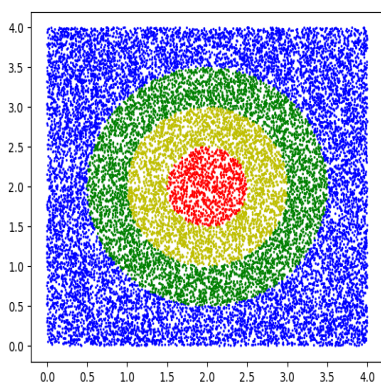
Primer 1

```
POKRETANJE: python pojavljivanja.py
INTERAKCIJA SA PROGRAMOM:
Unesite ime datoteke:
1.py
/home/student/vezbe/cas1/1.py
/home/student/vezbe/cas7/1.py
/home/student/vezbe/cas9/1.py
```

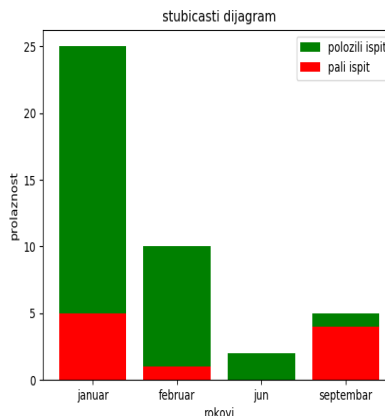
1.1 Tabelarno predstavljeni podaci i njihova vizualizacija

1.1.1 Matplotlib i NumPy

Matplotlib <https://matplotlib.org/> je najpopularnija Python biblioteka za vizuelizaciju podataka. NumPy je biblioteka sa velikim brojem funkcija za rad u domenu linearne algebre, Furijeovih transformacija i matrica. Osnovni objekti sa kojima radi ova numerička biblioteka su specijalna vrsta nizova koji obezbeđuju i do 50 puta bržu obradu od klasičnih Python listi. Uputstvo za instaliranje na različitim operativnim sistemima možete pogledati na sledećem [linku](#).



(a) primer tačkastog dijagrama



(b) primer stubičastog dijagrama

Slika 1.1: (a) slika za 19. zadatak, (b) slika za 20. zadatak

Zadatak 1.18 Napisati program kojim se iscrtavaju grafici sinusne i kosinusne funkcije na intervalu $[0, 2\pi]$ sa korakom 0.05. Sinusna funkcija treba da bude iscrtana zelenom, a kosinusna crvenom bojom. Debljinu linije postaviti na 3. Dodati odgovarajuće labela i legendu.

Zadatak 1.19 Napisati program kojim se nasumično generiše 20000 tačaka sa vrednostima koordinata iz intervala $[0, 4)$. Prikazati generisane tačke na tačkastom dijagramu pri čemu se boja tačke određuje na osnovu njene udaljenosti od tačke (2,2). Ako je tražena udaljenost ispod 0.5 tačku prikazati crvenom bojom, ispod 1 žutom, ispod 1.5 zelenom, inače iskoristiti plavu boju. Primer traženog dijagrama se može videti na slici 1.1(a).

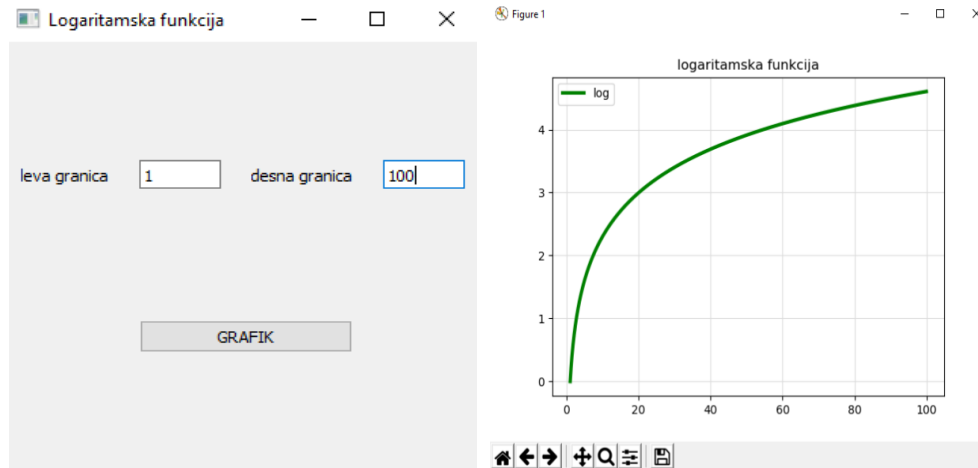
Zadatak 1.20 U zasebnim listama su dati podaci o broju studenata koji su položili, odnosno pali ispit u jednom od četiri ispitna roka za zimski semestar. Kreirati stubičasti dijagram koji prikazuje prolaznost po rokovima. Primer traženog dijagrama se može videti na slici 1.1(b).

Zadatak 1.21 Na osnovu rezultata jedne ankete, utvrđeno je da je 45% ispitanika glasalo za, 20% protiv, a 35% je bilo uzdržano. Napisati program kojim se iscrtava odgovarajući kružni grafik. Svaku vrednost obojiti različitom bojom, i dodati odgovarajuće labela.

1.1 Tabelarno predstavljeni podaci i njihova vizualizacija

Za pravljenje grafičkog interfejsa u narednim zadacima pogledati uvodne primere iz poglavlja "Komponentno programiranje".

Zadatak 1.22 Napisati program kojim se iscrtava grafik logaritamske funkcije na intervalu koji zadaje korisnik. Funkcija treba da bude iscrtana zelenom linijom debljine 3. Napraviti odgovarajući grafički interfejs na kom postoje dva polja za unos leve i desne granice intervala sa odgovarajućim labelama i jedno dugme (pogledati sliku (a)). Klikom na dugme započinje iscrtavanje grafika i prikaz istog (pogledati sliku (b)). Pretpostaviti da će zadati podaci biti ispravni.

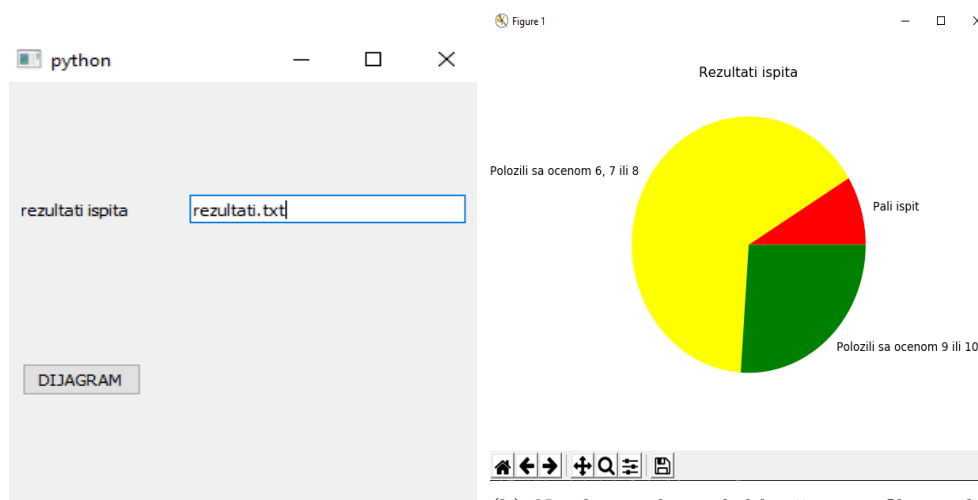


(a) Na slici je dat izgled aplikacije.

(b) Na slici je dat izgled grafika za interval $[1, 100]$.

Slika 1.2: Slike za 22. zadatak

Zadatak 1.23 Napisati program kojim se iscrtava kružni grafik na osnovu rezultata ispita koji se učitavaju iz datoteke čiji naziv zadaje korisnik. U datoteci je zadat niz sa šest brojeva koji predstavljaju redom broj studenata koji je pao ispit, odnosno položio isti sa ocenom od 6 do 10. Napraviti odgovarajući grafički interfejs na kom postoji polje za unos naziva datoteke i jedno dugme (pogledati sliku (a)). Klikom na dugme započinje čitanje podataka iz datoteke i iscrtavanje grafika i prikaz istog. Na grafiku prikazati tri grupe informacija (pali, položili sa ocenom od 6 do 8, položili sa ocenom 9 ili 10) određenom bojom (pogledati sliku (b)). Pretpostaviti da će zadati podaci biti ispravni.



(a) Na slici je dat izgled aplikacije.

(b) Na slici je dat izgled kružnog grafika za datoteku rezultati.txt sa nizom brojeva: 2 8 4 3 2 4

Slika 1.3: Slike za 23. zadatak

1.1.2 Pandas

Pandas ([sajt](#)) predstavlja brzu i fleksibilnu open source biblioteku za Python programski jezik koja pruža visoke performanse za analizu podataka. Neke od osnovnih funkcionalnosti koje se mogu naći u ovoj biblioteci su: rad sa podacima u različitim formatima (csv, txt, sql,...), statistika, indeksiranje, sortiranje, rangiranje, filtriranje, grupisanje (GroupBy), vizualizacija podataka. Uputstvo za instaliranje na različitim operativnim sistemima možete pogledati na sledećem [linku](#).

Zadatak 1.24 Napraviti seriju od prvih pet prirodnih brojeva sa imenom `brojevi`, a nakon toga seriju transformisati unarnom funkcijom $2*x+1$. Prikazati sve informacije o napravljenj seriji.

Zadatak 1.25 Napraviti tabelu od dve serije podataka čiji su elementi prvih deset prirodnih brojeva i vrednosti njihovih kvadrata redom. Nakon toga, dodati novu kolonu sa kubnim vrednostima elemenata prve kolone. Izvući iz tabele sve vrste u kojima su kubne vrednosti parne, a kvadrati nisu deljivi sa 3. Rezultat filtriranja sačuvati u tabeli `rezultat.csv`.

Zadatak 1.26 U fajlu `automobili.csv` su dati podaci o prosečnoj potrošnji različitih modela automobila (model, potrošnja po gradu i autoputu redom po kolonama). Učitati tabelu iz fajla i prikazati sumarne informacije o tabeli. Dodati novu kolonu sa prosečnom potrošnjom za svaki model i po njoj sortirati tabelu opadajuće i nakon toga prikazati pet modela sa najvećom potrošnjom goriva. Dodatno, izvući iz tabele vrste u kojima je prosečna potrošnja između 5 i 7 litara i rezultat filtriranja sačuvati u tabeli `ekonomicni.csv`.

Zadatak 1.27 U fajlu `podaci.json` su dati podaci o jednoj grupi dece u obliku liste listi. Za svako dete su zasebnoj listi dati ime, pol, starost (u godinama), masa (u kilogramima) i visina (u centimetrima) redom. Učitati podatke iz fajla u tabelu i dodati odgovarajuće nazive kolonama. Odrediti i ispisati na standardni izlaz:

- a) visinu najnižeg deteta
- b) koliko ima godina najstarije dete
- c) prosečnu visinu dece
- d) da li cela grupa može da stane u lift čija je nosivost 600 kg
- e) prikazati stubičastim dijagramom visinu dece u grupi

Zadatak 1.28 U fajlu `tacke.csv` su dati podaci o tačkama: naziv, dve koordinate i boja redom. Učitati tabelu iz fajla i prikazati naredne informacije:

- a) za koliko tačaka su učitani podaci
- b) koliko tačaka ima prvu koordinatu manju od 30 i boju koja nije NaN (nepoznata vrednost)
- c) izdvojiti one tačke koje su crvene ili čije su obe koordinate manje od 10
- d) koliko ima tačaka obojenih određenom bojom i dati grafički prikaz koristeći kružni grafik uz koji treba ispisati nazive boja
- e) prikazati grafički prvih 15 tačaka na osnovu njihovih koordinata
- f) tačke koje su crvene ili plave sačuvati u poseban csv fajl

2

Komponentno programiranje

2.1 PyQt5

PyQt5 je biblioteka koja nam omogućava da koristimo Qt Gui framework za pravljenje aplikacija iz Python-a. Uputstvo za instaliranje na različitim operativnim sistemima možete pogledati na sledećem [linku](#). Pregled osnovnih modula i funkcionalnosti ove biblioteke možete pogledati na sledećem [linku](#).

2.1.1 Uvod

Zadatak 2.1 Napisati program koji korisniku omogućava da vrši konverziju iz dinara u evre i obratno. Za vrednost 1 evra uzeti 117.5 dinara, a ukoliko dođe do greške, obavestiti o tome korisnika. Napraviti odgovarajući grafički korisnički interfejs, tako da postoji polje za unos vrednosti koja se konvertuje i za svaku vrstu konverzije odgovarajuće dugme.

Zadatak 2.2 Napisati program koji iscrtava tri kvadrata dužine stranice 100 i boji ih nasumično odabranim bojama u RGB formatu. Validne RGB vrednosti su celi brojevi iz intervala [0,255]. Kvadrata organizovati horizontalno sa razmakom od 10px.

Zadatak 2.3 Modifikovati program iz prethodnog zadatka dodavanjem dugmeta kojim se inicira iscrtavanje i bojenje kvadrata.

Zadatak 2.4 Napisati program koji za zadat alas nalog studenta pretražuje bazu sa podacima o studentima i prikazuje prosek studenta zaokružen na dve decimale. Podaci o studentima su zadati u fajlu `studenti.json` u obliku liste rečnika sa odgovarajućim ključevima. U slučaju da je zadat nepostojeći alas nalog, prikazati odgovarajuću poruku.

2.1.2 Zadaci za samostalni rad

Zadatak 2.5 Napisati program koji prikazuje matricu sa tablicom množenja dimenzije 9x9. Elementi matrice su dugmići na kojima je ispisana vrsta i kolona matrice kojoj pripadaju. Klikom na odgovarajući dugmić na labeli se prikazuje rezultat množenja vrste i kolone matrice. Upotrebiti odgovarajuću organizaciju komponenti.

Zadatak 2.6 Napisati program koji iscrtava krug, kvadrat ili pravougaonik u zavisnosti od dugmeta koje korisnik pritisne. Geometrijske elemente obojiti nasumično odabranom bojom u RGB formatu. Validne RGB vrednosti su celi brojevi iz intervala [0,255]. Dužine stranica, odnosno poluprečnika odabrati nasumično iz intervala [10,100].

Zadatak 2.7 Napisati program koji organizuje grupe za zajednički rad na seminarskom zadatku. Student zadaje svoj alas nalog i klikom na dugme dobija alas nalog kolege sa kojim radi seminarski kao i temu zadatka. Podaci o studentima koji rade seminarski su zadati u fajlu `studenti.txt` u obliku liste rečnika u kojima se čuva alas nalog i prosek studenta. Nazivi dostupnih tema za seminarski su zadati u fajlu `teme.txt` u zasebnim linijama. Tim se formira

nasumičnim odabirom studenta i teme. Student ne može samostalno da radi seminarski. Takođe, pamtiti do sada kreirane timove i izabrane teme tako da se pri ponovnom unosu istog studenta ne kreira novi tim već prikazu ranije generisane informacije. U slučaju da je zadat nepostojeći alas nalog, prikazati odgovarajuću poruku.

Zadatak 2.8 Napisati program koji simulira izvlačenje nagrada u jednom kolu. Napraviti proizvoljnu bazu nagrada (lista stringova) iz koje se nasumično bira sledeća za dodelu. Uppotrebite vertikalnu organizaciju komponenti. Prva dva elementa su polja za unos imena i prezimena učesnika nagradne igre. U sledećem redu postaviti dugme za generisanje nagrade. Pritiskom na dugme, nasumično izabrati nagradu (nakon čega je treba obrisati iz baze) i u labeli ispod dugmeta prikazati ime dobitnika i nagrade. U slučaju da više nema nagrada, prikazati poruku *Više sreće u sledecem krugu!*.

3

Konkurentno programiranje

3.1 Go

Go (<https://golang.org/>) je statički tipiziran jezik koji se kompilira, a razvija ga kompanija Google od 2007. godine. Go omogućava efikasnu konkurentnost koja je ugrađena u sam jezik, kao i automatsko upravljanje memorijom, odnosno sakupljanje smeća. Zbog efikasne konkurentnosti, Go je posebno dobar za izradu različitih vrsta serverskih aplikacija, ali je pre svega jezik opšte namene pa se može koristiti za rešavanje svih vrsta problema. Ima primenu u raznim oblastima kao što su grafika, mobilne aplikacije, mašinsko ucenje i još mnoge druge. Alat go predstavlja osnovni alat za upravljanje Go kodovima. Neke od definisanih komandi u okviru alata su: build za kompilaciju paketa, run za kompilaciju i izvršavanje Go programa, test za testiranje paketa, get za preuzimanje i instaliranje paketa, fmt za formatiranje koda paketa. Uputstvo za instaliranje i preuzimanje alata go za različite operativne sisteme možete pogledati na sledećem [linku](#).

3.1.1 Uvod u programski jezik Go

Osnovni pojmovi

Zadatak 3.1 Definisati novi tip podataka `Tacka` sa poljima koja predstavljaju celobrojne koordinatne tačke u ravni. Dodatno definisati funkciju `rastojanje` koja računa rastojanje dve tačke, kao i metod `rastojanje` nad strukturom `Tacka` koji računa rastojanje tačke od koordinatnog početka. Za dve tačke odrediti međusobno rastojanje i ispisati koja je bliža koordinatnom početku.

Zadatak 3.2 Definisati novi tip podataka `Krug` sa poljem koje predstavlja poluprečnik kruga i metod kojim se računa obim zadate strukture. Testirati metod nad 10 nasumično generisanih krugova sa celobrojnim poluprečnicima iz intervala $[1,10]$.

Zadatak 3.3 Definisati funkciju `suma` koja određuje sumu elemenata celobrojnog niza. Testirati funkciju nad nizom celih brojeva zapisanih u datoteci `brojevi.txt` u zasebnim redovima. U slučaju greške, ispisati odgovarajuću poruku i prekinuti program.

3.1.2 Gorutine

Zadatak 3.4 Definisati funkciju `suma` koja određuje sumu elemenata celobrojnog niza i rezultat obrade prosleđuje u kanal. Testirati rad funkcije nad nasumično generisanim nizom od 100 elemenata iz intervala $[0,100]$. Definisati dve gorutine koje pozivaju funkciju `suma` za prvu, odnosno drugu polovinu niza. Glavna gorutina treba da preuzme izračunate vrednosti iz kanala i da ispiše krajnji rezultat na izlaz.

Zadatak 3.5 Napisati program kojim se izračunava izraz $(2 + 36) * (15 + 100)$ upotrebom zasebnih gorutina za primenu matematičkih operacija koje se javljaju u izrazu. Glavna gorutina treba da ispiše vrednost izraza na standardni izlaz.

Zadatak 3.6 Napisati konkurentan program kojim se izračunava proizvod n -dimenzionalnog vektora skalarom. Korisnik unosi dimenziju vektora, elemente i skalar sa standardnog ulaza.

Koristiti zasebne gorutine za operisanje sa određenim elementima vektora. Broj gorutina treba da odgovara broju procesora koji su na raspolaganju. Ne praviti novi vektor za smeštanje rezultata. Glavna gorutina treba da ispiše rezultujući vektor na standardni izlaz. Zadatak rešiti:

- (a) sinhronizacijom niti preko katanca
- (b) bez sinhronizacije, podelom poslova po nitima unapred

Zadaci za samostalni rad sa rešenjima

Zadatak 3.7 Napisati konkurentan program kojim se izračunava skalarni proizvod dva n -dimenzionalna vektora. Korisnik unosi dimenziju vektora, a zatim i njihove elemente sa standardnog ulaza. Koristiti zasebne gorutine za operisanje sa određenim elementima vektora. Broj gorutina se zadaje kao argument komandne linije. Ne praviti novi vektor za smeštanje rezultata. Glavna gorutina treba da ispiše rezultujući skalar na standardni izlaz.

Zadatak 3.8 Napisati konkurentan program kojim se izračunava zbir elemenata dva niza dužine n ($[a_1, a_2, \dots, a_n] + [b_1, b_2, \dots, b_n] = [a_1 + b_1, a_2 + b_2, \dots, a_n + b_n]$). Korisnik unosi elemente sa standardnog ulaza. Za smeštanje rezultata koristiti prvi niz. Glavna gorutina treba da ispiše rezultujući niz na standardni izlaz, a za izračunavanje koristiti dodatne gorutine. Broj gorutina se zadaje kao argument komandne linije.

Zadatak 3.9 Napisati konkurentan program kojim se proverava koji su elementi niza prosti brojevi i ispisuje ih na standardni izlaz. U datoteci `brojevi.txt` je u prvoj liniji zadat broj elemenata, a zatim i elementi niza. Za proveru da li su odgovarajući elementi niza prosti brojevi koristiti zasebne gorutine. Broj gorutina treba da odgovara broju procesora koji su na raspolaganju.

Zadaci za vežbu

Zadatak 3.10 Rešiti zadatke iz prethodnog odeljka bez sinhronizacije, podelom poslova po nitima unapred. Broj niti treba da odgovara broju procesora koji su na raspolaganju.

Zadatak 3.11 Napisati konkurentan program kojim se izračunava proizvod dve kvadratne matrice reda n . Elementi matrice treba da budu nasumično generisani dvocifreni brojevi. Red matrice se zadaje kao argument komandne linije. Koristiti zasebne gorutine za operisanje sa određenim vrstama/kolonama matrice. Broj gorutina se zadaje kao argument komandne linije. Glavna gorutina treba da ispiše rezultujuću matricu.

Zadatak 3.12 Napisati konkurentan program koji proverava koji su elementi niza savršeni brojevi i ispisuje ih na standardni izlaz. Niz brojeva je zadat u datoteci čiji se naziv zadaje kao argument komandne linije. Za proveru da li su elementi niza savršeni brojevi koristiti zasebne gorutine. Broj gorutina treba da odgovara broju procesora koji su na raspolaganju. Savršenim se naziva prirodni broj kome je zbir pozitivnih delilaca različitih od sebe samog jednak tom broju ili, analogno, onaj broj n kojem je zbir delilaca $2n$. Primer savršenog broja: 28 ($1 + 2 + 4 + 7 + 14 = 28$).

Zadatak 3.13 Napisati konkurentan program kojim se određuje pozicija najmanjeg broja u zadatoj kvadratnoj matrici reda n . Elementi matrice treba da budu nasumično generisani brojevi manji od 100. Koristiti zasebne gorutine za operisanje sa određenim vrstama matrice. Red matrice i broj gorutina se zadaju kao argumenti komandne linije. Glavna gorutina treba da ispiše pronađenu poziciju.

3.1.3 Uslovni redovi čekanja

Zadatak 3.14 Banka Napisati program koji demonstrira rad banke i izvršavanje zahteva klijenata za transfer novca, tj. neku vrstu uplate. Kreirati banku sa 100 klijenata od kojih svaki mora da ima jedinstven identifikacioni broj računa i početni saldo. Za svakog klijenta kreirati zasebnu gorutinu koja šalje zahteve banci za tranfer određene količine novca sa računa tog klijenta na neki drugi račun. Banka obrađuje zahtev klijenta i ukoliko utvrdi da klijent na stanju ima dovoljnu količinu novca za traženi tranfer, obavlja ga i uslužuje sledećeg klijenta. Prikazivati na standardnom izlazu tok obrade zahteva klijenata i ukupno stanje u banci nakon svakog izvršenog transfera.

Zadaci za samostalni rad sa rešenjima

Zadatak 3.15 Čokoladni svet Postoji veliko skladište čokolade koje ima 100 hiljada čokoladica. Pored skladišta postoje 4 prodavnice u koje može da se smesti najviše 1000 čokoladica. Inicijalno svaka ima 100 čokoladica. Prodavnice se snabdevaju periodično iz skladišta na svakih 5 sekundi. One uvek uzmu toliko čokoladica da se dopune do maksimalnog kapaciteta. Za svaku prodavnicu kreirati zasebnu gorutinu koja obavlja dopunu njenih zaliha i upravlja kupcima. Postoji i 100 ljudi od kojih svako na 10 sekundi oseti potrebu za čokoladicom i odlazi do neke nasumično odabrane prodavnice. Potom u prodavnici, ako je dostupno, uzme nasumično od 1 do 7 čokoladica ili čeka u prodavnici dok se ne dopune zalihe. Za svaku osobu kreirati zasebnu gorutinu koja obavlja odlazak u prodavnicu i kupovinu određenog broja čokoladica. Modelovati sistem i prikazivati informacije kad god neko uđe u prodavnicu da kupi čokoladice i kad se obavi kupovina.

Zadatak 3.16 Železnička stanica Železnica nudi usluge prevoza robe. Vozovi kreću iz date stanice u nasumičnu sledeću samo ako su ili potpuno puni, ili potpuno prazni. Tada u slučaju da su potpuno puni, pokušavaju da istovare svu svoju robu u prvoj sledećoj stanici, inače pokušavaju da napune svoje vagone do punog kapaciteta. Postoji veza između svaka dva grada i pruga ima dva koloseka, odnosno, pruga nije deljeni resurs. Sa druge strane, stanice su deljeni resursi, iz razloga što u datom momentu više vozova može čekati na robu u njoj. Potrebno je kreirati četiri stanice i šest vozova koje inicijalno treba postaviti na različite polazne stanice. Vozovi će imati svoj kapacitet proizvoljno određen u opsegu od 5 do 10 tona. Takođe, inicijalno su svi prazni, dakle, moraće da uzmu robu iz polazne stanice. Za svaki voz kreirati zasebnu gorutinu koja obavlja prevoz robe. Stanice imaju proizvoljno određen kapacitet u opsegu od 10 do 15 tona i inicijalno su im skladišta puna. Modelovati sistem i prikazivati informacije kad god neki voz pristigne u stanicu i obavi preuzimanje ili predaju robe.

Zadaci za vežbu

Zadatak 3.17 Gradilište Lokalna građevinska firma gradi četiri petospratnice. Za svaku petospratnicu je obezbeđeno po 2 vozila, jedno za transport građevinskog materijala, a drugo za transport ostalih materijala. Prvo vozilo je kamion, a drugo veliki kombi. Razlika je u kapacitetu i brzini kretanja: kamion prenosi duplo više (10000 kg : 5000 kg), ali se kreće duplo sporije od kombija (postaviti da kamion pauzira vremenski duplo manje nego kombi). Građevinski materijal se nalazi u centralnom skladištu, dok se ostali materijali nalaze u dva dodatna manja skladišta. Da bi se rad na zgradi odvijao regularno, potrebno je imati dovoljne količine građevinskog i ostalog materijala. Kada jednog materijala ponestane (potrebno je bar 100 kg građevinskog i 50 kg ostalog materijala) izgradnja stoji, u suprotnom se svake sekunde utroši po 100 kg jednog i 50 kg drugog materijala. Na skladištima građevinskog i ostalog materijala uvek ima dovoljno da se napuni kamion ili kombi, dakle tu ne postoji nikakvo čekanje. Za izgradnju jednog sprata potrebno je 20 tona materijala (10 građevinskog i 10 ostalog). Kada se zgrade završe vozila obustavljaju rad. Na početku svi kamioni su na skladištu građevinskog materijala i kreću svaki ka petospratnici za koju su zaduženi. Kombiji su postavljeni na manjim skladištima. Vozila se, čim prebace materijal, vraćaju na skladište sa kog su pošli po nove zalihe. Kreirati zasebne gorutine za svako od vozila. Ispisivati koliko je trenutno upotrebljeno građevinskog, a koliko ostalog materijala, tj. kad je gradnja gotova, odnosno završena.

Zadatak 3.18 Diskoteka Na ulazu u diskoteku se nalazi bankomat sa neograničenom količinom novca. U diskoteci je 100 ljudi, i svako ima slučajan broj novčanica od 200 dinara, i to ne manje od 5, a ne više od 20. Budući da vreme brzo prolazi, svako troši po jednu novčanicu od 200 dinara u sekundi. Ako nema dovoljno novca, čovek ide do bankomata i podiže još 1000 dinara. Ako neko već podiže novac u tom trenutku, čeka ga da završi. Ako je podigao novac, vraća se u diskoteku da nastavi sa zabavom. Svakome treba oko 10 sekundi do bankomata i nazad. Kreirati zasebne gorutine za svakog čoveka u diskoteci. Ispisivati kretanje ljudi po diskoteci, da li se trenutno zabavljaju ili su krenuli do bankomata, odnosno, podigli novac sa istog.

4

Generičko programiranje

Potrebno je imati instaliran g++.

Literatura:

- (a) <http://www.generic-programming.org/>
- (b) <http://www.cplusplus.com/>

4.1 Osnove programskog jezika C++

4.1.1 Uvodni primeri

Zadatak 4.1 Napisati funkciju `void swap(int &a, int &b)` koja razmenjuje vrednosti dve celobrojne promenljive koristeći njihove reference. Učitati dva cela broja sa standardnog ulaza i testirati rad funkcije.

Zadatak 4.2 Definisati u zasebnom zaglavlju prostor imena `Segment` i u okviru njega funkciju `void ispis(int n)` koja, ukoliko je n neparan ispisuje neparne brojeve iz segmenta $[1, n]$, inače parne brojeve iz segmenta $[0, n]$. Definisati u zasebnom zaglavlju prostor imena `SegmentKvadrata` i u okviru njega funkciju `void ispis(int n)` koja ispisuje kvadrate svih celih brojeva iz segmenta $[0, n]$. Uključiti oba zaglavlja i za učitani pozitivan broj n testirati rad funkcija.

Zadatak 4.3 Definisati u zasebnom zaglavlju prostor imena `geometrija` i u okviru njega klasu `Trougao`. Obezbediti u klasi konstruktor sa tri realna argumenta, statički metod koji proverava ispravnost argumenata (da li zadate dužine zadovoljavaju nejednakost trougla), kao i metode za dobijanje dužina stranica trougla. Dodatno, definisati operatore za učitavanje i ispis objekta klase `Trougao`.

4.1.2 Šabloni funkcija. Opcioni argumenti šablona funkcija.

Zadatak 4.4 Napisati šablone funkcija za određivanje:

- (a) zbira dva objekta
- (b) n -tostruke vrednosti nekog objekta
- (c) minimalne vrednosti dva objekta

Primeniti prethodne šablone na:

- (A) cele brojeve
- (B) realne brojeve
- (C) objekte klase `Trougao`

Zadatak 4.5 Definirati u zasebnom zaglavlju prostor imena *vrsta* i u okviru njega klasu *Osoba*. Obezbediti u klasi konstruktor sa tri argumenta, imenom, prezimenom i godinom rođenja. Dodatno, definirati operator za poređenje osoba na osnovu godine rođenja i operatore za učitavanje i ispis objekta klase *Osoba*.

Zadatak 4.6 Napisati šablone funkcija za:

- (a) kreiranje niza tipa *T* dužine *n* date kao argument funkcije
- (b) učitavanje niza tipa *T* dužine *n* date kao argument funkcije
- (c) ispisivanje niza tipa *T* dužine *n* date kao argument funkcije
- (d) određivanje maksimuma niza tipa *T* dužine *n* sa podrazumevanom dužinom 2

Primeniti prethodne šablone na:

- (A) niz celih brojeva
- (B) niz realnih brojeve
- (C) niz objekata klase *Osoba*

4.1.3 Zadaci za vežbu

Zadatak 4.7 Napisati klasu *Datum* i u njoj obezbediti:

- (a) konstruktor sa podrazumevanim vrednostima (1.1.2017.)
- (b) pristupne metode
- (c) metod za izmenu datuma
- (d) metode za čitanje i ispisivanje datuma
- (e) aritmetičke operatore $+$, $-$ i operatore poređenja $<$, $==$, $>$

Zadatak 4.8 Napisati klasu *KompleksanBroj* i u njoj obezbediti:

- (a) konstruktor sa dva argumenta (realni i imaginarni deo) i podrazumevanim vrednostima (0,0)
- (b) pristupne metode
- (c) metod za izmenu kompleksnog broja
- (d) metode za čitanje i ispisivanje kompleksnog broja
- (e) aritmetičke operatore $+$, $-$, $*$
- (f) metod za određivanje modula
- (g) metod za određivanje argumenta

Zadatak 4.9 Napisati sledeće šablone funkcija:

- (a) Funkciju koja pronalazi maksimalni element u nizu.
- (b) Funkciju koja pronalazi minimalni element u nizu.
- (c) Funkciju koja pronalazi prosek niza.

Primeniti takve šablone funkcija nad nizovima:

- (A) Celobrojnih vrednosti
- (B) Realnih vrednosti
- (C) Trodimenzionih tačaka (kod tačaka udaljenost od (0,0,0) definiše magnitudu, a po toj udaljenosti se dalje računa i prosek)

4.2 Šabloni klasa

4.2.1 Uvodni primeri

Zadatak 4.10 Šablon klasa Koordinate za predstavljanje koordinata tačkaka u prostoru.

4.2.2 Zadaci za vežbu

Zadatak 4.11 Napraviti šablon od klase Trougao tako da dužine stranica mogu biti različitog tipa. Napisati program koji testira šablon klasu Trougao za stranice tipa int.

Zadatak 4.12 Napisati šablon klasu Radnik u okviru prostora imena kompanija. Radnika karakterišu jedinstveni identifikacioni broj id, tipa int i njegova plata (u dinarima), tipa T koji je parametar šablona. U klasi obezbediti:

- konstruktor sa dva argumenta za polja podatke
- operatore poređenja $<$, $==$ (poređenje radnika se svodi na poređenje njihovih plata, pretpostaviti da za objekte tipa T postoje definisani operatori poređenja)
- metode za računanje visine plate izražene u evrima i dolarima (smatrati da je kurs evra 119.2, a dolara 100.3), pretpostaviti da za objekte tipa T postoji definisano množenje sa konstantama i da će rezultat izračunavanja biti tipa T

Napisati program koji testira šablon klasu Radnik za platu tipa double.

Zadatak 4.13 Napisati šablon klasu Kvadar u okviru prostora imena geometrija koja kao polja podatke ima širinu, dužinu i visinu kvadra. Sva tri podatka su tipa T koji je parametar šablona. U klasi obezbediti:

- konstruktor sa tri argumenta za polja podatke
- metod za računanje zapremine kvadra (pretpostaviti da za objekte tipa T postoji definisana operacija množenja i da će rezultat izračunavanja biti tipa T)
- operatore poređenja $<$, $==$ (poređenje kvadara se svodi na poređenje njihovih zapremina, pretpostaviti da za objekte tipa T postoje definisani operatori poređenja)

Napisati program koji testira šablon klasu Kvadar za podatke tipa int.

4.3 STL. Funkcionalni. Lambda funkcije.

4.3.1 Uvodni primeri

Zadatak 4.14 Korisnik unosi cene artikala sve dok ne unese 0 (0 je oznaka za kraj ulaza, ne čuva se u kolekciji). Koristeći pogodnu kolekciju iz STL-a za čuvanje cena, napisati program koji:

- dodaje cene artikala u kolekciju u redosledu u kom stižu. Nakon učitavanja ispisati cenu svakog trećeg artikla počev od najnovijeg ka najstarijem.
- dodaje cene artikala u kolekciju u obrnutom redosledu od onog u kom stižu. Nakon učitavanja cena (čiji se unos završava sa 0), korisnik dodatno unosi još jednu cenu, nakon čega treba obrisati sve artikle skuplje od unete cene i ispisati modifikovanu kolekciju.
- dodaje cene artikala tako da neparne stavlja na početak, a parne na kraj kolekcije i tako do kraja ulaza. Ispisati dobijenu kolekciju i zbir unetih cena.
- eliminise duplikate cena iz unetog niza, a zatim ispisuje cene prvo u rastućem, a zatim i u opadajućem poretku.

Zadatak 4.15 Napisati program koji sa standardnog ulaza učitava pozitivan ceo broj n , a zatim podatke o n studenata u obliku *korisničko ime studenta sa alas-a prosek*. Nakon toga, korisnik unosi korisničko ime studenta čiji prosek želi da mu se prikaže. Ukoliko takav student ne postoji u bazi, omogućiti korisniku da doda podatke o tom studentu, a ukoliko postoji, prikazati trenutni prosek na standardni izlaz i omogućiti korisniku da promeni trenutni prosek za tog studenta. Koristiti pogodnu kolekciju iz STL-a za čuvanje podataka.

4.3.2 Razni zadaci sa rešenjima

Zadatak 4.16 Napisati program koji sa standardnog ulaza učitava pozitivan ceo broj n , a zatim dva vektora, gde je prvi vektor dužine n i proizvoljnog tipa, a drugi vektor dužine $n-1$ i sačinjen je od stringova čije su vrednosti $+$ ili $-$. Napisati šablon funkciju za učitavanje u vektor proizvoljnog tipa, kao i za ispis proizvoljnog vektora i šablon funkciju koja prihvata dva vektora pomenutog formata i primenjuje operaciju sabiranja ili oduzimanja nad prvim vektorom i ispisuje dobijeni rezultat. Na primer, ako su učitani vektori: $462 - 34$ i $++--$ dobiće se: $4 + 6 + 2 - (-3) + 4 = 19$. Testirati program nad vektorom celih i realnih brojeva. Pretpostaviti da su ulazni podaci u ispravnom formatu.

Zadatak 4.17 Napisati program koji sa standardnog ulaza učitava pozitivan ceo broj n , a zatim kolekciju dužine n proizvoljnog tipa, i menja je tako što nad svakim elementom kolekcije primenjuje unarnu transformaciju:

- (a) inkrementaciju.
- (b) dekrementaciju.
- (c) kvadriranje.

Napisati šablon funkciju za učitavanje i ispis proizvoljne kolekcije, kao i šablon funkciju koja menja kolekciju na pomenut način. Testirati program nad vektorom celih brojeva i listom realnih brojeva. Pretpostaviti da su ulazni podaci u ispravnom formatu.

Zadatak 4.18 Napisati program koji sa standardnog ulaza učitava pozitivan ceo broj n , a zatim kolekciju dužine n proizvoljnog tipa, i menja je tako što zadržava samo elemente koji ispunjavaju sledeći unarni uslov:

- (a) da su deljivi sa 3.
- (b) da nisu deljivi sa 5.
- (c) da su veći od 7.
- (d) da su manji od 11.

Napisati šablon funkciju za učitavanje i ispis proizvoljne kolekcije, kao i šablon funkciju koja menja kolekciju na pomenut način. Testirati uslove deljivosti/nedeljivosti nad vektorima celih brojeva, a poredbene uslove nad listama realnih brojeva. Pretpostaviti da su ulazni podaci u ispravnom formatu.

Zadatak 4.19 Uopštavanje prethodnog zadatka: Napisati program koji sa standardnog ulaza učitava pozitivan ceo broj n , a zatim kolekciju dužine n proizvoljnog tipa i menja kolekciju tako što zadržava samo elemente koji ispunjavaju unarni uslov:

- (a) da su deljivi sa dodatno učitanim celim brojem.
- (b) da su veći od dodatno učitano realnog broja.

Napisati šablon funkciju za učitavanje i ispis proizvoljne kolekcije, kao i šablon funkciju koja menja kolekciju na pomenut način. Testirati uslov deljivosti nad vektorom celih brojeva, a poredbeni uslov nad listom realnih brojeva. Pretpostaviti da su ulazni podaci u ispravnom formatu.

Zadatak 4.20 Napisati program koji sa standardnog ulaza učitava pozitivan paran ceo broj n , a zatim vektor celih brojeva dužine n .

- (a) Prvu polovinu vektora sortirati opadajuće korišćenjem odgovarajuće lambda funkcije koja definiše kriterijum poredjenja.
- (b) Drugu polovinu vektora sortirati opadajuće korišćenjem odgovarajućeg funkcijskog objekta koji daje informaciju da li je neki element veći od drugog elementa.

Ispisati vektor nakon sortiranja njegovih polovina. Pretpostaviti da su ulazni podaci u ispravnom formatu.

Zadatak 4.21 Napisati program koji sa standardnog ulaza učitava pozitivan ceo broj n , a zatim vektor celih brojeva dužine n i izračunava sumu elemenata vektora koji su parni. Zadatak rešiti:

- (a) korišćenjem unarne funkcije koja proverava da li element vektora paran.
- (b) korišćenjem funkcijskog objekta koji daje informaciju da li je element paran.
- (c) korišćenjem odgovarajuće lambda funkcije.

Napisati šablon funkciju koja izračunava sumu elemenata vektora koji zadovoljavaju zadati uslov parnosti prosleđen kao argument funkcije. Pretpostaviti da su ulazni podaci u ispravnom formatu.

Zadatak 4.22 Napisati program koji sa standardnog ulaza učitava pozitivan ceo broj n , a zatim kolekciju dužine n proizvoljnog tipa i ispisuje elemente nule linearne jednačine sa realnim koeficijentima koji se dodatno učitavaju. Ispisati i broj pronađenih nula u kolekciji. Napisati šablon funkciju za učitavanje i ispis proizvoljne kolekcije, kao i šablonski funkcional pod nazivom `Linear` koji pri kreiranju prihvata dva realna argumenta, odnosno koeficijente linearne jednačine. Testirati program nad vektorom celih i realnih brojeva. Pretpostaviti da su ulazni podaci u ispravnom formatu.

Zadatak 4.23 Napisati program koji sa standardnog ulaza učitava pozitivan ceo broj n , a zatim isto toliko imena osoba.

- (a) Odrediti koje ime je najčešće i koliko puta se pojavilo.
- (b) Sortirati kolekciju imena opadajuće po broju pojavljivanja.

Pretpostaviti da su ulazni podaci u ispravnom formatu.

Zadatak 4.24 Program učitava podatke o prodavnicu u sledećem formatu, broj vrsta artikala, a zatim za svaki artikal ime (pretpostaviti da je jedinstveno), količinu i cenu. Prikazati redovne cene i stanje u prodavnici za sve artikule, a zatim prikazati i cene artikala za vreme akcije - popusta od 10% na sve. Implementirati šablon funkciju `for_each` iz biblioteke `algorithm` za modifikovanje kolekcije sa podacima o prodavnici na traženi način.

Zadatak 4.25 Modifikovati prethodni program ako se akcijski popust (u procentima) unosi kao vrednost sa standardnog ulaza. Ukoliko je uneta vrednost veća od 50, podrazumevano primeniti akciju u pola cene. Napisati odgovarajući funkcional pod nazivom `Akcija` koji za argument prihvata vrednost popusta koji treba da se obračuna za sve artikule.

Zadatak 4.26 Program učitava podatke o prodavnicu u sledećem formatu, broj vrsta artikala, a zatim imena svih artikala, pa nakon toga raspoloživu količinu za sve artikule u odgovarajućem redosledu. Prikazati stanje u prodavnici za sve artikule. Zatim za one artikule kojih nema dovoljno na stanju (minimalno 100), postaviti količinu na 100 i prikazati novo stanje u prodavnici. Implementirati šablon funkciju `replace_if` iz biblioteke `algorithm` za modifikovanje kolekcije sa podacima o prodavnici na traženi način.

Zadatak 4.27 Modifikovati prethodni program ako se minimalna količina za artikule unosi kao vrednost sa standardnog ulaza. Ukoliko je uneta vrednost veća od 100, podrazumevano postaviti količinu na 100. Napisati odgovarajući funkcional pod nazivom `Dopuna` koji za argument prihvata vrednost koju treba postaviti kao novu količinu za artikule kojih nema dovoljno na stanju.

4.3.3 Zadaci za vežbu

Zadatak 4.28 Za vežbu implementirati preostale šablone definisane u STL biblioteci algorithm. <http://www.cplusplus.com/reference/algorithm/>