

PRAKTIKUM IZ PROGRAMIRANJA 1

- domaći zadatak broj 5 -

Sastaviti program na programskom jeziku Python, kojim se vrši određena vrsta obrade nad tekstualnim datotekama koje sadrže podatke o linijama gradskog prevoza u Beogradu. Glavni program treba da:

- učitava nazive ulazne i izlazne datoteka sa kojima će se raditi, kao i ostale podatke neophodne za realizaciju te obrade,
- poziva odgovarajuće funkcije za čitanje datoteke i obradu,
- ispisuje sve dobijene rezultate na način tražen postavkom zadatka;

Zavisno od rednog broja problema, svaki student treba da sastavi jedan od programa koji su dati u prilogu ovog dokumenta.

Podaci o jednoj liniji se nalaze u dve odvojene datoteke, za svaki smer posebno (`dirA` i `dirB`). Imena datoteka se sastoje od broja linije, donje crte (`_`) i oznake smera (`dirA` ili `dirB`). Podaci se u ulaznoj datoteci nalaze po sledećem formatu: broj stajališta (ceo broj), naziv stajališta (niz znakova), geografska širina (realan broj), geografska dužina (realan broj) i zona (ceo broj). Primer nekoliko linija ulazne datoteke `2_dirA.txt` sledi. Polja su odvojena znakom uzvika (!).

Ulazna datoteka:	2_dirA.txt
	7!Pristaniste!44.8189915!20.4495147!1
	85!Masinski fakultet!44.8077358!20.476938!1
	63!Vukov spomenik!44.8055219!20.4778908!1

Podaci o geografskim koordinatama (geografskoj širini i dužini) stajališta su dati u sfernom koordinatnom sistemu. Ukoliko je potrebno, da bi se pronašlo rastojanje d između dve geografske koordinate, potrebno je primeniti `haversine` formulu:

$$t1 = \sin^2\left(\frac{lat1 - lat2}{2}\right)$$

$$t2 = \sin^2\left(\frac{lon1 - lon2}{2}\right)$$

$$d = 2 * R * \text{asin}(\sqrt{t1 + t2 * \cos(lat1) * \cos(lat2)})$$

gde R predstavlja prečnik planete zemlje, $lat1$ i $lat2$ geografsku širinu prve, odnosno druge koordinate, a $lon1$ i $lon2$ geografsku dužinu prve, odnosno druge koordinate **u radijanima**. Smatrati da je prečnik planete Zemlje 6371 km. **Obratiti pažnju da su geografska širina (latitude) i geografska dužina (longitude) inicijalno zadate u stepenima, dok u haversine formuli treba da budu u radijanima.** Formula za pretvaranje stepeni u radijane je `radians = degrees * 3.14 / 180`.

Format ispisa je definisan postavkom zadatka i priloženim test primerima. Realne brojeve ispisivati na dve decimale. Datoteke se mogu otvoriti za pregled u bilo kom editoru teksta ili pomoću *Microsoft Excel* alata.

Voditi računa o dekompoziciji programa na potprograme prema gore navedenoj raspodeli i zahtevima pojedinačnih zadataka. Po potrebi, dozvoljeno je uvoditi i dodatne potprograme u rešenje. Student sam treba da definiše imena funkcija, kao i argumente potrebne za njihov rad. **Potprogrami ne smeju pristupati promenljivama glavnog programa direktno, već samo putem svojih argumenata i povratne vrednosti.** Nije dozvoljeno prekidanje rada programa pomoću sistemskih funkcija (npr. `exit()` ili `quit()`).

Prilikom rada programa proveravati i obraditi moguće greške korišćenjem mehanizma izuzetaka. U slučaju greške pri radu sa datotekom (nepostojanje datoteke), na standardnom izlazu ispisati poruku `DAT_GRESKA`. U slučaju nekorektnog formata podataka unetih sa standardnog ulaza (format je definisan postavkom zadatka), na standardnom izlazu ispisati

poruku **PODACI_GRESKA**. Zatim, korektno prekinuti izvršavanje programa. Proveru raditi samo za navedene podatke.

Ako nešto u postavci zadatka nije dovoljno precizno definisano ili ako su neki od zahteva međusobno suprotstavljeni, usvojiti razumnu pretpostavku i rešiti zadatak korišćenjem te pretpostavke. Po potrebi, analizirati dostupne primere ulaza i izlaza. Na samoj odbrani obavestiti demonstratora o usvojenoj pretpostavci ili pretpostavkama. Programski kod rešenja zadatka treba da bude uredno komentarisano, tako da pri pregledu programa lako može biti uočeno šta radi bilo koja programska celina.

Radi boljeg testiranja programa, odabrati nekoliko dodatnih skupova podataka sa kojima će program biti testiran. Svaki primer treba da sadrži ulazne podatke i očekivani izlaz za te podatke.

Napomene:

1. Rok za predaju petog domaćeg zadatka je **ponedeljak, 08.01.2024.** putem kursa predmeta na **Moodle** platformi za elektronsko učenje. Tačan termin za predaju će biti naknadno definisan za sve studente. Termin će biti ograničenog vremenskog trajanja.
2. Domaći zadaci će biti testirani i ocenjivani korišćenjem javnih i tajnih testova.
3. Studentima će nekoliko dana pre roka za predaju biti dostupno okruženje za testiranje rešenja domaćeg zadatka na Moodle platformi za elektronsko učenje korišćenjem javnih i tajnih testova.
4. Prilikom predaje domaćeg zadatka studenti će rešavati i kratak test znanja u vezi rešenja domaćeg zadatka i relevantnog gradiva iz oblasti programiranja na programskom jeziku Python koje obuhvata temu domaćeg zadatka.
5. Odbrana petog domaćeg zadatka će biti održana u **utorak, 09.01.2024.** Raspored studenata po termini biće objavljen nakon završetka predaje.
6. Domaći zadaci se rešavaju **samostalno**. Predmetni nastavnici zadržavaju pravo da nakon predaje domaćih zadataka izvrše proveru sličnosti i preuzmu odgovarajuće disciplinske mere.
7. Svi drugi detalji oko predaje, ocenjivanja i odbrane domaćeg zadatka će biti blagovremeno objavljeni.
8. Formula za redni broj problema **i** koji treba rešavati je sledeća (R – redni broj indeksa, G – poslednje dve cifre godine upisa):

$$i = (R + G) \bmod 4$$

9. Kao rešenje domaćeg zadatka potrebno je predati sadržaj sledeće datoteke:

- **dz5.py**, koja sadrži izvorni tekst programa na programskom jeziku Python;

29.12.2023. godine

sa predmeta

0. Napisati program koji pomaže putnicima u pronalaženju svih stajališta gradskog prevoza koja se nalaze u zadatom radijusu (poluprečniku) u odnosu na zadate koordinate putnika. Program putem standardnog ulaza prihvata ime izlazne datoteke, smer kretanja (`dirA` ili `dirB`), koordinate putnika (geografsku širinu i dužinu), radijus (u metrima) i spisak linija gradskog prevoza koje su od interesa za putnika. Korisnik mora navesti oznaku bar jedne linije gradskog prevoza. Program treba da odredi sva stajališta zadatih linija koja se nalaze u zadatom radijusu (u metrima) od koordinata putnika. Ukoliko više linija deli isto stajalište, ono u rezultatu treba da se pojavi samo jednom. Rastojanje između dve geografske tačke odrediti po datoj `haversine` formuli. Prilikom ispisa, datoteka treba da bude sortirana neopadajuće prema udaljenosti stajališta od putnika. Ukoliko su dva stajališta podjednako udaljena, sortirati ih leksikografski po nazivu. Format jednog reda izlazne datoteke je isti kao kod ulazne datoteke.

Program treba da:

- 1) Učita ime izlazne datoteke i smer kretanja sa standardnog ulaza.
- 2) Učita koordinate putnika i radijus sa standardnog ulaza. Ukoliko koordinate putnika ne predstavljaju valjan realan broj, prijaviti grešku o nekorektnim podacima.
- 3) Učita spisak linija sa standardnog ulaza koje su od interesa za putnika. Ukoliko podaci o liniji ne postoje, prijaviti grešku o nepostojećoj datoteci.
- 4) Izvrši zahtevanu obradu prema tekstu zadatka.
- 5) Formira izlaznu datoteku prema tekstu zadatka.
- 6) Vodi računa i obradi moguće izuzetke koji mogu nastati prilikom rada programa.

Primeri

Primer 1	
Standardni ulaz	<code>izlaz.txt dirA 44.8077350 20.476938 500 2 5 7 10</code>
Ulazna datoteka:	<code>2_dirA.txt 5_dirA.txt 7_dirA.txt 10_dirA.txt</code>
Izlazna datoteka:	<code>izlaz.txt</code>
Standardni izlaz:	
Primer 2	
Standardni ulaz	<code>izlaz.txt dirB 44.8055219 20.4778908 1000 12 27 32 25</code>
Ulazna datoteka:	<code>12_dirB.txt 27_dirB.txt 32_dirB.txt 25_dirB.txt</code>
Izlazna datoteka:	<code>izlaz.txt</code>
Standardni izlaz:	
Primer 3	
Standardni ulaz	<code>izlaz.txt dirA 44.8077350 20.476938 500 1</code>
Ulazna datoteka:	<code>1_dirA.txt #ne postoji</code>
Izlazna datoteka:	<code>-</code>
Standardni izlaz:	<code>DAT_GRESKA</code>
Primer 4	
Standardni ulaz	<code>izlaz.txt dirA 44.80f 20.476938 500 2 5 7 10</code>
Ulazna datoteka:	<code>-</code>
Izlazna datoteka:	<code>-</code>
Standardni izlaz:	<code>PODACI_GRESKA</code>

1. Napisati program koji određuje rastojanja među stajalištima i ukupnu dužinu za zadate linije javnog gradskog prevoza i smer kretanja. Program putem standardnog ulaza prihvata smer linije (`dirA` ili `dirB`) i spisak linija gradskog prevoza koje su od interesa za obradu. Korisnik mora navesti oznaku bar jedne linije gradskog prevoza. Program treba da izvrši zahtevanu obradu i formira zasebne izlazne datoteke sa podacima za svaku zadatu liniju. Ime izlazne datoteke formirati od imena linije, donje crte (`_`), smeru, donje crte (`_`) i sufiksa `distance.txt`. Rastojanje između dve geografske tačke odrediti po datoj `haversine` formuli. Format jednog reda izlazne datoteke je sledeći: `naziv stajališta 1!naziv stajališta 2!rastojanje`. Na kraj datoteke ispisati ukupnu dužinu linije u posebnom redu. Rastojanja zaokružiti na dve decimale. Stajališta i njihova rastojanja treba da budu u istom poretku kao što je ulazni.

Program treba da:

- 1) Učita imena ulazne i izlazne datoteke sa standardnog ulaza.
- 2) Učita smer kretanja sa standardnog ulaza. Ukoliko smer kretanja nije u dozvoljenom obliku, prijaviti grešku o nekorektnim podacima.
- 3) Učita spisak linija sa standardnog ulaza koje su od interesa za putnika. Ukoliko podaci o liniji ne postoje, prijaviti grešku o nepostojećoj datoteci.
- 4) Izvrši zahtevanu obradu prema tekstu zadatka.
- 5) Formira izlaznu datoteku prema tekstu zadatka.
- 6) Vodi računa i obradi moguće izuzetke koji mogu nastati prilikom rada programa.

Primeri

Primer 1	
Standardni ulaz	<code>dirA</code> <code>2 5 7 10</code>
Ulazna datoteka:	<code>2_dirA.txt 5_dirA.txt 7_dirA.txt 10_dirA.txt</code>
Izlazna datoteka:	<code>2_dirA_distance.txt 5_dirA_distance.txt</code> <code>7_dirA_distance.txt 10_dirA_distance.txt</code>
Standardni izlaz:	
Primer 2	
Standardni ulaz	<code>dirB</code> <code>27 32</code>
Ulazna datoteka:	<code>27_dirB.txt 32_dirB.txt</code>
Izlazna datoteka:	<code>27_dirB_distance.txt 32_dirB_distance.txt</code>
Standardni izlaz:	
Primer 3	
Standardni ulaz	<code>dirA</code> <code>1</code>
Ulazna datoteka:	<code>1_dirA.txt #ne postoji</code>
Izlazna datoteka:	<code>-</code>
Standardni izlaz:	<code>DAT_GRESKA</code>
Primer 4	
Standardni ulaz	<code>dirC</code> <code>27 32</code>
Ulazna datoteka:	<code>-</code>
Izlazna datoteka:	<code>-</code>
Standardni izlaz:	<code>PODACI_GRESKA</code>

2. Napisati program koja za zadate linije javnog gradskog prevoza određuje stajališta sa istim nazivom u oba smera, kao i njihovu međusobnu udaljenost. Program putem standardnog ulaza prihvata spisak linija gradskog prevoza koje su od interesa za obradu. Korisnik mora navesti oznaku bar jedne linije gradskog prevoza. Program treba da izvrši zahtevanu obradu i formira zasebne izlazne datoteke sa podacima za svaku zadatu liniju. Ime izlazne datoteke formirati od imena linije, donje crte (_) i sufiksa `station.txt`. Rastojanje između dve geografske tačke odrediti po datoj `haversine` formuli. Format jednog reda izlazne datoteke je sledeći: `naziv stajališta!udaljenost!zona`. Stajališta treba da budu uređena po rastućem leksikografskom poretku svojih naziva. Udaljenost zaokružiti na dve decimale.

Program treba da:

- 1) Učita spisak linija sa standardnog ulaza koje su od interesa za putnika. Ukoliko podaci o liniji ne postoje, prijaviti grešku o nepostojećoj datoteci.
- 2) Izvrši zahtevanu obradu prema tekstu zadatka.
- 3) Formira izlaznu datoteku prema tekstu zadatka.
- 4) Vodi računa i obradi moguće izuzetke koji mogu nastati prilikom rada programa.

Primeri

Primer 1	
Standardni ulaz	2 5 7 10
Ulazna datoteka:	2_dirA.txt 5_dirA.txt 7_dirA.txt 10_dirA.txt 2_dirB.txt 5_dirB.txt 7_dirB.txt 10_dirB.txt
Izlazna datoteka:	2_station.txt 5_station.txt 7_station.txt 10_station.txt
Standardni izlaz:	
Primer 2	
Standardni ulaz	27 32
Ulazna datoteka:	27_dirA.txt 32_dirA.txt 27_dirB.txt 32_dirB.txt
Izlazna datoteka:	27_station.txt 32_station.txt
Standardni izlaz:	
Primer 4	
Standardni ulaz	1
Ulazna datoteka:	1_dirA.txt #ne postoji
Izlazna datoteka:	-
Standardni izlaz:	DAT_GRESKA
Primer 5	
Standardni ulaz	27A 32E 12L
Ulazna datoteka:	-
Izlazna datoteka:	-
Standardni izlaz:	PODACI_GRESKA

3. Napisati program koji pomaže putnicima u planiranju putovanja javnim gradskim prevozom od zadate početne lokacije do zadate krajnje lokacije korišćenjem zadatih linija javnog gradskog prevoza. Program putem standardnog ulaza prihvata ime izlazne datoteke, koordinate početne i krajnje lokacije (geografsku širinu i dužinu) i spisak linija gradskog prevoza koje su od interesa za putnika. Korisnik mora navesti oznaku bar jedne linije gradskog prevoza. Program treba da izvrši zahtevanu obradu i formira izlaznu datoteke sa podacima za svaku zadatu liniju. Smatrati da putnik ne želi da menja liniju gradskog prevoza tokom putovanja. Kao početnu i krajnju stanicu uzeti one najbliže zadatim koordinatama. Prvo razmatrati samo stanice smeru A, a nakon toga, ako putanja u smeru A ne postoji, razmatrati stanice smeru B. Smatrati da korisnik ne može da prelazi iz jednog smeru u drugi tokom svog putovanja. Rastojanje između dve geografske tačke odrediti po datoj *haversine* formuli. Format jednog reda izlazne datoteke je isti kao kod ulazne datoteke. Pre ispisivanja putanje za određenu liniju ispisati u zasebnoj liniji oznaku linije i smer kretanja po formatu **naziv linije!smer**.

Program treba da:

- 1) Učita ime izlazne datoteke sa standardnog ulaza.
- 2) Učita koordinate početne i krajnje lokacije. Ukoliko koordinate ne predstavljaju valjan realan broj, prijaviti grešku o nekorektnim podacima.
- 3) Učita spisak linija sa standardnog ulaza koje su od interesa za putnika. Ukoliko podaci o liniji ne postoje, prijaviti grešku o nepostojećoj datoteci.
- 4) Izvrši zahtevanu obradu prema tekstu zadatka.
- 5) Formira izlaznu datoteku prema tekstu zadatka.
- 6) Vodi računa i obradi moguće izuzetke koji mogu nastati prilikom rada programa.

Primeri

Primer 1	
Standardni ulaz	izlaz.txt 44.777633 20.5299696 44.7569219 20.5519362 309
Ulazna datoteka:	309_dirA.txt 309_dirB.txt
Izlazna datoteka:	izlaz.txt
Standardni izlaz:	
Primer 2	
Standardni ulaz	izlaz.txt 44.7569219 20.5519362 44.795629 20.499284 309 7 46
Ulazna datoteka:	309_dirA.txt 309_dirB.txt 7_dirA.txt 7_dirB.txt 46_dirA.txt 46_dirB.txt
Izlazna datoteka:	izlaz.txt
Standardni izlaz:	
Primer 3	
Standardni ulaz	izlaz.txt 44.777633 20.5299696 44.7569219 20.5519362 1
Ulazna datoteka:	1_dirA.txt 1_dirB.txt #ne postoje
Izlazna datoteka:	-
Standardni izlaz:	DAT_GRESKA
Primer 4	
Standardni ulaz	izlaz.txt 44.aaa 20.5519362 xx.795629 20.499284 309 7 46
Ulazna datoteka:	-
Izlazna datoteka:	-
Standardni izlaz:	PODACI_GRESKA