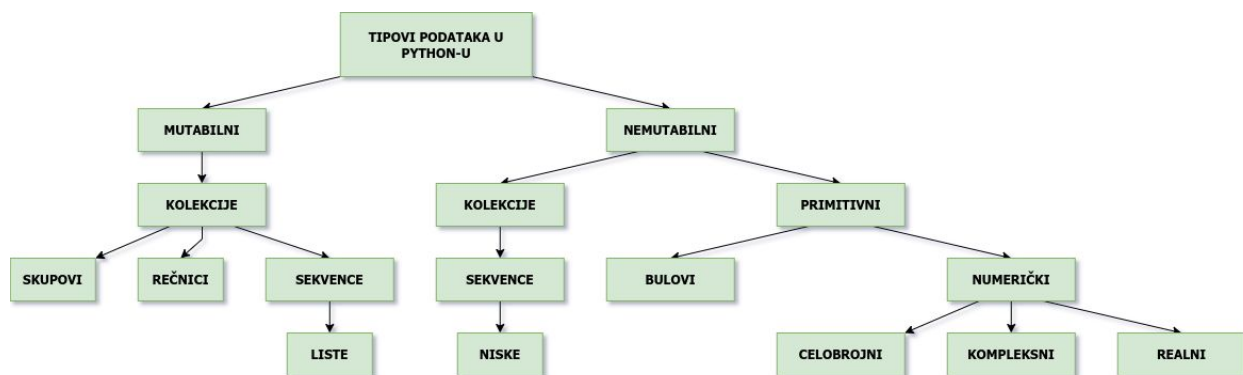


# Termin XVIII

15. februar 2020.

### Redovni zadaci



### 1. torke.py

Torka je fiksna (nemutabilna) sekvenca tipova. Ovo podrazumeva da same vrednosti te torke ne mogu da se menjaju, na sličan način kao što se sami karakteri u okviru string-a takođe ne mogu menjati. Za formiranje torki koriste se `()`.

Napisati program koji formira listu  $n$  učenika čiji se podaci unose iz zasebne linije standardnog ulaza. Odrediti ukupan broj šestaka. Podaci o učenicima se čuvaju u zasebnim torkama. Format podataka o učeniku je kao što sledi: (Ime, Prezime, Razred/Odeljenje).

#### PRIMER

##### ULAZ

4

Petar Petrović VIII/4

Milena Miletic VI/2

Katarina Simic VI/1

Stanko Dunkic VII/2

##### IZLAZ

2

## 2. skupovi.py

Skup je neuređena mutabilna kolekcija jedinstvenih tipova. Skupovni tip u Python-u predstavlja matematičku notaciju skupa, te su moguće identične operacije, poput unije i preseka dva ili više skupova. Prednost korišćenja skupova u odnosu na liste je u samoj brzini proveravanja da li je neki element sadržan u odgovarajućoj kolekciji. Skup se formira pomoću **set()**.

U dve linije standardnog ulaza nalaze se po jedna sekvenca celih brojeva koju treba učitati u dve odvojene liste. Napisati program koji od takvih listi formira skupove jedinstvenih elemenata, a zatim određuje koji su jedinstveni elementi zajednički za obe sekvence brojeva sa standardnog ulaza.

### PRIMER

#### ULAZ

9 6 7 4 4 3 2 5 5  
2 8 1 1 6 7 7 4

#### IZLAZ

2 4 6 7

## 3. kompleksni\_brojevi.py

Do sada smo bez problema rešavali jednačine tipa:

$$x^2 - 1 = 0;$$

Međutim, nekada treba da rešimo jednačinu tipa:

$$x^2 + 1 = 0.$$

U ovakvim situacijama skup realnih brojeva nije dovoljan. Tada ovaj skup moramo da proširimo skupom kompleksnih brojeva u kome će ova jednačina imati rešenje.

Napisati program koji određuje rešenja ispod navedene jednačine u skupu kompleksnih brojeva pri čemu se date konstante **a** i **b** učitavaju iz dve linije standardnog ulaza (brojevi **a** i **b** su pozitivni brojevi veći od 0).

$$ax^2 + b = 0$$

### PRIMER

#### ULAZ

3  
27

#### IZLAZ

x1 = 3j  
x2 = -3j

## 4. rečníci.py

Rečníci su mutabilna kolekcija tipova odgovarajuće strukture. Naime, rečníci podrazumevaju pravilo da tačno jednom ključu odgovara tačno jedna vrednost. Iako u jednom rečniku možemo imati iste vrednosti, sami ključevi u rečniku moraju biti jedinstveni.

U popularnom restoranu “*Dođi na klop*” vlasnik Sima Simić ima veliku odgovornost. Kako u njegovom restoranu ima ogroman broj imenovanih jela, nekada nije toliko lako naći cenu za odgovarajuće jelo, te konobari izgube dragoceno vreme dok pronađu odgovarajuću cenu. Sima je jednog dana odradio modifikaciju cenovnika i dopunio je cenovnik nazivima novih jela i njima odgovarajućim cenama. Jedan konobar je po primanju porudžbine utvrdio da je za jelo “*Sarma od kiselog kupusa*” postavljena astronomska cena i odmah je javio Simi da ispravi. Vlasnik mu je dao novu cenu i nastavio sa danom. Napisati program koji formira rečnik **cenovnik** čiji su ključevi jedinstveni nazivi jela restorana, a vrednosti cene odgovarajućih jela. Dodatno ispisati koliki će račun biti posetioca koji je naručio 1 porciju jela “*Pileci file*” i 2 porcije “*Sarma od kiselog kupusa*”. U prvoj liniji standardnog ulaza nalazi se broj **N** koji određuje broj izmodifikovanih jela. U narednih **N** linija nalaze se podaci o jelima, a zatim se u narednom redu učitava broj **C** koji predstavlja novu vrednost jela “*Sarma od kiselog kupusa*”.

### PRIMER

#### ULAZ

3

Pileci file-179.99

Sarma od kiselog kupusa-914.26

Spagete bolonjeze-229.99

64.99

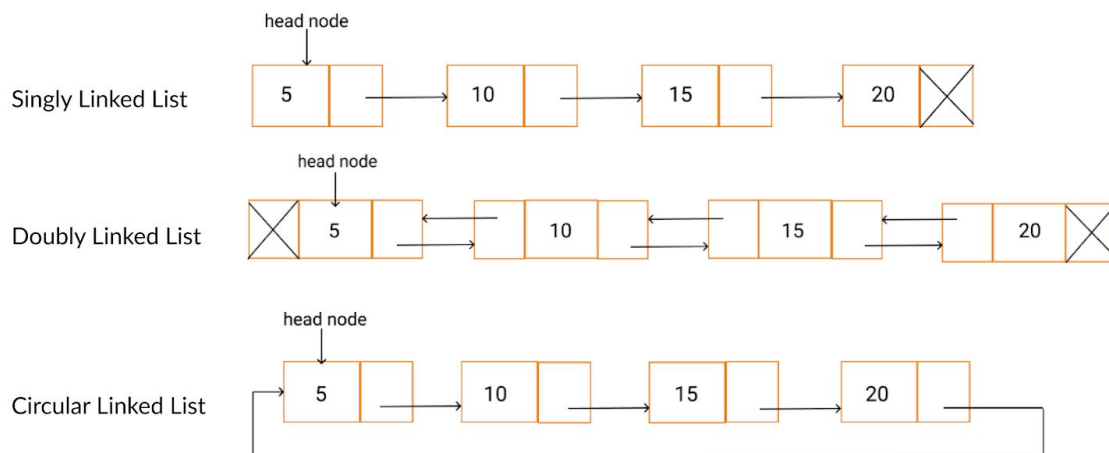
#### IZLAZ

309.97

## 5. liste.py

Lista je linearna struktura podataka pri čemu se čvorovi (u kojima se čuvaju podaci) vezuju u jedan lanac. Taj lanac je usmeren od početnog do krajnjeg čvora. Kako bismo se kretali kroz listu, moramo krenuti ili od početnog ili od krajnjeg čvora. Svaki čvor pored samog podatka zna i za adresu narednog čvora u listi. Liste mogu biti jednostruko ili dvostruko ulančane. Uz to mogu biti i kružne.

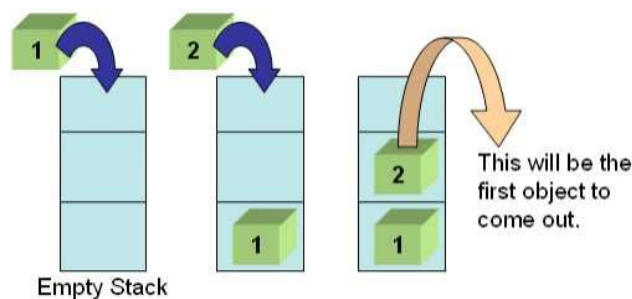
Pomoću torki i imaginarne memorijske trake (predstavljena je listom od 1024 brojeva) simulirati dodavanje u jednostruko ulančanu listu, a zatim, prolazeći kroz čvorove ispisivati elemente.



## 6. stekovi.py

Stek je linearna struktura podataka kod koje se ti podaci uređuju po LIFO varijanti (eng. *Last-In/First-Out* - Poslednji koji ulazi, prvi izlazi). Stekovi se najčešće implementiraju preko listi. Ključne funkcije za rad sa stekovima jesu **push()** (dodaje na stek) i **pop()** (skida sa steka). Funkciju **push()** možemo simulirati preko **list.append()**, dok za skidanje sa steka postoji funkcija **pop()**.

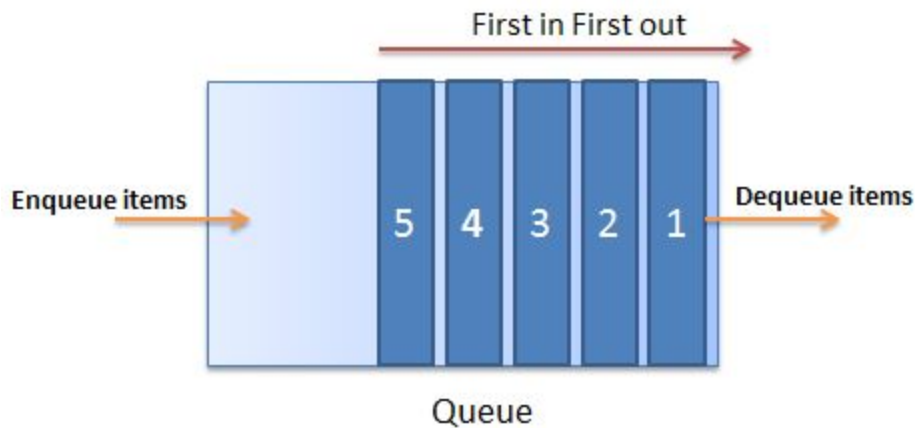
Napisati program koji simulira stek celih brojeva koji se puni i prazni. Dodatno, napisati funkciju koja prazni stek.



## 7. redovi.py

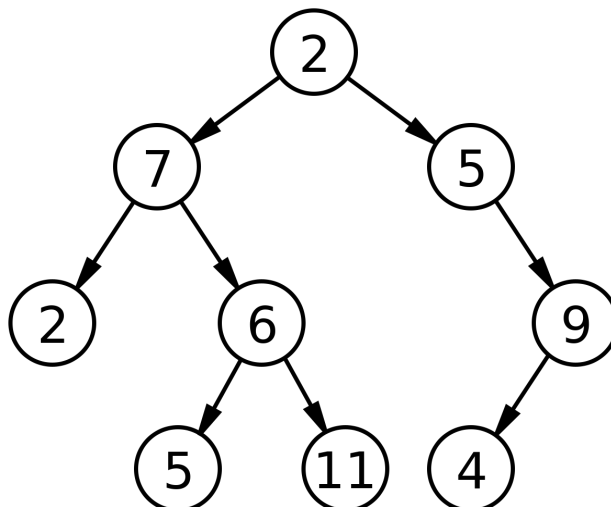
Red je linearna struktura podataka kod koje se ti podaci uređuju po FIFO varijanti (eng. *First-In/First-Out* - Prvi koji ulazi, prvi izlazi). Redovi se najčešće implementiraju preko listi. Ključne funkcije za rad sa stekovima jesu ***enqueue()*** (ubacuje u red) i ***dequeue()*** (izlazi iz reda). Funkciju `enqueue()` možemo simulirati preko ***list.append()***, dok za izbacivanje iz reda koristimo funkciju ***pop(0)***.

Napisati program koji simulira red celih brojeva koji se puni i prazni.



## 8. stabla.py

Stablo je nelinearna struktura podataka pri čemu se čvorovi vezuju u stablo odnosno svaki čvor može imati više sopstvenih lanaca. Važno je napomenuti da u okviru stabla ne sme postojati ciklus koji se uspostavlja među čvorovima. Stabla mogu biti binarna i čvorovi ovakvih stabala imaju najviše dva lanca.



## 9. grafovi.py

Graf je nelinearna struktura podataka sastojana od čvorova i veza između tih čvorova. Za razliku od stabala, ciklusi u grafu se mogu zadesiti. Grafovi mogu biti: ciklični ili aciklični, težinski i netežinski i usmereni ili neusmereni.

